

ИНФОРМАТИКА

ЖӘНЕ ЕСЕПТЕУ ТЕХНИКАСЫ

НЕГІЗДЕРІ

*Жалпы орта білім беретін мектептердің
9-сыныбына арналған оқулық*

2-басылымы

*Өзбекстан Республикасы Халыққа
білім беру министрлігі тарапынан бекітілген*

9



*Чулпан атындағы баспа-полиграфия шығармашылық үйі
Ташкент – 2015*

УДК: 372.8:004=512.122(075)
КБК 32.81(5Kaz)я721
Б 73

Авторлар:

**Б.Ж. Болтаев, А.Р. Азаматов, А.Д. Асқаров,
М.Қ. Содиқов, Г.А. Азаматова**

Жауапты редактор:

Н. Тайлақов – педагогика ғылымының докторы, профессор.

Пікір жазғандар:

*М. Арипов – ӨзҰУ «Информатика және қолданбалы бағдарламалау»
кафедрасының профессоры, физика-математика
ғылымдарының докторы,*

*М. Таиов – Наманган облысы Чуст ауданындағы 52-жалпы
орта білім беретін мектептің жоғары санатты
информатика-математика оқытушысы.*

Шартты белгілер:



Есте сақта



Сұрақтар мен тапсырмалар



Практикалық жұмыс немесе бақылау сабағы

**«Республикалық мақсатты кітап қорының қаражаты
есебінен шығарылды».**

ISBN 978-9943-05-747-0

© Б.Ж. Болтаев және басқалар, 2015
© Чулпан атындағы БПШҰ, 2011
© Чулпан атындағы БПШҰ, 2015

1-сабақ. Есептерді компьютерде шығару кезендері

Практикалық жұмыс барысында адамға өте көп мәселе шешуге тура келеді. Олардың кейбірі оңай, ал кейбірі күрделі есепкесаппен байланысты болады. Ал кейбір есептерді шығарған кезде белгілі бір амалдар тобын мыңдаған рет орындауға тура келуі мүмкін. Сондықтан, міндетсінбей, өте тез жұмыс істейтін сенің «жәрдемшің» – компьютер осы жұмысыңа жәрдем бере ала ма, егер жәрдем бере алса, онда мәселелер компьютерде қалай шешіледі, деген, сұрақ туындауы мүмкін.

Бұл сұраққа жауап беруден алдын бірнеше есептер мен олардың шешілуін қарастырайық.

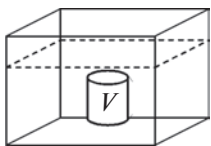
1-есеп. Көлемі 20 см^3 дене суға батырылды. Оған әсер ететін көтергіш күштің мәнін табу керек.

Есептің шартын талдаймыз: физика курсынан белгілі, суға батырылған дене өзінің көлеміне тең су көлемін сығып шығарады, ал оған сығып шығарылған судың ауырлығына тең көтергіш күш әсер етеді. Бұл күш Архимед күші деп аталады.

Сызбасы:

Берілгені:

Формулалар:



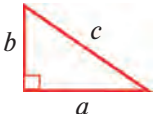
$V = 20 \text{ см}^3 = 20 \cdot \frac{1}{100} \cdot \frac{1}{100} \cdot \frac{1}{100} \text{ м}^3;$ $\rho = 1000 \frac{\text{кг}}{\text{м}^3};$ $g = 9,81 \frac{\text{Н}}{\text{кг}}.$	$F_A = \rho \cdot V \cdot g.$
<p>Табу керек: $F_A - ?$</p>	

Шешуі: $F_A = 1000 \frac{\text{кг}}{\text{м}^3} \cdot \frac{20}{1000000} \text{ м}^3 \cdot 9,81 \frac{\text{Н}}{\text{кг}} =$
 $= 0,1962 \frac{\text{кг}}{\text{м}^3} \cdot \text{м}^3 \cdot \frac{\text{Н}}{\text{кг}} = 0,1962 \text{ Н}.$

Жауабы: 0,1962 Н.

2-есеп. Мўқтар торкөзді қағазға қызыл түсті қарындашпен тірегі 16 тордан тұратын, ал биіктігі тіректің $3/4$ бөлігіне тең тік бұрышты үшбұрыш сызды. Осы үшбұрыштың периметрін тап.

Есеп шартын талдаймыз: біріншіден, есептің шешімін табу үшін үшбұрыштың қандай түсті қарындашпен сызылғанының маңызы жоқ екенін, яғни бұл біз үшін «керексіз» ақпарат; екіншіден, үшбұрыштың **тік бұрышты** болуы – шешуші маңызы бар ақпарат болып табылатынын анықтаймыз. Егер екі тордың 1 см-ге тең екені ескерілсе, онда геометрия курсынан есептің шешімі мынадай көріністе өрнектеледі:

Сызбасы:	Берілгені:	Формулалар:
	$a = 16 \text{ тор} = 8 \text{ см};$ $b = 8 \text{ см} \cdot 3/4 = 6 \text{ см}$	Периметр: $P_{\text{үшбұр}} = a + b + c$ Пифагор теоремасы: $c^2 = a^2 + b^2.$
	Табу керек: $P_{\text{үшбұр}} - ?$	

Шешуі: Пифагор теоремасынан:

$$c = \sqrt{a^2 + b^2} = \sqrt{(8\text{см})^2 + (6\text{см})^2} = \sqrt{100\text{см}^2} = 10 \text{ см}.$$

Бұдан: $P_{\text{үшбұр}} = 8 \text{ см} + 6 \text{ см} + 10 \text{ см} = 24 \text{ см}.$

Жауабы: 24 см.

3-есеп. Ерлан кітаптың төрт беті мен төрт жолын оқып шықты. Кітаптың бетінде қанша жол болса, әрбір бетте сонша белгі бар. Егер Ерлан оқып шыққан ақпарат 6560 байтқа тең болса, кітаптың бір бетінде неше жол бар екенін анықта.

Есептің талдауын жасауға өтеміз.

Есептің бастапқы деректері:

- Ерлан кітаптың 4 бетін және 4 жолын оқып шықты;
- Ерлан оқып шыққан ақпарат 6560 байт;
- беттегі жолдардың саны жолдағы белгілердің санына тең.

Есептің мақсаты:

Кітаптың бетіндегі жолдардың санын анықтау.

Есептің шартына сәйкес теңдеу құрастыру.

Есепте ізделіп отырған жолдардың санын x деп белгілейміз. Онда шартқа орай әрбір жолдың x белгісі болады. Демек, кітаптың бір бетінде x^2 (x белгіден тұратын x жол) белгі бар. Есептің шартына қарағанда Ерлан $4x^2 + 4x$ (4 бет пен 4 жол) белгі оқып шыққан. Есептің шартына сәйкес осы белгілердің саны 6560 байтқа (бір белгі – бір байт) тең:

$$4x^2 + 4x = 6560.$$

Бұл теңдеуді $x^2 + x - 1640 = 0$ көріністегі квадрат теңдеуге келтіріп, біз есептің шартына сай келетін теңдеу шығардық.

Тендеу шешімдерінің тізбегі мынадай болады :

Саған мәлім болған квадрат тендеудің шешімін шығару әдісі пайдаланылады:

- 1) дискриминантты есептейміз: $D = 1^2 - 4 \cdot 1 \cdot (-1640) = 6561 = 81^2$.
- 2) $D > 0$ болғандықтан, тендеудің екі шешімі табылады:

$$x_1 = \frac{-1 - 81}{2 \cdot 1} = -41, \quad x_2 = \frac{-1 + 81}{2 \cdot 1} = 40.$$

Есептің нәтижесін талдау:

Тендеудің екі шешімі бар екен. Бірақ кітап жолдарының саны теріс сан бола алмайды. Демек, тендеудің есепті қанағаттандыратын шешімі $x = 40$ болады. **Жауабы:** 40 жол.

Жоғарыда келтірілген есептерді талдағанда олардың шешімін шығару төменде келтірілген бірнеше кезеңнен тұратынын байқауға болады:

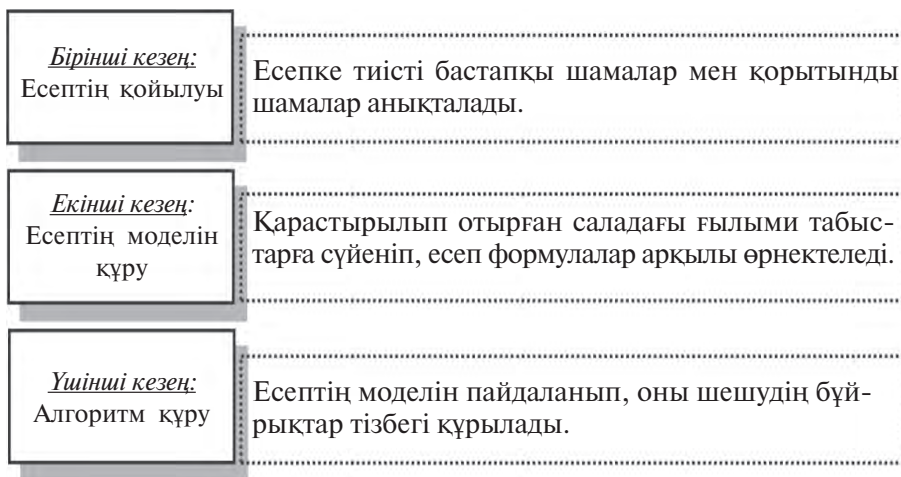
1. Әрбір есепті шығару үшін алдымен **оның қойылуы**, яғни есепте берілген бастапқы шамалар және оның мақсаты (ізіліп отырған нәтижелі шама) анықталады.

2. Есепті шешу үшін қажетті **формулалар**, басқаша айтқанда, **математикалық қатынастар** құрастырылады.

3. Мәселенің шешімін шығаруда орындалатын **амалдардың** (формулалардың, қатынастардың) **тізбегі** анықталады (бұл жағдаят әсіресе 2-және 3-есепте айқын көзге түседі).

4. Нәтижені алу және талдау.

Жоғарыда келтірілген есептерге ұқсас, басқа да есептерді компьютердің жәрдемімен шешу мүмкін және ол жоғарыдағы 4 кезеңге қосымша **амалдарды компьютер түсінетін тілге аудару** және **компьютер жадына ендіру сияқты** екі кезенді қамтиды:



<p><u>Төртінші кезең:</u> Бағдарлама жасау</p>	<p>Алгоритмдегі нұсқаулар тізбегі компьютер түсінетін тілге аударылады.</p>
<p><u>Бесінші кезең:</u> Бағдарламаны компьютер жадына енгізу</p>	<p>Құрастырылған бағдарлама компьютер жадына енгізіледі.</p>
<p><u>Алтыншы кезең:</u> Нәтиже алу және оны талдау</p>	<p>Бағдарламаны іске қосып, нәтижені талдаған соң, кателер мен кемшіліктер жойылады.</p>

Есептерді компьютерде шығару кезеңдерінің кейбірі белгілі бір білім мен дағдыны талап етеді. Сондықтан олар арнайы тақырыптар арқылы баяндалады.



Сұрақтар мен тапсырмалар

1. Есептің шешімін компьютерде шығарудың неше кезеңі бар?
2. Неліктен шығарылған нәтижеге талдау жасау керек?
3. Есеп-қисап жұмыстарын калькулятормен орындағанда қандай қате туындайды?
4. $23 + 46 \cdot 3 - 24 : 3$ арифметикалық өрнектің мәнін есептеуде орындалатын амалдар тізбегін анықта.
5. Есептің шартына сай келетін теңдеу құру үшін мысалдар келтір.

Жағыулар

Төмендегі есептердің шартын талдап, олардың шешімін кезеңдерге бөліп шығар.

1. Тоспа суға қатысты жылдамдығы 15 км/сағ болған қайықтың өзен ағымымен 2 сағатта басып өткен жолы, ағымға қарсы 3 сағатта басып өткен жолына тең болса, өзен ағымы жылдамдығын тап (бағыт: жылдамдық = жол/уақыт).

2. Тік бұрышты төртбұрыштың жақтары сәйкесінше 4 см мен 3 см болса, оның диагоналы ұзындығын тап (бағыт: тік бұрышты төртбұрыштың диагоналы оны екі тік бұрышты үшбұрышқа бөледі, демек, диагональ сол үшбұрыштардың гипотенузасы болады).

2-сабақ. Модель және оның түрлері

Бізді қызықтырып, зерттеліп отырған нәрсе яки процесс **объект** деп аталады. Мәселен, күн жүйесіндегі планеталар, спорт доптары, мектептегі компьютерлер — бұлардың бәрі объектілерге мысал болады. Бір түрдегі объектілердің өзгеше қасиеті — **сипа-**

ты болады. Ал бөлек алынған әр объект басқалардан дәл осы **сипатының мәнімен** ерекшеленеді. Мәселен, компьютерлер деп аталатын объектілердің сипаттамасына: істеп шығарған фирманың аты, негізгі платаның маркасы (motherboard), процессордың аты, процессор жылдамдығы (CPU), винчестер, шапшаң істейтін жад (RAM) пен видеожадының ақпарат сыйымдылығы жатса, айқын бір компьютер сипатының мәні: істеп шығарған фирманың аты FUJITSU SIEMENS, негізгі плата маркасы D1170, процессордың аты Pentium IV, оның жылдамдығы 3,06 Ггерц, винчестердің ақпарат сыйымдылығы 160 Гбайт, шапшаң зерденің сыйымдылығы 1 Гбайт, бейнезерденің сыйымдылығы 512 Мбайт болады.

Егер зерттеліп отырған объектілер планеталар болса, онда:

Планеталар сипаттамасы	пішіні	массасы	радиусы	айналу жылдамдығы
Жердің сипатты мәні	шар тәрізді	$5976 \cdot 10^{21}$ кг	6378 км	30 км/сек

«Доп» деп аталатын объекті үшін:

Доптар сипаттамасы	пішіні	массасы	радиусы	материалы
Жай доптың сипатты мәні	сфера тәрізді	2,2 кг	15 см	резенке

Көп жағдайда, белгілі бір салаға қатысты ізденістер жүргізгенде нағыз объект емес, оның әлдеқандай мағыналық көшірмесі зерттеледі. Бұған, бір жағынан карағанда, белгілі бір себептерге (мәселен, найзағайдың тұрақты еместігі, күннің аулақтығы, объектімен жұмыс істеу үшін өте көп қаржы керектігі, адам өміріне қауіп төндіруі және т. б.) бола шын объектіні тікелей үйренуге жағдай болмаса, екінші жағынан, ізденістер жүргізу үшін кейде объектінің әлдеқандай мағыналы көшірмесін үйренудің өзі жеткілікті болады. Әрине, бұндай жағдайда объектінің көшірмесі зерттеліп отырған саланың талаптарына толық сай болуы керек.



Модель – шын объектінің қарастырылып отырған саланың белгілі талаптарына сай болатын **көшірмесі**.

Модель сөзі (лат. **modulus** – өлшем, мөлшер, шама) саған ұшак, машинажасау яки кемежасау үйірмесі арқылы белгілі. Тұрмыста объектілердің модельдеріне өте көп мысал келтіруге

болады. Мәселен, жердің модельдері ретінде глобус немесе карта; ұшақтың моделі ретінде оның кішірейтілген көшірмесі; автомашинаның моделі болып саған таныс ойыншықтар; найзағайдың моделі ретінде жоғары кернеулі электр көзіндегі қысқа тұйықталу яки дәнекерлейтін электродтың жануы; адам моделі ретінде оның жасушасы мен фотосуреті немесе қуыршақ; адам миының есептеуге қатысты моделі ретінде калькулятор яки компьютер қызмет етеді.

Шын объект пен оның моделі жүргізіліп жатқан тәжірибелерде бірдей нәтиже бергенде ғана қарастырылып отырған саланың талаптарына сай бола алады. Мәселен, ұшақ пен оның кішірейтірілген көшірмесі — модель бірдей аэродинамикалық заңдарға бағынады. Модельге қатысты табылған нәтиже шын ұшақ үшін де орынды. Жобаланған ұшақты құрып болған соң, оны зертханада арнайы қондырғыларда — ұшаққа ауа ағысын жіберетін стендтерде сынақтан өткізеді. Бұндай жағдайда зертханадағы стенд атмосфераның моделі ретінде қызмет етеді.

Тұрмыста сондай процестер де болады, олардың моделі деп математикалық қатынастар мен формулалар қарастырылады. Осындай жағдайда тандалған модель шын объектінің қасиеттерін өзінде үйлестірген болуы керек, яғни зерттеліп отырған объектінің және тандалған модельдің қасиеттері бірдей қатынастар мен формулалар арқылы өрнектелуге тиісті.



Зерттеліп тұрған объектінің математикалық қатынастар, белгілер мен байланыстар арқылы өрнектелген бейнесі объектінің **математикалық моделі** деп аталады.

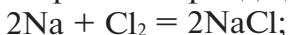
Зерттеліп отырған объектінің сипаттамасын математикалық қатынастар мен белгілер арқылы бейнелеу процесі математикалық модельдендіру деп аталады. Бұрынғы сабақта қарастырылған кітап бетіндегі жолдардың санын табу жөніндегі мысал квадрат тендеу арқылы өрнектелген. Демек, есепті квадрат тендеу түрінде бейнелеу процесі математикалық модельдендіру, ал оған тән тендеу — есептің математикалық моделі болады. Сондай-ақ, Архимед күші, Пифагор теоремасы, периметрді есептеп шығаратын формула да математикалық модель болып табылады.

Математикалық модельдендіру процесі ежелден астрономия, химия, физика пәндерінде қолданылып келген. Мысал ретінде Нептун планетасының ашылуын алуға болады. 1846 жылы француз астрономы У. Лаверье Уран планетасының ерекше қозғалы-

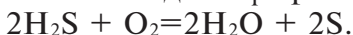
сына Күн системасының сол кезге дейін мәлім болмаған планетасы себепші екенін математикалық тұрғыдан дәлелдеп берген. Сол жылы Леверьенің тұжырымдарына сүйеніп, неміс астрономы Галилей Нептун планетасын телескоппен бақылаған.

Химиялық реакциялардың математикалық моделіне мысалдар:

1) хлор мен натрийдің қосылу реакциясы:



2) табиғи газдан күкірт алу реакциясы:



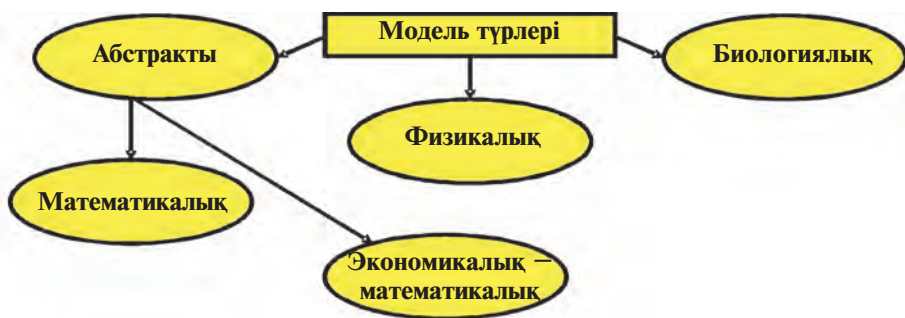
Физикалық құбылыстардың математикалық моделіне мыналар мысал болады:

1) Ньютонның екінші заңы, яғни денеге әсер ететін күштің формуласы: $F = ma$, мұндағы m – дененің массасы, a – үдеу;

2) Ньютонның бүкіләлемдік тартылыс заңы: $F = G \frac{m_1 m_2}{R^2}$,

мұндағы m_1, m_2 – бір-біріне әсер етіп жатқан денелердің массалары, R – олардың арасындағы қашықтық, G – гравитация тұрақтысы.

Қазіргі таңда да модельдендіру химия, биология, медицина, экономика сияқты ғылымның түрлі бағыттарында кең қолданылып, өте қызықты нәтижелер беруде. Жалпы алғанда, модельдер объектілерді бейнелеудің таңдап алынған құрал-әдісіне қарай төмендегі схемада көрсетілген негізгі үш топқа бөлінеді:



1. Абстракты модельдер өз кезегінде екі топқа бөлінеді: **математикалық** және **экономикалық-математикалық модельдер**.

Математикалық модельдер объектінің құрылысын, өзара байланыс заңдылықтарын математикалық қатынастар, формулалар және математикалық-логикалық сипаттама арқылы өрнектейді. Мұндай модельдерге біз өткен сабақта мысалдар келтірген едік.

Экономикалық-математикалық модельдер XVIII ғасырдан қолданыла бастады. Ф. Кене өзінің «Экономикалық кестелер» атты еңбегінде бірінші рет бүкіл қоғамдық өндіріс процесінің қалыптасуын модель арқылы бейнелеуге әрекет жасады. Қазіргі таңда экономикалық модельдердің көмегімен экономикалық дамудың жалпы заңдылықтары тексеріледі. Түрлі экономикалық көрсеткіштер, соның ішінде ұлттық табыс, тұтыну, жұмысбастылық, қорлар, инвестиция көрсеткіштерінің өзгерісін және қатынасын талдау, оны алдын ала айту үшін күрделі экономикалық модельдер қолданылады. Тәуелсіз Өзбекстанда реформалардың **5 принципке сүйеніп, базар қатынастарына көшу моделі** де экономикалық-математикалық модель болып табылады (осы принциптерді есте сақта!).

2. Физикалық модель объектінің жаратылысы мен құрылысының дәлме-дәл дерлік көшірмесі болып, бірақ одан мөлшері (өлшемі, жылдамдығы т. с. с.) жағынан ерекшеленеді. Бұған мысал ретінде ұшақ, кеме, автомашина, пойыз және т.б. модельдерін келтіруге болады.

3. Биологиялық модель түрлі жанды объектілерге, олардың құрамдас бөліктеріне (жасуша, ұлпа т. б.) тән биологиялық құрылыстарды, функциялар мен процестерді модельдендіру кезінде қолданылады. Биологиялық модель адамдар мен жануарларда байқалатын белгілі бір күйді немесе ауруды лабораториялық тірі жәндіктің бойында сынап көруге мүмкіндік береді.

Мәселен, зиянды вирусты өлтіретін дәріні тексеру үшін адамның өзінде емес, оның біраз қанын алып, дәріні осы қанда сынап көру жеткілікті болады.

Төменде физикалық және биологиялық процестердің математикалық модельдеріне мысалдар келтірілген.

1-есеп. Қысымы p , алып отырған көлемі V және температурасы T болған идеал газдың күйін бейнелейтін математикалық модель құр.

Есептің шешімін шығару үшін Клапейрон формуласын, яғни идеал газдың қысымы, көлемі мен температурасын байланыстыратын формуланы пайдаланамыз:

$$\frac{pV}{T} = \text{const.}$$

Бұл формула идеал газдың температурасы өзгерген сайын оның қысымы мен көлемінің өзгеріп тұратынын айқын көрсетіп тұр.

2-есеп. Гүлдің өсу процесін бейнелейтін модель құр.

Өсімдіктердің өмір сүруі мен өсуі үшін ауа, жарық, су мен азық керек болуын сен ботаника курсынан білесің. Мәселен, кейбір гүлдер қараңғы,

күрғақ жерде жақсы өседі, ал басқа гүлдер көбінесе жарық пен ылғалдықты талап етеді. Сондықтан есептің моделі мына тендеулер жүйесі арқылы өрнектеледі:

$$\begin{cases} T = T_0 \cdot (1 + \alpha t); \\ I = I_0 \cdot (1 + \beta t); \\ H = H_0 \cdot (1 + \gamma t), \end{cases}$$

мұндағы t — уақыт; T — ауа температурасы; I — жарықтық мөлшері; H — гүл денесіндегі ылғалдық мөлшері; α , β , γ — температураға, жарықтыққа және ылғалдыққа сәйкес келетін тұрақты шамалар.

Қарастырылған есептердің модельдеріне назар аударсақ, мынаны айтуға болады: қай салада болса да, математикалық модель жасау үшін тек қана математикаға қатысты емес, басқа салаларға да қатысты білімге жеткілікті дәрежеде ие болу керек екен.



Сұрақтар мен тапсырмалар

1. *Объект деп нені айтады?*
2. *Объектінің сипаты мен сипатының мәні жайлы мысалдарға сүйеніп, айтып бер.*
3. *Модель деген не?*
4. *Объектіге және оған сай модельдерге мысалдар келтір.*
5. *Математикалық модель деп нені айтады? Олар қандай салаларда қолданылады?*
6. *Математикалық модельдің басқа модельдерден айырмашылығын түсіндір.*
7. *Нептун планетасын қалай ашқан?*
8. *Математикалық модельдердің химия және физика салаларындағы пайдаланысына мысалдар келтір.*
9. *Модельдер неше түрге бөлінеді?*
10. *Қандай абстракты модельдер бар?*
11. *Экономикалық-математикалық модельдер туралы айт.*
12. *Қандай физикалық модельдерді білесің?*
13. *Биологиялық модельдердің маңызы туралы айтып бер.*

Жаттығулар

Төмендегі объектілердің сипаты мен сипатының мәнін жаз.

1. Объект: облыстар (бағыт: олардың аты, ауданы, халық саны, негізгі экономикалық өнімі, ...).
2. Объект: сыныптастар (бағыт: олардың жынысы, бойы, шаш түсі, салмағы, көз түсі, ...).
3. Объект: кітаптар (бағыт: олардың аты, беттер саны, түсі, салмағы, бағасы, ...).

3-сабақ. Есептерді компьютерде шығару кезендері және модельдер түрі бойынша тақырыптарды қайталау

1. Төмендегі есептердің шартын талдап, шешімін кезендерге бөліп шығар.

А. a мен b катеттері бар тік бұрышты үшбұрыштың гипотенузасын есепте.

Ә. a мен b катеттері бар тік бұрышты үшбұрыштың ауданын тап.

Б. Тең қабырғалы үшбұрыштың қабырғасы a болса, оның биіктігін тап.

2. Төмендегі объектілердің сипаты мен сипатының мәнін жаз.

а) объект: сенің облысындағы (қаландағы) колледждер (бағыт: олардың аты, құрылған жылы, мамандандырылғаны, қабылданатын оқушылар саны, ...).

ә) объект: Асакадағы автомобиль зауытынан шығатын автомобильдер (бағыт: олардың үлгісі, өндіріле бастаған жылы, саны, түсі, ...).

3. Төмендегі есептерге сай келетін модель құрып, олардың шешімін шығар.

А. Банкіге жылына A пайыздық пайда келтіретін B сум мөлшеріндегі ақшаның M жылдан кейінгі жағдайын бейнелейтін модель құр.

Бағыт. 1-жылдың соңында алынатын табыс $\frac{B}{100} \cdot A$ сум болады. Сон-

дықтан банкідегі ақша жыл соңында $\frac{B}{100} \cdot A + B = B \cdot \left(\frac{A}{100} + 1\right)$ сум болады.

Екінші жылдың соңында алынатын табыс $B \cdot \left(\frac{A}{100} + 1\right) \cdot \frac{A}{100}$ сум болады.

Демек, банкідегі ақша екінші жылдың аяғында

$B \cdot \left(\frac{A}{100} + 1\right) \cdot \frac{A}{100} + B \cdot \left(\frac{A}{100} + 1\right) = B \cdot \left(\frac{A}{100} + 1\right) \cdot \left(\frac{A}{100} + 1\right) = B \cdot \left(\frac{A}{100} + 1\right)^2$ сум болады.

Үшінші және төртінші жылдардың соңында банкідегі ақшаның қанша болатынын есептеп шығып, формулаларды жинақтап қорыт.

Ә. Ұшақ A қаладан B қалаға дейінгі 2100 км-ге тең аралықты 3 сағатта, ал B қаладан M қалаға дейінгі 4800 км-ге тең аралықты 6 сағатта ұшып өтті. Ұшақ қандай орташа жылдамдықпен ұшқан? (бағыт: орташа жылдамдық $= (1\text{-жол} + 2\text{-жол}) / (1\text{-уақыт} + 2\text{-уақыт})$).

4-сбақ. Алгоритм ұғымы

Адам баласы бүкіл өмірі бойына онай да күрделі мәселелер мен есептер шешуді өз алдына мақсат етіп қояды. Әдетте, мақсатына жету үшін ол орындалуға тиісті амал-қимылдарын өмір тәжірибесіне яки игеріп алған біліміне сүйеніп, белгілі бір тәртіпке келтіреді. Бұған көптеген мысал келтіруге болады.

1-мысал. Шайды демдеу мақсаты қойылған болсын. Онда шайды демдеп отырған адамға біз үшін үйреншікті болып кеткен мынадай жұмыстарды орындауға тура келеді:

- 1) шәйнектің қақпағы алынсын;
- 2) шәйнек қайнаған сумен шайылсын;
- 3) бір қасық мөлшеріндегі құрғақ шай шәйнекке салынсын;
- 4) шәйнекке толғанша су құйылсын;
- 5) шәйнектің қақпағы жабылсын;
- 6) шәйнек үстіне сүлгі жабылып, бес минуттай демделсін.

2-мысал. Бір жердің ені N метр, ұзындығы M метр. Осы жердің өлшемі 12×25 см (ені 12 см, ал ұзындығы 25 см) болған қышпен толтыру үшін неше қыш сатып алу керектігін табу қажет болсын. Есепті шығарушы адам геометрия курсынан алған біліміне сүйеніп, мынадай амалдарды тізбектеп орындайды:

- 1) жердің $S_{жс}$ ауданы сантиметр өлшем бірлігінде табылсын;
- 2) бір қыштың $S_{к}$ ауданы сантиметр өлшем бірлігінде табылсын;
- 3) қыштар саны жердің $S_{жс}$ ауданын қыштың ауданына бөлгенде шығатын шама деп алынсын. Осындай амалдар тізбегін мына математикалық формула бойынша өрнектеуге болады:

$$S_{сан} = \frac{S_{жер}}{S_{кыш}} = \frac{N \cdot 100 \cdot M \cdot 100}{12 \cdot 25} .$$

3-мысал. $19632107 + 19702202$ амалы орындалсын. Бұл амалды сен қалайша орындаған болар едің? Иә, әрине, бұл сандарды сен баған көрінісінде былайша дерлік қосасын:

- 1) сандарды разрядтарына сай келетін тәртіппен бір бағанға бірінің астына екіншісін жазып шығасын;
- 2) сандардың бірлік разрядындағы цифрларын қосып, нәтиженің бірлік цифрын бірліктер астына жазып, ондық цифрын ойша сақтайсын;
- 3) сандардың ондық цифрларын бір-бірімен және ойша сақталған цифрмен қосып, нәтиженің бірлік цифрын ондықтар астына жазып, ондық цифрын ойша сақтайсын; 3-пункттегі ережені сандардың жүздік, мыңдық разрядтары үшін қайталайсын. Бұл амалдардың төменде келтірілген көрінісі саған жақсы таныс:

$$\begin{array}{r} 19632107 \\ +19702202 \\ \hline 39334309 \end{array}$$

Жоғарыда келтірілген амалдар тізбегін, басқа сөзбен айтқанда, нұсқаулар яки бұйрықтар тізбегін бірер кісі орындаған соң, алдымызға қойған мақсатқа жетуге болады. Белгілі бір нәтижеге әкелетін амалдардың тізбектеп орындалуын талап ететін, тұрмысымызда күн сайын, әр сағатта кездесетін әртүрлі ережелер арасындағы қағидалар информатиканың негізгі ұғымдарының бірі — **алгоритм** сөзі арқылы өрнектеледі.

Алгоритм сөзі IX ғасырда өмір сүріп (783—850), өзінің ғылыми еңбектерімен әйгілі отандасымыз, ұлы астроном, математик әрі географ Әбу Абдулла Мұхаммед ибн **Мұса әл-Хорезмидің** атынан шыққан. Әл-Хорезмидің арифметикаға арналған туындысы XII ғасырда Испанияда латын тіліне аударылған. Бұл аударманың XIV ғасырда қолмен жазылған жалғыз көшірмесі Кембридж университетінің кітапханасында сақталуда. Шығарма латын тіліндегі «**Dixit Algorithmi**», яғни «Дейді әл-Хорезми» деген сөздермен басталады.

Алгоритмдегі әрбір тапсырма яки нұсқау белгілі бір **амалдың** орындалуын көздейді. Ондағы амалдарды орындайтын объект **орындаушы** ұғымымен байланысты. Кез келген алгоритм амалдарды белгілейтін ереже болып, олардың тізбегі арқылы берілген мәндерден ізделіп отырған нәтиже алынады. Мұндай амалдар тізбегі **алгоритмдік процесс**, ал әрбір амал **алгоритмнің қадамы** деп аталады.



Алгоритм дегенде белгілі бір мақсатқа жетуге бағытталған, **орындаушыға** орындау үшін молжалданған нұсқаулар тізбегі түсініледі.

Демек, жоғарыда келтірілген нұсқаулар (бұйрықтар) тізбегі **алгоритм**, ал оларды орындап отырған адам — **орындаушы** болады екен. Бірінші мысалдағы нұсқаулар «шайды демдеу алгоритмі» деп аталады. Бұдан мынадай қорытынды шығаруға болады: адам өмірінде көздеген мақсатына жету үшін орындаушы ретінде көптеген алгоритмдерді орындайды. Көптеген алгоритм адам үшін әдет болып кеткен. Бұларға мысал ретінде тамақ дайындау, тамақтану, киімді жарасымды етіп кие білу, бөлмеден шығу, жазу, бір жерден басқа жерге орналасу және т.б. ережелерді келтіруге болады.

Алгоритмдегі нұсқаулардың рет-ретімен орындалуын бұзу, қандай нәтижеге келтіретінін сен көз алдына келтіре аласың. Мәселен, «шайды демдеу алгоритміндегі» бірінші мен үшінші пункттегі нұсқаулардың орнын ауыстырып орындаудың өзі

жеткілікті. Әдетте, алгоритмдегі нұсқаулар орындаушыға түсінікті болуы үшін жеңіл амалдардан құрастырылады. Екінші мысалдағы алгоритмнің бірінші пункттегі нұсқауын мынадай үш нұсқауға бөлуге болады:

1 а) жердің N метр ені сантиметр өлшем бірлігіне аударылсын;

1 ә) жердің M метр ұзындығы сантиметр өлшем бірлігіне аударылсын;



1 б) жердің ауданы $S_{жер}$ табылсын.

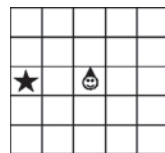
Әрине, бұл жерде алгоритмді орындаушы әрқашан адам бола бере ме, деген сұрақ туындайды. Бұл сұраққа берілген жауап мынадай:



Алгоритмді орындаушы – алгоритмде көрсетілген нұсқауларды яки бұйрықтарды орындай алатын абстракты немесе ақиқат (техникалық яки биологиялық) жүйе.

Әрбір орындаушының орындай алу мүмкіншілігі бар нұсқаулар яки бұйрықтар жиынтығы **орындаушының нұсқаулар жүйесі** (қысқаша – **ОНЖ**) деп аталады. Мысалы, «16 санынан квадрат түбір шығарылсын» деген нұсқау 2-сыныптағы оқушының нұсқаулар жүйесіне жатпайды, бірақ 8-сыныптағы оқушының нұсқаулар жүйесіне тиісті болады. Ескерте кету керек, Информатикада алгоритмнің негізгі орындаушысы ретінде **компьютер** қызмет етеді. Төменде орындаушының нұсқаулар жүйесін мысалмен түсіндіреміз.

4-мысал. Бауырсақ үшін «алдағы» тор – қалпақшасы көрсетіп тұрған тор, мәселен, ол оңға бұрылса,  көріністе болады. Бауырсақ алдағы 1 торға жүре алады, немесе тұрған торкөзінде оңға я солға қарай жылжи алады. Бауырсақ бір тордан бірнеше рет өте алады да. Ол өзі тұрған тордан  деп белгіленген торға белгілі бір жолмен бара алатын болса, қажетгі нұсқаулар тізбегін құрастыр. Есептің шартынан орындаушының – Бауырсақтың нұсқаулар жүйесін жазуға болады, яғни **ОНЖ**={**алға; оңға; солға**}. Енді есептің шешімі ретінде төмендегі алгоритмдердің біреуін алу мүмкін:



Қадамдар саны	1-алгоритм	2-алгоритм	3-алгоритм
1	1) солға;	1) оңға;	1) алға;
2	2) алға;	2) оңға;	2) солға;
3	3) алға.	3) оңға;	3) алға;
4		4) алға;	4) алға;
5		5) алға.	5) солға;
6			6) алға.

Демек, **есептің шешіміне** келтіретін алгоритм **жалғыз бола бермеуі** де мүмкін екен.

Жоғарыда қарастырылған есептерден және айтылған мәселелерден мынадай қорытынды шығарамыз: алгоритмді орындау барысында орындаушы көзделген мақсатын білмеуі де мүмкін.

Мәселен, төмендегі алгоритмді орындаудан көзделген мақсат не екені алдын ала білінбейді:

- 1) N және M натуралды сандар алынсын;
- 2) S саны нөлге тең деп алынсын;
- 3) N және M сандарының үлкені өзінен кіші санның айырмасына тең деп алынсын да, S -ке бір қосылсын;
- 4) егер N және M сандарының екеуі де үлкен болса, 3-пунктке өтілсін, ондай болмаса, кейінгі пунктке өтілсін;
- 5) жауап ретінде S жазылсын.

Бұл алгоритм мына есептің шешімін табуға мүмкіндік береді:

5-мысал. Қабырғалары N және M натуралды сандарға тең тік бұрышты төртбұрыш берілген.

Егер әр қадамда ең үлкен ауданы бар квадрат кесіліп алына берсе, неше квадрат кесіп алуға болады?

Біз осы сабақтағы есептерді компьютерде шығарудың негізгі кезеңдерінің бірімен байланысты, **алгоритм, алгоритмді орындаушы, орындаушының нұсқаулар** жүйесі сияқты информатиканың негізгі ұғымдарымен танысып шығып, мына қорытындыға келдік: орындаушы **алгоритм арқылы басқарылады**.



Сұрақтар мен тапсырмалар

1. Алгоритм дегенде нені түсінесің?
2. Алгоритм сөзінің пайда болу тарихын айтып бер.
3. Алгоритмге мектеп тұрмысынан мысалдар келтір.
4. Берілген тақырыпты оқулықтан табу алгоритмін құр.
5. «Палау» пісіру алгоритмін құр.
6. Компьютерді іске қосу алгоритмін құр.
7. Алгоритмді орындаушы жөнінде не білесің?
8. Орындаушы қандай нұсқауларды орындай алмайды?
9. Орындаушының нұсқаулар жүйесіне мысалдар келтір.
10. Сыныптасың орындай алмайтын нұсқауларды жаз.
11. Төменде келтірілген нұсқаулар алгоритм деп табыла ма? Олардың орындалуы қандай мақсатты көздейді?
 - 1) өзеннен бір шелек су алынсын;
 - 2) шелектегі су өзенге төгілсін;
 - 3) 1-пунктке өтілсін.

5-сабақ. Алгоритмнің негізгі қасиеттері

Алдыңғы сабақта алгоритм және алгоритмді орындаушы жөнінде сөз қозғалған еді. Енді алгоритмнің негізгі қасиеттерімен кеңінен танысамыз:

1. Түсініктілік. Алгоритм орындаушыға түсінікті болу үшін орындаушының мүмкіншіліктерін білу қажет. Егер орындаушы адам болса, онда алгоритмді адамның мүмкіншіліктеріне сүйеніп жасау керек болады. Мұндағы көзделген мақсат, алгоритмге сүйене отырып, адам түсінетін тіл, оның білімі, өмір тәжірибесі, кәсіби деңгейі, жасы, керек десе оның дене мүмкіншіліктері есепке алынуы керек. Егер орындаушы техникалық құрал (мысалы, компьютер, электронды сағат, станок) болса, онда алгоритм осы техникалық құрылғылардың мүмкіншіліктеріне сәйкес жасалуы қажет.

Демек, қандай да бір берілген нұсқау орындаушының нұсқаулар жүйесінен алынған болуы, яғни орындаушы оның қалай орындалатынын білу керек.

2. Анықтық. Алгоритмдегі барлық амалдар, нұсқаулар яки бұйрықтар бір мағыналы және анық болуы керек. Мәселен, «біраз тұз салынсын» (бір ас қасық па, бір шай қасық па, әлде бір шыны аяқ мөлшеріндегі тұз ба?), «керегіне қарай су құйылсын» (керек дегенде судың қандай көлемі түсініледі: 1 литр, 100 литр, әлде 1 тонна ма?), «шығарма жазып келінсін» (қай тақырыпқа?) сияқты анық емес нұсқаулар әртүрлі (көбінесе керексіз) нәтижелерге келтіреді.

Бұдан мынадай қорытындыға келеміз: анықтық қасиетіне сүйеніп, алгоритмді орындаушы нұсқаулар тізбегін ойланбай (қатесіз) орындайды да, қосымша түсініктемені талап етпейді.

3. Дискреттілік (үздіктілік, жекешелік). Алгоритмдегі есепті шығару процесі жеке алынған қарапайым нұсқаулар тізбегі ретінде бірте-бірте орындалуы керек. Бұл алдыңғы сабақтағы мысалдардан айқын көрінеді.

4. Нәтижелілік (шектілік). Алгоритмнің сипаттамасында «белгілі бір мақсатқа жетуге арналған» деген сөз тіркесі қолданылады. Бұл мақсатты жоғарыда келтірілген мысалдардан көруге болады: шайды демдеу, қыштар санын анықтау, қосындысын табу. Бұлардың бәрі алгоритмнің **нәтижелілік** қасиетімен байланысты.

Бұл қасиеттің мазмұны сол, әрбір алгоритм шекті қадаммен орындалған соң, сөзсіз нәтиже беруі шарт. Соны да айту керек, алгоритм алдын ала қойылған мақсатқа жеткізе бермеуі де мүмкін. Бұған кейбір алгоритмнің бұрыс жасалғаны яки жіберілген басқа қате себеп болады. Екінші жағынан қарағанда, қойылған мәселенің жөнді шешімі әрқашан да бола бермейді. Соған қарамастан, кері **нәтиже** де нәтиже деп қабылданады.

1-мысал. $x^2 + x + 1 = 0$ квадрат тендеудің шешімі анықталсын.

Төменде келтірілген « $ax^2 + bx + c = 0$ ($a \neq 0$) түрдегі квадрат тендеуді шешу» алгоритмін пайдаланып, тендеу шешімінің жоктығына көз жеткіземіз. Бұл да – нәтиже.

- 1) a, b, c шамалар анықталсын;
- 2) дискриминант: $D = b^2 - 4ac$ есептелсін;
- 3) егер $D < 0$ болса, тендеудің шешімі жоқ деп алынсын да, 6-пунктке өтілсін;

4) егер $D = 0$ болса, жалғыз ғана шешім $-\frac{b}{2a}$ -ға тең деп алынсын да, 6-пунктке өтілсін;

5) бірінші шешім $\frac{-b - \sqrt{D}}{2a}$ -ға тең, ал екінші шешім $\frac{-b + \sqrt{D}}{2a}$ -ға тең деп алынсын;

6) аяқталсын.

Егер назар аударған болсаң, дискриминант нөлден кіші және нөлге тең болған жағдайлар тексерілді, ал нөлден үлкен болған жағдай тексерілген жоқ. Мұның себебін ойлап көр!

Демек, алгоритм әрқашан шекті кадамнан тұруға, сөзсіз нәтиже беруге тиісті екен.

5. Жалпылық. Бірер есепті шығаратын алгоритм жалпы жағдай үшін, яғни тек алғашқы мәліметтері бойынша ерекшеленетін, бір типтегі есептер тобына арналып құрылады. Мәселен, жоғарыда келтірілген « $ax^2 + bx + c = 0$ ($a \neq 0$) түріндегі квадрат тендеудің шешімін табу» алгоритмі еркін a, b, c сандар үшін нәтиже есептеп береді, яғни алгоритмнің жалпылық қасиеті дұрыс.

Берілген екі натуралды санның ең үлкен ортақ бөлгішін (ЕҮОБ) табу жөніндегі төменде келтірілген **Эвклид алгоритмі** де барша натуралды сандар үшін орынды.

2-мысал. N және M натуралды сандарының ең үлкен ортақ бөлгіші табылсын:

- 1) егер $N = M$ болса, нәтиже N деп алынсын және 4-пунктке өтілсін;
- 2) N және M сандарының үлкені өзінен кіші санның айырмасына тең деп алынсын;
- 3) 1-пунктке өтілсін;
- 4) аяқталсын.

Мынаны қорыта айтуға болады: жоғарыда келтірілген қасиеттердің бәрі орындалғанда ғана нұсқаулар тізбегі алгоритм болып саналады және белгілі бір (дұрыс яки бұрыс) нәтижеге келтіреді.



Сұрақтар мен тапсырмалар

1. Алгоритмнің қандай негізгі қасиеттері бар?

2. Түсініктілік қасиетін орындайтын және орындамайтын нұсқауға мысал келтір.
3. Орындаушыға түсінікті болу үшін нұсқауларды қандай жүйеден алу керек?
4. Орындаушы алгоритмді ойланбай орындаған кезде қандай қасиеттің маңызы өте зор?
5. Алгоритмнің дискреттілік қасиетін мысалдармен түсіндір.
6. Алгоритмнің нәтижелілік қасиетін мысалдармен түсіндір.
7. Нәтижелілік қасиетін орындамайтын нұсқаулардың тізбегіне мысалдар келтір.
8. Алгоритмнің жалпылық қасиетін мысалмен түсіндір.
9. Эвклид алгоритмінің жәрдемімен бірнеше нәтиже шығар.

6-сабақ. Алгоритм ұғымы және алгоритмнің негізгі қасиеттері бойынша тақырыптарды қайталау сабағы

1. Орындаушы ретінде сен төмендегі нұсқаулардан қайсысын орындай алмайсың? Не себептен?

А. Ауырлығы 200 кг тас көтерілсін.

Ә. 7-ге 2 көбейтілсін.

Б. 1-ден 31622400000 дейін саналсын.

2. Алдына қойылған мақсатқа жету үшін алгоритмді орындаушы қандай қарапайым нұсқауларды орындауға тиісті екенін, яғни орындаушының нұсқаулар жүйесін анықта.

А. Ашық есік орындаушының сол жағынан 5 қадам алыс тұрса, мақсат — «есіктен шығу».

Ә. Орындаушы сукұбырдың шүмегі мен цилиндр тәріздес стакан алдында тұрған болса, мақсат — «жарты стакан су алу».

Б. Берілген $44 \cdot 15 + 12 \cdot 15 : 20 - 43$ сандық өрнектің мәні анықталсын.




3. Берілген нұсқаулар жәрдемімен есептің шешіміне келтіретін алгоритм құр.

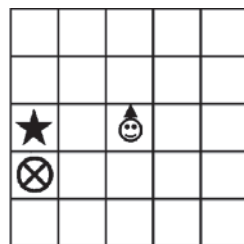
А. «Қасқыр, ешкі және қырыққабат» деп аталатын ежелгі есеп. Өзеннің сол жағалауындағы дикан қасқыр, ешкі, қырыққабатпен бірге тұр. Ол солардың бәрін оң жақтағы жағалауға көшіріп өткізуге тиісті. Оның қайығы өте тар, сондықтан ол тек бір ғана жолаушыны — қасқырды, яки ешкіні, яки қырыққабатты алып өткізе алады.

Оның үстіне, егер қасқыр мен ешкі бір жағада қалып қойса, қасқыр ешкіні жеп қояды, ал егер ешкі мен қырыққабатты бірге қалдырып кетсе, ешкі қырыққабатты жеп қояды.

Жануарлар тек дикан бар кезде ғана тыныш тұрады. Диканның нұсқаулар жүйесі мынадай болады:

{ешкіні өткіз; қасқырды өткіз; қырыққабатты өткізіп таста; жүзіп өт}.

В. Бауырсақ үшін «алдағы» тор — қалпақшасы көрсетіп тұрған тор. Ол оңға бұрылса,  көріністе болады. Бауырсақ алдағы бір торға жүре алады, әлде тұрған торында оңға қарай бұрыла алады, яғни {алға; оңға} деген нұсқауларды орындай алады. Бауырсақ бір тордан бірнеше рет өтуі мүмкін, бірақ  деп белгіленген бөгеуші тордан өте алмайды. Егер ол өзі тұрған тордан  деп белгіленген торға бірер жолмен бара алатын болса, қажетті нұсқаулар тізбегін құрастыр.



7-сабақ. Алгоритмнің бейнелену әдістері

Бұрынғы сабақтардағы алгоритмдер сөздер арқылы бейнеленген еді. Ескерте кету керек, алгоритмнің бейнелену әдістері әртүрлі болады, сондықтан біз олардың кең таралған түрімен танысамыз.

1. Алгоритмнің сөздер арқылы бейнеленуі.

Бұрынғы сабақтарда келтірілген бірнеше есептер адамның ауызекі тілінде қолданылатын сөздер арқылы бейнеленген еді (мәселен, шайды демдеу, қосындыны есептеу алгоритмі).

Алгоритмнің осылай бейнелену әдісіне сәйкес атқарушыға арналған нұсқау сөздер арқылы бұйрық түрінде беріледі.

Мысал ретінде **A** литрлік және **B** литрлік ыдысы бар, суатқа жақын жерде тұрған орындаушының {**A-ны толтыр; B-ны толтыр; A-дан B-ға құй; B-дан A-ға құй; A-ны босат; B-ны босат**} сияқты нұсқаулар жүйесін келтіруге болады. Осы орындаушыға қатысты есептің мақсаты — өлшеп алынатын су көлемін не *A*, не *B* ыдыстың жәрдемімен шығару.

1-есеп. **A=3** және **B=5** болса, «Сушының **1 литр** суды өлшеп алу» алгоритмін құр. Бұл есептің мақсатқа жеткізетін алгоритмін сөздер арқылы жасау қолайлы:

Қадамдар	Алгоритмдегі нұсқаулар	A ыдыста	B ыдыста
1	A-ны толтыр	3 литр	0 литр
2	A-дан B-ға құй	0 литр	3 литр
3	B-ны толтыр	3 литр	3 литр
4	A-дан B-ға құй	1 литр	5 литр

2. Алгоритмнің формулалар арқылы бейнеленуі.

Бұл әдіс математика, физика, биология сияқты пәндерді үйренген кезде пайдаланылады. Есінде болса, сөздер арқылы берілген 4-сабақтағы

2-есептің алгоритмін формула арқылы бейнелеген едік. Формуладағы «+», «-», «x», «:» сияқты арифметикалық амалдардың есептеу ережелерін сақтау шартымен орындалуы да алгоритмге мысал болады. 5-сабақта келтірілген « $ax^2 + bx + c = 0$ ($a \neq 0$) түріндегі квадрат тендеуді шешу» алгоритмі мына формула арқылы бейнеленгенін сен жақсы білесің:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

3. Алгоритмнің кесте арқылы бейнеленуі.

Алгоритмнің осындай бейнеленуі де саған таныс. Мәселен, мектептегі сабақ кестесі, Пифагордың көбейту кестесі, лоторея ұтыстар кестесі, Химиялық элементтер кестесі осыған мысал бола алады. Мұндай кестелердің пайдаланылуы белгілі бір алгоритм қолдануды талап етеді.

Бір функцияның графигін сызған кезде де функцияның аргументтер мәніне сәйкес мәндер кестесін пайдаланамыз. Бұл да алгоритмнің кестелік көрінісіне мысал болады.

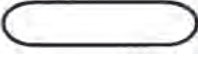



Мысалы, саған математика курсынан мәлім, $y = x^2$ алгоритмі бойынша қозғалып жатқан орындаушы мына кестеде көрсетілген нүктелерді басып өтеді:

x	-3	-2	-1	0	1	2	3
y	9	4	1	0	1	4	9

4. Алгоритмнің графикалық түрде бейнеленуі.

Алгоритмнің осылайша бейнеленуі саған математикадағы функцияның графигі, қажет үйді табу үшін массивтерде орнатылған үйлердің жайғасу схемасы, автобустардың жүру схемасы арқылы таныс.

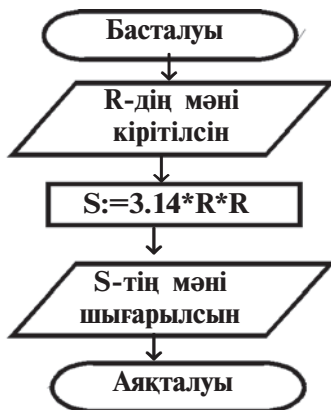
Алгоритмдеу негіздерін үйренудің тағы бір қолайлы графикалық тәсілі — **блок-схема** әдісі. Блок-схема бағыттаушы сызықтар арқылы тұтасқан, белгілі бұйрықты яки нұсқауды білдіретін арнаулы геометриялық фигуралар — **блок**тардан тұрады:

	алгоритмнің басталуын немесе аяқталуын білдіреді
	мәліметтерді енгізу және шығару белгісі
	жай амалды, яғни мән беру немесе тиісті нұсқауды орындау процесін білдіреді
	шартты тексеру белгісі

	кайталаудың басталуын білдіреді
	жәрдемші алгоритмге жүгінуді білдіреді
	схемадағы амалдың бағытын көрсетеді
$:=$	мән беру командасы

2-мысал. Радиусы R -ге тең шеңбердің ауданын есептеу алгоритмін құр. Бұл мәселенің алгоритмін екі әдіспен, яғни сөздер арқылы және графикалық көріністе жасауға болады:

- 1) басталсын;
- 2) R -дің мәні анықталсын;
- 3) R -ді R -ге және 3,14-ке көбейтіп, нәтижесі S деп белгіленсін;
- 4) жауап ретінде S жазылсын;
- 5) аяқталсын.



5. Алгоритмнің бағдарлама көрінісінде бейнеленуі.

Бәріне белгілі, компьютер бағдарлама бойынша істейді де басқарылады. Сен қазірше MS Word, MS Paint, MS Excel сияқты қолданбалы **бағдарламалармен** жұмыс істеп келдің. Ескерте кету керек, әрбір қолданбалы **бағдарлама** да өте үлкен әрі күрделі алгоритмнің бір түрі болып табылады. Демек, сол тәрізді алгоритмдер **алгоритм орындаушысына**, яғни **компьютерге түсінікті болуы керек**.

Әдетте, алгоритмнің компьютер түсінетін тілде жазылуы **бағдарлама** деп аталады. Ал компьютер түсінетін тіл **бағдарламау тілі** деп аталады. Дүниеде мыңдаған бағдарламалау тілдері бар және жетілдіріліп келуде. Қазіргі таңда **BASIC, Pascal, VBA, Delphi, C, C++** бағдарламалау тілдері кеңінен таралған әрі үйренуге қолайлы.



Сұрақтар мен тапсырмалар

1. Алгоритмнің бейнелену әдістері жайлы мағлұмат бер.
2. Алгоритмнің сөздермен бейнеленуіне өмірден мысалдар келтір.
3. Қай пәндерде алгоритмді формулалар жәрдемімен беру қолайлы?
4. Алгоритмнің формуламен бейнеленуіне физикадан мысал келтір.

5. Алгоритмнің кестелік түрде берілуіне мысалдар келтір.
6. Алгоритмнің графикалық түрде бейнеленуіне мысалдар келтір.
7. Блок-схема деген не?

Жаттығулар

1. MS Paint графикалық редакторын пайдаланып, «**Өзбекстан – келешегі ұлы мемлекет!**» мәтініндегі сөздерді бірінің астына екіншісін келтіріп жазатын алгоритмді сөздер арқылы құр.

2. MS Word бағдарламасының WordArt объектін пайдаланып, «**Өзбекстан – Отаным менің!**» сөз тіркесінің жазылу алгоритмін қолайлы әдіспен бейнеле.

3. Берілген екі натуралды санның ең кіші ортақ еселігін (ЕКӨЕ) табу алгоритмін құр.

8-сабақ. Алгоритмнің бейнелену әдістері бойынша тақырыпқа арналған практикалық жұмыс

1. Төмендегі есептердің алгоритмдерін сөздер арқылы құр.

А. x -тің мәні белгілі болса, $y = 23 \cdot x - 1963$ функциясының мәнін есептейтін алгоритм құр.

Ә. Орындаушының нұсқаулар жүйесі тек **{5-ті қос; 3-ті айыр}** командалардан тұрады. Осы орындаушы 0 санынан 11 санын шығару үшін алгоритм құр.

Б. Орындаушының нұсқаулар жүйесі тек **{1-ді қос; 2-ге көбейт}** нұсқаулардан тұрады. 0 санынан 17 санын шығару үшін, осы орындаушының алгоритмін үш әдіспен құр.

В. $A=5$ және $B=8$ болса, орындаушы 4 литр су өлшеп алуы үшін алгоритм құр.

Бағыз: Ә және Б есептердің шешімін шығарған кезде мынадай көрінісі бар кесте жасау – алгоритм қадамдарын орындаудың нәтижесін көріп тұруға мүмкіндік береді:

Қадам	Нұсқау	Нәтиже
0	–	0
1		
2		

2. Төмендегі есептердің алгоритмдерін блок-схема арқылы құр.

А. Радиусы R -ге тең шеңберге жазылған квадраттың қабырғасын табу алгоритмін құр.

Ә. Үш тиын берілген. Олардың бірі жасанды әрі ауыр. Өлшеу үшін екі табағы бар таразы тастарынсыз берілген. Жасанды тиынды табу алгоритмін құр.

Б. Үш тиын берілген. Олардың бірі жалған, сондай-ақ тек салмағымен ерекшеленеді (оның шынында да ауыр я жеңіл екені мәлім емес). Өлшеу үшін екі табағы бар таразы тастарынсыз берілген. Ең аз тарту жолымен жасанды тиынды табу алгоритмін құр.

9-сабақ. Алгоритмнің негізгі түрлері

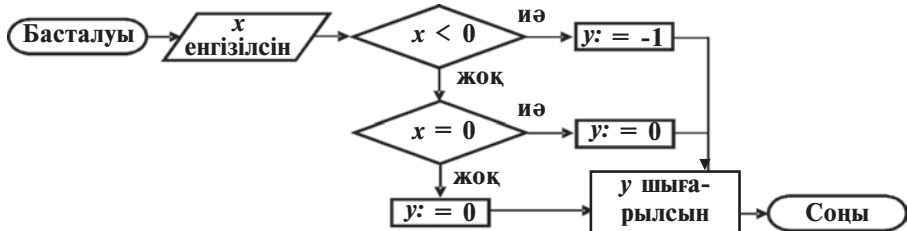
Кез келген алгоритм логикалық құрылысына, яғни орындалу тәртібіне қарай негізгі үш: **сызықтық, тармақталатын және қайталанушы** түрге бөлінеді.

Сызықтық алгоритмдер. Барша нұсқаулары тек тізбекпен, реттік нөміріне қарай орындалатын алгоритмдер **сызықтық алгоритмдер** деп аталады. «Шай демдеу», шеңбердің ауданын есептеу алгоритмдері сызықтық алгоритмдерге мысал болады. Бірақ күнделікті тұрмысымыздағы көптеген процестер шарттармен ғана басқарылады.

Тармақталатын алгоритмдер. Шарт бойынша орындалатын нұсқаулары бар алгоритмдер **тармақталатын алгоритмдер** деп аталады. Алгоритмдердің бұл түрі өмірімізде күн сайын, әр қадамда кездеседі. Есіктен шығарда есіктің ашық не жабық екеніне, тамақтанғанда қарнымыздың аш не тоқ екеніне, әлде тамақтың түріне, көшеге киініп шыққанда – ауа райына, бір жерге бармақ болып, көлік құралдарын таңдауда – төлеуге мұршамыз бар ақшаға байланысты. Демек, тармақталатын алгоритмдер сызықтық алгоритмдерден таңдау мүмкіншілігімен ерекшеленеді екен. Бұрынғы сабақтардағы квадрат теңдеуді шешу, екі санның ЕҮОБ-ін табу сияқты алгоритмдер тармақталатын алгоритмдерге мысал болады.

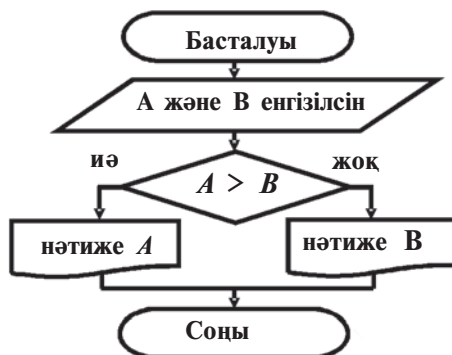
1-мысал. Алгоритм $y = \begin{cases} -1, & \text{егер } x < 0 \\ 0, & \text{егер } x = 0 \\ 1, & \text{егер } x > 0 \end{cases}$ формула арқылы берілген. Функ-

цияның мәнін есептеп шығаруға дейін тармақталған алгоритм блок-схеманың жәрдемімен былай бейнеленеді:



2-мысал. A мен B екі сан берілген. Олардың қайсысы үлкен екенін анықтайтын алгоритм құр.

- 1) басталсын;
- 2) A және B енгізілсін;
- 3) егер $A > B$ болса
4-пункте өтілсін,
5-пункте өтілсін;
- 4) нәтиже A деп алынсын және 6-пунктке өтілсін;
- 5) нәтиже B деп алынсын және 6-пунктке өтілсін;
- 6) аяқталсын.



Бұл есеппен мынадай қорытынды шығаруға болады: егер $A > B$ шарты орындалса, 5-пунктегі нұсқау орындалмайды, ондай болмайтын болса, яғни $A \leq B$ болса, онда 4-пунктегі нұсқау орындалмайды. Бұл алгоритм тармактануды айқын елестетуге мүмкіндік береді.

Қайталанушы (циклді) алгоритмдер. Есептерді талдаған кезде алгоритмдегі кейбір нұсқаулардың қайта-қайта орындалғанын байқауға болады. Мәселен, ең үлкен квадраттарды кесіп алу есебі (4-сабақ, 5-есеп), Эвклид алгоритмі (5-сабақ, 2-есеп) осыған мысал бола алады. Біздің тұрмысымызда да көптеген процестер қайталанады. Мысалы, сабақтардың апта сайынғы қайталануы, таңертеңгі ас ішу, мектепке әр күні бару және т.б. Нұсқаулары қайта-қайта орындалатын алгоритмдер **қайталанушы алгоритмдер** деп аталады.

Қайталанушы алгоритмдер « $I := I + 1$ », « $S := S + I$ » яки « $P := P * I$ » сияқты қатысқан нұсқауларымен көзге түседі (мұндағы * – көбейту амалы). Осындай нұсқаулардың мазмұнын түсіну үшін қайталанудың бірнеше қадамын қарап шығуға тура келеді.

Әдетте, қосынды санның бастапқы мәні $S := 0$ (ағылшынның SUM, яғни қосынды мағыналы сөзінің бас әрпі) деп, ал көбейтіндінің бірінші мәні $P := 1$ (ағылшын тілінен PRODUCT, яғни көбейтінді деген мағынаны білдіреді) деп алынады, өйткені бұл мәндер, яғни 0 мен 1 сәйкесінше қосынды мен көбейтіндінің қорытынды санына әсер етпейді:

1-қадам: $I := 1$ болсын, онда $S := S + I = 0 + 1 = 1$, $P := P * I = 1 * 1 = 1$;

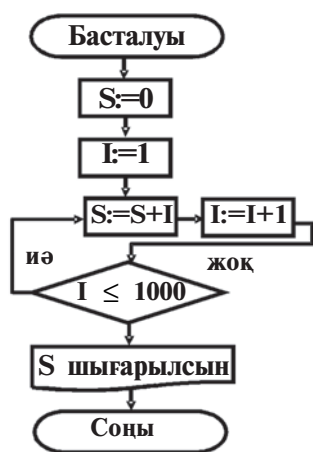
2-қадам: $I := I + 1 = 1 + 1 = 2$, $S := S + I = 1 + 2 = 3$, $P := P * I = 1 * 2 = 2$;

3-қадам: $I := I + 1 = 2 + 1 = 3$, $S := S + I = 3 + 3 = 6$, $P := P * I = 2 * 3 = 6$;

4-қадам: $I := I + 1 = 3 + 1 = 4$, $S := S + I = 6 + 4 = 10$, $P := P * I = 6 * 4 = 24$.

3-мысал. 1-ден 1000-ға дейінгі сандардың қосындысын, яғни $S=1+2+3+\dots+1000$ -ды есептейтін алгоритм құр.

- 1) басталсын;
- 2) $S=0$ деп алынсын (яғни $S:=0$);
- 3) I -дің мәні 1 деп алынсын (яғни $I:=1$);
- 4) S -ке I қосылып, нәтиже S деп алынсын (яғни $S:=S+I$);
- 5) I -ге 1 қосылып, нәтиже I деп алынсын (яғни $I:=I+1$);
- 6) егер $I \leq 1000$ болса, 4-пунктке өтілсін;
- 7) жауап S деп алынсын;
- 8) аяқталсын.

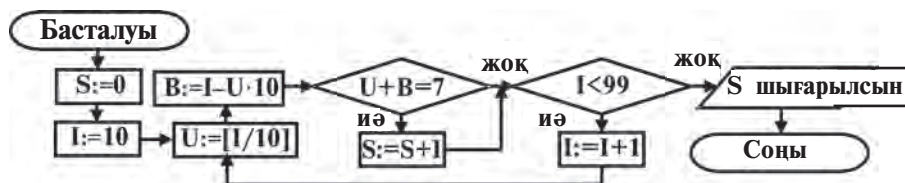


Сөздер арқылы бейнеленген алгоритмнің блок-схемамен үйлесімділігін көрсету үшін біз жақшалар ішіне түсініктемелер бердік. Әдетте қайталанушы алгоритмдегі « $I:=I+1$ » сияқты өрнек **санауыш** деп аталады. Бұл мысалдың шешімін сызықтық алгоритм арқылы ұйымдастыруға болады. Бұл үшін кез келген N натуралды сандарға орынды $1+2+3+\dots+N \equiv N \cdot (N+1) : 2$ теңбе-теңдігін пайдалану жеткілікті (осы алгоритмді өз бетінмен құр).

Төмендегі мысалдардың шешімін шығару біраз қиынға соғады.

4-мысал. Екі разрядты сандар арасынан цифрлар қосындысы 7-ге тең болған сандардың қосындысын есептейтін алгоритм құр ($[a]$ – a санының бүтін бөлігі, / – бөлу амалы).

5-мысал. «Отан – сәждеге бас иетін орындай қасиетті!» деген сөйлемді 20 рет жаздыру алгоритмін құр. Бұл мысалдың алгоритмін сөздер арқылы бейнелейміз:



- 1) I -нің мәні 1 деп алынсын;
- 2) «Отан – сәждеге бас иетін орындай қасиетті» жазылсын;
- 3) I -ге 1-ді қосып, ол I деп алынсын;
- 4) егер $I \leq 20$ болса, онда 2-пунктке өтілсін;
- 5) аяқталсын.

Карастырылған алгоритмдерге назар аударсақ, олар сызықтық, тармақталатын яки қайталанушы бөлімдерден тұратынын көруге болады. Демек, тұрмыста кездесетін алгоритмдер негізінде осы үш түрдегі алгоритмдердің ажырағысыз бірлігі болып келеді.



Сұрақтар мен тапсырмалар

1. Қандай алгоритмдерді сызықтық алгоритм деп атайды? Мысалдар келтір.
2. Қандай алгоритмді тармақталатын алгоритм деп атайды? Мысалдар келтір.
3. Қандай алгоритмді қайталанушы алгоритм деп атайды? Мысалдар келтір.
4. Сызықтық, тармақталатын және қайталанушы алгоритмдердің айырмашылығын түсіндір.
5. Берілген үш санның қайсысы үлкен екенін анықтайтын алгоритм (ИКТ алгоритмін) құр.

Жаттығулар

1. Төмендегі алгоритмдер алгоритмнің қай түріне мысал болатынын және олардың нәтижесін анықта:

а) $a:=3$; $x:=2*a+a*a$. $a=?$, $x=?$

ә) $x:=1$; $x:=x+11$, $x:=x*x-4$. $x=?$

б) $a:=15$; $b:=a$; $a:=a-b$. $a=?$, $b=?$

в) 1) $a:=3$;

2) егер $a > 2$ болса, онда $x:=2*a+a*a$ және 4-пунктке өтілсін, ондай болмаса, 3-пунктке өтілсін;

3) $x:=9-a*x$;

4) нәтиже x жазылсын;

5) аяқталсын.

г) 1) $x:=1$;

2) егер $x > 2$ болса, онда $x:=x+11$ және 4-пунктке өтілсін, ондай болмаса, 3-пунктке өтілсін;

3) $x:=x*x-4$;

4) нәтиже x жазылсын;

5) аяқталсын.

д) 1) $a:=15$;

2) $b:=a$;

3) егер $a > b$ болса, онда $a:=a-b$ және 5-пунктке өткізілсін, ондай болмаса, 4-пунктке өтілсін;

4) $a:=a+b$;

5) нәтиже a , b жазылсын; б) аяқталсын.

2. Берілген санның таңбасын анықтайтын алгоритмді блок-схема көрінісінде құр.

3. $y = x^2 - 1$ функциясының мәндерін x -тің $[1; 10]$ аралықтағы бүтін мәндерінде есептейтін алгоритмді блок-схема түрінде құр.

10-сабақ. Алгоритмнің негізгі құрылымдарына қатысты практикалық тапсырма

Жоғарыда айтылғанындай, кез келген алгоритмді сызықтық, тармақталатын және қайталанушы алгоритмдердің өзара үйлескен тіркесіндей етіп бейнелеуге болады. Сондықтан төменде келтірілген құрылымдарды жақсы ұғып алсаң, мақсатқа лайық болады. Бұл құрылымдардың маңызды жағы сол, олардың бір кірісі мен бір шығысы бар.

1. Сызықтық құрылым. Тізбектеп келетін жай әрекетті, яғни мән немесе тиісті нұсқаулар беруден ғана құрылады.

Сөздер арқылы	Блок-схема көрінісінде
жай әрекет 1	
жай әрекет 2	
...	
жай әрекет N	

1-мысал. a, b, c – үш сан берілген. a мен b сандар қосындысының жартысын, a мен c сандар айырмасының мәнін, b мен c сандар көбейтіндісінің квадратын есептейтін алгоритм құр.

2-мысал. Радиусы R -ге тең шеңбердің ұзындығы мен дөңгелек ауданын, сондай-ақ шар көлемін есептейтін алгоритм құр (бағыт: $L=2\pi R$; $S=\pi R^2$;

$$V = \frac{4}{3} \pi R^3).$$

2. Тармақталған құрылымдар. Бұл құрылымдар шартты тексеру нәтижесіне (иә не жоқ) қарай екі жолдың біреуін ғана тандап алуға мүмкіндік береді. Бұл құрылымдар негізінде 2 түрлі көріністе болады:

а) егер – онда көрінісі:

Сөздер арқылы	Блок-схема көрінісінде
егер шарт онда нұсқаулар тобы соңы	

3-мысал. Егер берілген a сан оң болса, онда оның квадратын және квадрат түбірін шығаратын алгоритм құр.

4-мысал. Үш — a, b, c сандар берілген. Олардың арасындағы теріс болмаған сандардың квадрат түбірін шығаратын алгоритм құр.

б) егер — онда — ондай болмаса көрінісі:

Сөздер арқылы	Блок-схема көрінісінде
<p>егер шарт онда нұсқаулар тобы 1 ондай болмаса нұсқаулар тобы 2 соңы</p>	

5-мысал. Берілгені — a, b, c сандары. Егер $a < b - c$ шарты орындалса, «Иә» деген, ондай болмаса, «Жоқ» деген жауап шығаратын алгоритм құр.

6-мысал. a мен b сандар берілген. Егер олардың көбейтіндісі оң сан болса, онда олардың әрқайсысының квадратын, ондай болмаса, әрбіріне 100-ді қосып шығаратын алгоритм құр.

3. Қайталанушы құрылымдар. Бұл құрылымдар бірнеше нұсқаулар тобын қайта-қайта орындауға жағдай жасайды. Бұл құрылымдар да негізінде екі түрлі көріністе болады:

а) дейін:

Сөздер арқылы	Блок-схема көрінісінде
<p>шарт дейін нұсқаулар тобы соңы</p>	

7-мысал. Натуралды x -тің мәні берілген a санынан кіші болғанда $y = ax^2 + 20$ функциясының мәндерін есептейтін алгоритм құр.

8-мысал. Берілген A мен B оң сандардың мәндері тең болғанша, осы сандардың үлкенінен кішісін алып, үлкенімен ауыстырып келетін алгоритм құр.

б) параметр ... ден ... дейін:

Сөздер арқылы	Блок-схема көрінісінде
<p>Параметр В-дан О-ға дейін нұсқаулар тобы соңы</p>	

Бұл жерде параметр санауыш болып саналады, сондықтан ол B мен O -ға карағанда, қалағанынша әріп болуы мүмкін.

9-мысал. «Өзбекстан – келешегі ұлы мемлекет!» сөз тіркесін, осы оқу жылында Отанымыздың тәуелсіздігі неше рет атап өтілген болса, сол санға дейін жаздыру алгоритмін құр.

Енді жоғарыда келтірілген құрылымдардың жәрдемімен бұрынғы сабақтарда берілген мысалдарды оп-оңай бейнелеуге болады.



Сұрақтар мен тапсырамалар

1. *Сызықтық алгоритмге сай келетін құрылымға мысалдар келтір.*
2. *Тармақталатын құрылымдардың қандай түрлері бар?*
3. *Тармақталған құрылымдардың пайдаланыстағы қолайлылығына мысалдар келтір.*
4. *Қайталанушы құрылымдар жөнінде мәлімет бер.*
5. *Қандай есептердегі «дейін» құрылымын пайдалану мақсатқа сай болады?*

Жаттығулар

1. v км/сағат жылдамдықпен қозғалып келетін машинаның T сағат басып өткен жолын есептейтін алгоритм құр.
2. Радиустары сәйкесінше R_1 , R_2 , R_3 болған шеңберлердің жалпы ауданының квадратын есептеп шығаратын алгоритм құр.
3. Қабырғалары сәйкесінше a мен b болған квадраттардың аудандары айырмашылығының модулін табатын алгоритм құр.
4. a мен b екі сан берілген. Егер b сан a -дан кіші болса, онда b -ні нөлмен ауыстыратын, ондай болмаса, b -ні өзгеріссіз қалдыратын алгоритм құр.



11-сабақ. Қайталауға арналған тапсырмалар

1. Есептерді компьютерде шығарудың бірінші үш кезеңін, қабырғасы a -ға тең болған квадратқа ішкі сызылған шеңбердің ауданын табу жөніндегі есепке сүйеніп көрсетіп бер.
2. Есептерді компьютерде шешудің алғашқы үш кезеңін есептің шешіміне сүйеніп көрсет: 50 литрлік ыдыста 5 килограмм ас тұзы қосылған 20 литр араласпа бар. Егер ыдысқа тағы да 10 литр су қосса, араласпадағы тұздың мөлшерін пайыз есебімен тап.
3. Берілгені – a , b , c сандар. Егер $a+b+c < 0$ шарты орындалса, онда $y = a^2 - b^2$ мәнін, ал ондай болмаса, $y = a^2 + c^2$ мәнін есептейтін алгоритм құр.
4. – 100-ден 50-ге дейінгі сандар аралығындағы тақ сандардың көбейтіндісін есептейтін алгоритм құр.

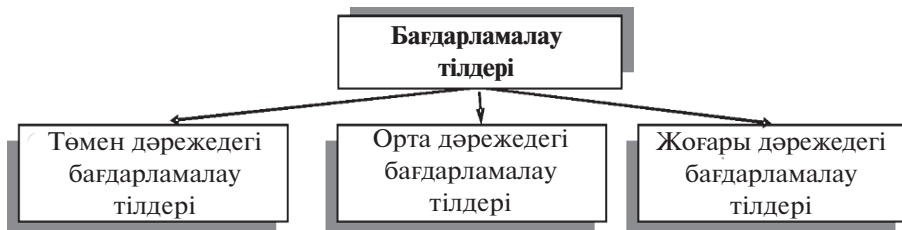
12-сабақ. Бағдарлама және бағдарламалау тілдері

Белгілі болғанындай, компьютер техникасын тиімді пайдалану, екі бөлім — техникалық және бағдарламалық қамсыздандырудың тығыз байланыстылығын талап етеді. Бұл байланыс компьютер техникалық қамсыздандырудың жедел қарқынмен жетілуіне байланысты бағдарламалық қамсыздандырудың да тез қарқынмен дамуына себеп болады және керісінше. Мұның себебі мәлім, тиісті бағдарламалық қамсыздандыруы жоқ әр компьютер бір «қымбат ойыншық» болып қала береді.

Өткен сабақтардағы есептерді компьютерде шығарған кезде сен объект, модель, алгоритм деген қажетті ұғымдар туралы мәлімет алдың. Бізге мәлім, компьютерде белгілі бір есепті шығару үшін алдымен оның моделі мен алгоритмі құрылады, кейін осы алгоритм белгілі бір заң-ережелер негізінде компьютер түсінетін тілде нұсқаулар мен командалар түрінде жазылады. Компьютер түсінетін тілде шығарылған мәтін **бағдарлама мәтіні**, ал алгоритм **бағдарлама** деп аталады.

Компьютерге арналған бағдарлама жасау процесі **бағдарламалау** деп, бағдарламаны жасаушы адам **бағдарламашы** деп аталады. Ал компьютер түсінетін тіл **бағдарламалау тілі** деп аталады.

Бағдарламалау тілдерін шартты үш топқа бөлуге болады:



Төменгі дәрежедегі бағдарламалау тілдері компьютер қондырғыларымен тікелей байланысты болып, нұсқаулар арнайы цифрлар (кодтар) жәрдемімен жазылады. Осы сияқты нұсқаулардан құрастырылған бағдарламалардың көлемі үлкен, оларды редакциялау да өте қиын болады. Тұңғыш электронды-есептеуіш

машиналарда («ENIAC», «МЭСМ» және т.б.) есептер шығару үшін осындай нұсқаулардың жәрдемімен бағдарламалар құрас-тырылған.

Бағдарламалау тілдерінің тарихы. Бағдарламалау тілдері, негізінен, екінші дүниежүзілік соғыстан кейін жасала бастады. Бірақ оның басталу тарихы едәуір көп жылдарға барып тіреледі.

Археологиялық қазбалардан табылған қыш тақтайшада, осыдан 3800 жыл бұрын (б.ж.с. дейінгі 1800-жылдары) Вавилондағы пайызбен байланыс-ты күрделі амалдар алгоритмі бейнеленген. Онда анық есеп шығарылған, яғни егер бидай өнімі жылына 20 пайызға артып отырса, оның мөлшері екі есе өсуі үшін неше жыл, неше ай керек болатынын көрсететін алгоритм құрылған.



Чарльз
Баббидж

XIX ғасырдың француз өнертапқышы **Жозеф Мари Жаккард** 1804 жылы жұқа мата тоқу барысында тоқу станогында перфокартаға ұқсас таспаны қолданып, осылайша перфокар-таның негізін салған еді.

1836 жылы ағылшын ғалымы **Чарльз Баббидж** қазіргі за-манғы компьютерлердің тікелей ұрпағы саналатын аналити-калық машина жасауға кірісіп, *бұл мәселені теориялық тұрғыдан шешті*. Сол машинаның басты ерекшелігі — оның бағдарлама бойынша жұмыс істеуі және есеп-қисап нәтижелерін «есте» сақтап қалуы еді.

1843 жылы ағылшын математигі **Огаста Ада Байрон** (Лав-лейс) — ақын Лорд Байронның қызы аналитикалық машина нұсқаулар бойынша істеу керек деп айтты. Ол берілген шарт-тар орындалғанша адымдардың тізбегін (циклді) қамтамасыз етуші командаларды жазды. Дәл осылай ол бағдарламалау тілін ойлап тапты. Осы және басқа да жаңалықтар, компью-терлер пайда болған соң, оларды істетуге қажетті тілдің жа-ратылуын талап етті.



Ада Лавлейс

Бағдарлама жасауды оңайлату үшін, адам тіліне жақын нұсқаулар жүйесін қолдану мәселесін қойды да, ол мәселе өз шешімін тапты. Осы сияқты бағдарламалау тілдері **орта дәрежедегі бағдарламалау тілдері** (кейде ассамблерлер) деп айтыла бастады. Мұндай тілдерге **AVTOKOD-BEMSH, AVTOKOD-MADLEN** және басқалар жатады. Олар **ВЭСМ-6, Минск-22, Минск-32, IBM-360** электронды-есептеу машиналарында қолданылады. Мысалы, **ST 5, BSUM** өрнектегі 5 цифры BSUM деп аталған ұяшыққа орналастыру (**ST-store-орналастыру**) деген команда-ны береді.

Жоғары дәрежедегі бағдарламалау тілдеріндегі командалар адамның тіліне жақын сөздер кешенінен тұрады. Олардың көмегімен амалдарды төменгі дәрежедегі тілдерден гөрі оңай орындауға болады, өйткені бағдарламашы-дан айтарлықтай адресстер мен кондырғыларға тікелей қатысы бар ақпарат-

тарды білу талап етілмейді. Компьютерлер осындай тілде құрылған бағдарламаларды орындай алуы үшін, **трансляторлар** деп аталатын арнайы бағдарлама оларды цифрлы көрініске өткізіп береді.

Соңғы жылдары өте көп жоғары деңгейдегі бағдарламалау тілдері құрастырылған. Солардың қатарына **Pascal, Ada, KARAT, C++, Delphi, Visual Basic Application, Java** сияқты тілдерді қосуға болады. Қазіргі таңда жасалып отырған бағдарламалау тілдері белгілі бір бағыттағы есептерді шешуге арналған, оларды **объектіге бағытталған бағдарламалау** тілдері деп атайды.

Төмендегі кестеде бағдарламалау тілінің даму тарихынан мәлімет берілген.

Бағдарламалау тілі	Жасалған жылы	Бағдарламалау тілі	Жасалған жылы
Plankalkyul	1946	Logo	1967
Қысқа код	1949	Algol 68	1968
Ассемблер «Edsak», АО	1950	APL	1969
Автокод«Madlen»	1953	Paskal	1970
Жедел кодтау	1955	Fort	1971
A-2,Flou-metik	1956	Prolog,Ci,Ada	1972
IPL-1,Mat-metik	1957	Smoltok	1980
Фортран	1958	VBA	1990
Algol 58	1959	VC++	1993
APT,LISP,Kobol,Algol-60	1960	Java	1994
PL/I,Basic	1964	Delphi	1995
Algol W	1965	C#	2000

Осы күнге дейін құрастырылған бағдарламалау тілдерінің кең таралғаны — **Паскаль** (Pascal — ағыл.тілі) бағдарламалау тілі. Паскаль бағдарламалау тілін 1969 жылы Никалас Вирт жасады. Паскаль тілі бағдарламалауға үйрету үшін жасалған болса да, жоғары маманды бағдарламашылар арасында кеңінен таралды. Әрине, тиімді бағдарламалау тілдері өзгермей қалмайды. Сондықтан да түрлі үлгідегі компьютерлер үшін Паскаль тілінің оларға бейімделген нұсқалары құрастырылған, олар Паскаль тілінің түпкі нұсқасынан ерекшеленуі мүмкін.



Сұрақтар мен тапсырмалар

1. Бағдарлама деп нені айтады?
2. Бағдарламалау тілі дегенде нені түсінесің?
3. Түрлі дәрежелі тілдер бір-бірінен қандай қасиеттерімен ерекшеленеді?

4. Бағдарламалау тілдері электронды-есептеуіш машиналардың түріне байланысты бола ма? Жауабыңды дәлелде.
5. Бірнеше жоғары дәрежедегі бағдарламалау тілдерін олардың жасалған жылымен қоса айтып бер.

13-сабақ. Турбо Паскаль 7.0 интегралданған ортасы

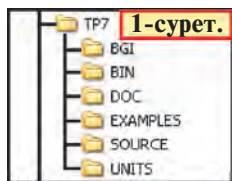
Паскаль бағдарламалау тілінің кең таралуы мен қолданылуының басты себебі — бағдарламаның ықшамдылығы мен қолданыстағы қолайлылығы. Алғашқыда Паскаль тілі университеттерде қолданылған, кейінірек оның түрлі үлгідегі компьютерлерге арналған нұсқалары құрастырылды.

1981 жылы Паскаль тілінің халықаралық стандарты ұсынылды. Борланд фирмасы құрастырған Паскаль тілінің **Турбо Паскаль 7.0** нұсқасы қазіргі таңда кеңінен пайдаланылады. Онда пайдаланушыларға өте қолайлы жүйе — **бағдарламалаудың интегралданған ортасы бар**.

Интегралданған орта — бағдарламауға көмектесетін бағдарлама, ол мынадай басты міндеттерді орындау керек:

- ең алдымен, бағдарлама мәтінін енгізуге мүмкіндік беру;
- жиі-жиі енгізілетін бағдарлама мәтінін сыртқы жадқа сақтау;
- бағдарламаны іске қосу үшін трансляторға ие болу;
- синтаксистік қателіктерді анықтау құралына ие болу.

Турбо Паскаль 7.0 интегралданған ортасы санап өтілген міндеттерден басқа, әлі де көп міндеттерді де жүзеге асыруға мүмкіндік береді.



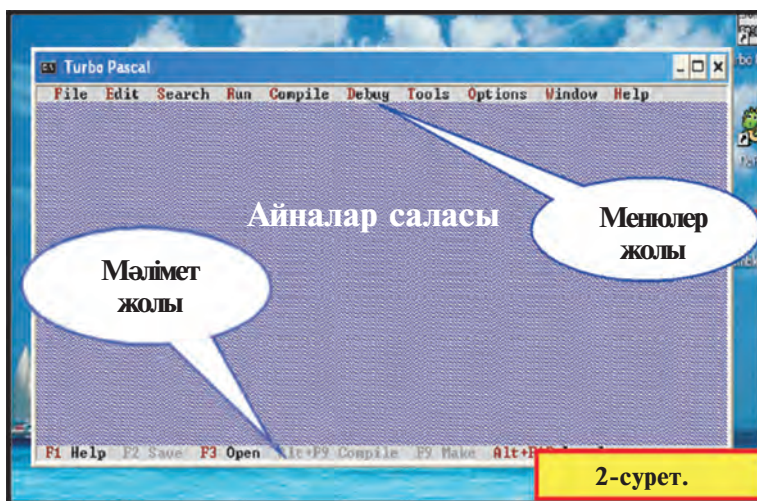
Turbo Pascal 7.0 бағдарламалау жүйесі сыртқы жадтағы, әдетте, «TP7» атты каталогқа орналастырылады (жалпы жағдайда басқа каталогқа да орналастыруға болады). Ол жүзден астам файлдан тұрады, файлдар міндеттеріне қарай бірнеше каталогтарға орналастырылған (1-сурет).

Турбо Паскаль интегралданған ортасын іске қосатын **Turbo.exe** файлы «BIN» каталогында орналасқан. Қалған каталогтарда негізінде көмекші файлдар, сондай-ақ Турбо Паскаль тілінің мүмкіндіктерін көрсететін бағдарламалар орналасады.

Мәселен, «BGI» каталогында графикалық жағдайда істеу үшін қажетті файлдар орналасқан.

Турбо Паскальдің қалған каталогтары мен файлдары туралы мәліметті оқулықта келтірілген қосымша әдебиеттер тізіміндегі қолданбалардан керегінше алуға болады.

Turbo.exe файлын іске қосқан соң, экранда Турбо Паскаль интегралданған ортасының интерфейсі ашылады. Ол **меню жолынан, терезелер саласы мен деректер жолынан** тұрады (2-сурет). Турбо Паскаль тілінің бірнеше түрдегі терезелері бар, олардың ең маңыздысы — **бағдарлама мәтіні редакторының** терезесі. Бұл терезені шығару үшін **File** менюсінің **New** (Жана) командасын тандаса жеткілікті.



Меню жолына көшу үшін **[F10]** клавишін басады. Содан кейін солға я оңға қарай бағыттаушы клавиштер арқылы менюді тандап, **[ENTER]** клавишін басады. Қажетті менюді «жүгірткінің» жәрдемімен тандауға болады.

Турбо Паскаль интегралданған ортасының интерфейсі мәтін редакторының интерфейсіне ұқсайды. Оның бағдарлама мәтіні бағдарлама редакторының терезесіне дәл мәтін редакторындағыдай жазылады. MS Word сияқты, Турбо Паскаль да бірнеше терезе ашып, олардың әрқайсысымен арнайы жұмыс істеуге мүмкіндік береді. Бұл бір уақытта бірнеше бағдарламалармен жұмыс істеуге жағдай жасайды. Ағымдағы уақытта істеп тұрған терезе **актив терезе** дейіледі. Бағдарлама мәтіні редакторындағы мәтіннің аты **NONAME00.PAS** етіп ұсынылады да, мәтіннің аты **атаусыз00** және файл кеңейтуі **pas** екенін білдіреді.

Файл (Fayl) менюсі Open (Ашу – сыртқы жадтағы файлды жедел жатқа жүктеу), Save (Сақтау), Save as (... сияқты сақтау), Exit (шығу), **Edit (Редакциялау)** менюсі Cut (қырқып алу), Copy (көшіру), Paste (ендіру), **Run** менюсі бағдарламаны іске қосу, **Compile** менюсі бағдарламаны компиляциялау (бағдарламаны «машина тіліне» аударып, «EXE» кеңейтуі бар файл көрінісінде сақтау) амалдарын қамтиды.

Меню құрамындағы амалдарды белгілі (**шапшаң әрекетті**) клавиштерді басу арқылы да орындауға болады. Төмендегі кесте негізгі амалдарды орындауға арналған клавиштер тізімі келтірілген:

F3	ашу	Ctrl+F9	бағдарламаны іске түсіру
F2	сақтау	Alt+F5	бағдарламаның экранда көрінісі
Alt+F3	актив терезені жабу	Alt+F9	бағдарламаны компиляциялау
Alt + x	шығу	F6	бір терезеден екінші терезеге өту

Бағдарлама мәтінін редакциялауда мына клавиштерді пайдалануға болады:

Бағыттаушы клавиштер (←, →, ↑, ↓) – жүргішті (курсор) керекті бағытқа жылжыту;

Shift + (←, →, ↑, ↓) – бағдарлама мәтінінің бөлігін жүргіш тұрған жерден бастап таңдалған бағытта белгілеу;

Ctrl+Insert – бағдарлама мәтінінің белгіленген бөлігінің көшірмесін буфер-жатқа алу;

Shift+Insert – жатқа сақтап алынған бөлікті бағдарлама мәтінінің жүргіш тұрған жеріне жайғастыру;

Shift+Delete – бағдарлама мәтінінің белгіленген бөлігін қырқып алу.



Сұрақтар мен тапсырмалар

1. Турбо Паскальді іске қосатын файл қайсы каталогта орналасқан?
2. Бағдарламалаудың интегралданған ортасы деген не?
3. Турбо Паскаль интегралданған ортасындағы бағдарлама мәтінінің редакторын аш.
4. NONAME.PAS деген нені білдіреді? Қайсы бағдарламада осы атауға ұқсас ат ұсынылады?
5. Турбо Паскаль жұмыс алаңындағы бағдарламаны іске қосу үшін қайсы клавиштерді басады?

Жаттыулар

Паскаль бағдарлама мәтінінің редакторын ашып, мына жұмыстарды орында:

- а) Өзбекстан Республикасы әнұранының бірінші өлең шумағын енгіз;
- ә) жаңа терезе ашып, оған республикамыз әнұранының екінші өлең шумағын енгіз;
- б) 2-терезедегі мәтіннің (әнұранның 2-өлең шумағының) көшірмесін алып, 1-терезедегі әнұранның 1-өлең шумағына қосып орналастыр;
- в) 1-терезедегі мәтінді «Әнұран.txt» деген атымен сақта;
- г) 2-терезені сақтамай жап.

14-сабақ. Паскаль бағдарламалау тілінің әліпбиі мен құрылысы

Кез келген бағдарламау тілі сияқты, Турбо Паскаль бағдарламалау тілінің де өз әліпбиі мен синтаксистік ережелері бар.

Турбо Паскаль тілі ASCII кодына жататын белгілер жинағын қамтиды, бұлар:

Латын әліпбиінің 26 бас және кіші әріптері: Aa, Bb, Cc, Dd, Ee, Ff, Gg, Hh, Ii, Jj, Kk, Ll, Mm, Nn, Oo, Pp, Qq, Rr, Ss, Tt, Uu, Vv, Ww, Xx, Yy, Zz (түсіндірмелер мен мәтіндер жазу үшін кирилл әріптерін де қолдануға болады);

Он араб цифры: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9;

Он алтылық цифрлар: 0-ден 9-ға дейінгі араб цифрлары, A, B, C, D, E, F және a, b, c, d, e, f әріптері;

Арнайы белгілер: "." (нүкте), "," (үтір), ":" (қос нүкте); ";" (нүктелі үтір), "'" (апостроф), "''" (тырнақша), "!" (леп белгісі), "?" (сұрақ), "%" (пайыз), "\$" (доллар), "@" (коммерция белгісі), "&" (амперсанд), "#" (тор), "^" (ырғақ беру); түрлі жақшалар: (,), {, }, [,]; жұп белгілер: :=, .., (*, *), (., .).

Басқару белгілері: #0-ден #31-ге дейін кодтандырылған белгілер (# – белгі кодының ондық санақ жүйесіндегі мәнін білдіреді, басқару белгілері істегенде экранда көрінбейді).

Турбо Паскаль тілінде негізінде мынадай амалдар мен оларға сәйкес белгілер қолданылады:

Арифметикалық амалдар: "+" (қосу), "-" (азайту), "*" (көбейту), "/" (бөлу);

Қатынас амалдары: "=" (тең), "<" (кіші), ">" (үлкен); жұп белгілер: "<>" (тең емес), "<=" (үлкен емес), ">=" (кіші емес);

Логикалық амалдар:

AND («ЖӘНЕ» – логикалық көбейту амалы)	OR («НЕМЕСЕ» – логикалық қосу амалы)
NOT («ЕМЕС» – логикалық жою амалы)	XOR (ұқсастықты жою амалы)

Барша бағдарламалау тілдері сияқты, Паскаль бағдарламалау тілінің де өз жазу емлесі, зандылықтары мен ережелері бар, соларға сүйеніп, жоғарыда келтірілген әріптердің, белгілер мен амалдардың жәрдемімен нұсқаулар, командалар құрастырылады. Әрбір нұсқау я команда ";" (нүктелі үтір) белгісімен аяқталады. Бағдарлама мәтінінде бір жолға ең көп 127 белгі жазуға болады.

Бағдарлама түсінікті болуы үшін көбінесе оған түсіндірмелер енгізіледі. Түсіндірменің жәрдемімен бағдарлама мен оның бөлімдері қандай міндетті атқаратыны баяндалады. Паскаль тіліндегі түсіндірме { және } яки (* және *) түрдегі жақшаның ішіне жазылады. Мысалы, {бұл түсіндірме мысал ретінде келтірілді} яки (* түсіндірмені осылай жазу мүмкін*).

Әдетте, Паскаль тіліндегі бағдарламалар **Program** деген арнайы сөзбен басталады. Осы сөзден кейін бағдарламаның аты жазылады: **Program** квадрат_тендеу; {квадрат тендеуді шешу бағдарламасы}.

Бағдарламаның аты оның міндетіне сай келсе, мақсатқа лайықты болады. Бұл – қажетті бағдарламаны басқа бағдарламалар арасынан тез ажыратып алуға мүмкіндік береді. Айтып өту керек, бағдарламаның аты оның ішіндегісіне ешқандай ықпал жасамайды, жалпы алғанда, бағдарламаға ат қою шарт емес. Паскаль бағдарламалау тілінде бағдарлама жазу үшін мыналар қолданылады:

Константалар (тұрақтылар) – бағдарлама істегенде мәні өзгермейтін шамалар;

Айнымалылар – бағдарлама істегенде мәні өзгертін шамалар;

Өрнектер – тиісті амалдармен байланысқан тұрақтылар, айнымалылар мен функциялар;

Операторлар – бағдарламалау тілінің белгілі бір аяқталған амалын беруге арналған команда;

Функциялар мен процедуралар – өзіндік аты бар арнайы бағдарламалық бөлімдер. Оларға негізгі бағдарламадан жүгінеді;

Белгілер — бағдарламада басқаруды беріп отырған операторды көрсетеді.

Паскаль тілінің нұсқаулары мен командалары, **модуль** деп аталатын **TPU** кеңейтуі бар арнайы файлдарға орналасқан. Оларға мысал ретінде **system** (систем), **crt** (сиерти), **graph** (граф) модульдерін келтіруге болады. Модульдердің әрбірі белгілі бағыттағы нұсқаулар мен командаларды қамтиды. Мысалы, **system** модулі Паскальдің стандартты (негізгі) нұсқауларын, **crt** модулі экранмен және клавиатурамен істеуге (экранды тазалау, экранға бірнеше бөлек айна орнату,...), ал **graph** модулі графикалық күйде істеуге арналған нұсқаулар мен командаларды қамтиды. Бағдарламадағы модульдер керегіне қарай қолданылады. Бір модульдің құрамына енген нұсқауды пайдалану үшін бағдарламаның басында (тақырыптан кейін) ол туралы нұсқау беру керек. Бұл Паскальдің арнайы **Uses** сөзінің көмегімен іске асырылады. Мысалы, бағдарламадағы графикті пайдалану үшін оған **Uses graph**; жазуын енгізу қажет. Егер бағдарламада бірнеше модуль қолданылатын болса, олар бір-бірінен үтірмен ажыратылып жазылады. Мысалы: **Uses crt, graph**;

Паскаль интегралданған ортасын іске қосқанда **system** модулі автоматты түрде жатқа жүктеледі. Сондықтан **Uses system**; жазуы қолданылмайды. Кейбір бағдарламалар үшін **system** модулінің өзі жеткілікті болады.

Бағдарламаны жазудан бұрын оған қатысатын шамаларды анықтап алу, айнымалыларға ат қою және оларды **сипаттау** (түрін көрсету) керек болады. Содан кейін ғана бағдарламаның **негізгі бөлігі** басталады, яғни Паскальда бағдарлама екі бөлімнен тұрады.

Паскаль тіліндегі бағдарламалардың жалпы жағдайдағы құрылымы мынадай:

Program бағдарлама аты; {міндетті емес}

Uses {Модульдер тізімі}

Label {Белгілер тізімі}

Const {Константаларды сипаттау}

Var {Айнымалыларды сипаттау}

Процедуралар мен функциялар

Begin

{Негізгі бөлім}

End.

Label, Const, Var, Begin, End – Паскаль тілінің арнайы сөздері, **label** – белгі, **const** (constant – константа) – тұрақты шама, **var** (variable) – айнымалыларды сипаттау, **begin** – басталу, **end** – аяқталу деген мағыналарды білдіреді.

Идентификатор дегенде тұрақтылар, айнымалылар, процедуралар, функциялар, модельдер, бағдарламалардың аты түсініледі. Идентификаторлар **стандартты** және **пайдаланушы** деген түрлерге бөлінеді. Стандартты идентификатор – бағдарлама тарапынан алдын ала белгіленген болады.

Пайдаланушы идентификаторын бағдарламашының өзі таңдап алады, оның ұзындығы еркінше болуы мүмкін, бірақ бірінші 63 белгісінің (ерекшелейтін) **мағынасы бар** болады. Идентификатордың аты латын әрпінен я асты сызылған (_) белгісінен басталуы және бос орынсыз жазылуы шарт. Бірінші белгіден кейін әріптер, цифрлар, асты сызылған белгілер жазуға болады. Турбо Паскаль тілінде идентификатор аттарын, кез келген регистрде (төменгі яки жоғары) жазуға да болады, яғни **aFa, Afa, aFa** сияқтылар бірдей **ат** деп есептеледі. Өйткені **Турбо Паскаль трансляторы** бағдарламаны компиляциялаған (бағдарламаны машина тіліне аударған) кезде идентификатор аттарындағы, көмекші сөздердегі барлық үлкен әріптерді кіші әріптерге ауыстырады. Аттар апостроф ішіне алынбайды, яғни **‘Мен‘** және **‘мен‘** сияқтылар ат бола алмайды.

Паскаль бағдарламалау тілінде төменде келтірілген сөздер резервке сақталып, оларды пайдаланушы идентификаторы ретінде қолдануға болмайды:

and, asm, array, begin, case, const, constructor, destructor, div, do, downto, else, end, exports, file, for, function, goto, if, implementation, in, absolute, assembler, export, external, far, forward, index, interrupt, near, private, public, resident, virtual, inherited, inline, interface, label, library, mod, nil, not, object, of, or, packed, procedure, program, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, while, with, xor.

Паскаль бағдарламалау тілі бұл сөздерді бағдарламада істетуге жол бермейді, сондай-ақ қате хабарды экранда **«Error 2: Identifier expected»** (резерв идентификатор) жазуы арқылы бейнелейді.



Сұрақтар мен тапсырмалар

1. Паскаль бағдарламалау тілінің өліпбиін айтып бер.
2. Логикалық амалдарды растық кестесіне сүйеніп түсіндір.

3. Оператор деген не?
4. Бағдарламаға ат қою туралы айт.
5. Идентификатор туралы мәлімет бер.
6. Паскальдағы бағдарламаны қандай бөлімдер құрайды?
7. Бағдарламаның сипаттау бөлімі туралы мәлімет бер.

Жаттығулар

1. Оң жақ бағандағы белгілердің сол жақ бағандағы тобына тура келетінін анықта.

Логикалық амалдар
Қатынас символдары
Арнайы символдар

%, \$, @, &, (,), {, }, [,]
NOT, OR
<, <=, >, >=

2. Сол жақ бағандағы сөздерге оң жақтағы сипаттамалардың сәйкесін анықта.

Модульдерді қолдану
Таңбаларды сипаттау
Бағдарламаға ат қою
Айнымалыны сипаттау

Program 9_сынып;
Var a21: integer;
Label 19;
Uses Crt;

3. Идентификатор аттарын «Дұрыс жазылған» және «Қате жазылған» деген топтарға ажыратып, себебін түсіндір.

a	1 күн	Менің Бірінші Бағдарламам	BMA
Шекара#4	Кейінгі жыл	Күн_21_жыл_1963	and

15-сабақ. Тұрақты және айнымалы шамалар

Паскаль тілінде негізгі үш түрлі: **тұрақты**, **айнымалы** және **кестелік** көріністегі шамалар пайдаланылады. Олар символдық, жолдық, логикалық және сандық түрдегі мәндерді қабылдай алады.

Тұрақты шамалар

Символдық тұрақтыларды апостроф ішіне алынған бір белгі – әріп, цифр яки арнайы белгі құрайды. Мысалы: 'a'; 'B'; '9'; '-'; '' және т. б.

Жолдық тұрақтыларды (белгілерден тұратын жол) саны 0-ден 122 белгіге дейінгі және апостроф ішіне алынған әріптер, цифрлар, арнайы белгілер тізбегі құрайды. Мысалы:

'Ташкент'; 'A 549'; 'B***M.'; '47%'; 'BMA = '; '..—...—' және т.б.

Апостроф ішіне ештеме жазылмаса, ол **бос жол** делінеді.

Логикалық тұрақтылар — True (рас) немесе False (жалған) логикалық мәндердің бірі.

Сандық тұрақтылар екі түрде — **бүтін** немесе **нақты** болуы мүмкін. **Бүтін сандар** — таңбалы яки таңбасы жоқ — 2147483648 -ден +2147483647 -ге дейінгі бүтін сандар. Егер бүтін сандық тұрақты шама осы аралықтан шығып кетсе, транслятор осы қате туралы хабар береді. Нақты сандар өз кезінде жылжымайтын нүктелі және жылжымалы нүктелі сандарға бөлінеді.

Ондық бөлшектердің бүтін және бөлшек мәнін ажыратушы «үтірдің» орнына Паскаль бағдарламалау тілінде «нүкте» жазылады.

Жылжымайтын нүктелі сандар — ондық бөлшек түріндегі сандар:

— 2.753; 283.45; 0.517; — 0.0013.

Жылжымалы нүктелі сандар — экспоненциал көріністе (Е яки е көмегімен) өрнектелген сандар. Сандар жазудың осы әдісі өте кіші яки өте үлкен сандарды өрнектеу үшін өте қолайлы. Ол былай оқылады:

2.1E+07 — «онның 7-дәрежесіне көбейтілген 2.1»;

2.301e-63 — «онның минус 63-дәрежесіне көбейтілген 2.30».

Мысалы, $3400000000 = 3,4 \cdot 10^9$ саны Турбо Паскальда 3.4E+09 сияқты экспоненциал түрінде жазылады. **Е** әрпінен алдын жазылған сан **мантисса** деп, ал **Е** әрпінен кейін жазылған сан **тәртіп** деп аталады. Мантисса бүтін я жылжымалы нүктелі сан, ал тәртіп тек бүтін сан болуы мүмкін.

Мысалы:

$37.3879 E-3 = 0.0373879$; $5.31 E+5 = 531000$; $-0.075 E-5 = -0.00000075$; $-2.37 E-4 = -0.000237$

Паскаль тілінде жазылған бағдарламаға **сипатталған тұрақтылар** қатысуы мүмкін. Мысалы,

Const A=21071963; _m10m10='2301'; Pi=3.141516;

Айнымалы шамалар

Айнымалылар бағдарламаның сипаттау бөлімінде сөзсіз сипатталған, яғни олардың түрі көрсетілген болу керек. Бағдарламадағы айнымалының сипаттамасы Паскальдің **Var** көмекші сөзімен басталады:

**Var айнымалы : түрі;
айнымалы : түрі;**

Егер бірнеше айнымалының түрі бірдей болса, оларды бөлек сипаттамай, бірге сипаттауға болады:

Var 1-айнымалы, 2-айнымалы,..., n-айнымалы : түрі;

Бүтін сандық мәндер қабылдайтын айнымалылар бүтін сандық айнымалылар деп аталады. Олар 5 түрге бөлініп, бір-бірінен қабылдайтын мәндердің шекарасымен және компьютер жадынан алатын орнымен (көлемі бойынша) ерекшеленеді. Төмендегі кестеде бүтін сандық айнымалыларды сипаттауға қажетті Паскаль тілінің арнайы сөздері, оларға тән мәндер шекарасы және иелейтін жадының көлемі келтірілген:

Түрі	Мәндер шекарасы	Иелейтін жад көлемі
Byte	0 ...255	8 бит = 1 байт
ShortInt	-128 ...127	8 бит = 1 байт
ShortInt	-128 ...127	8 бит = 1 байт
Word	0 ...65 535	16 бит= 2 байт
Integer	-32 768 ...32 767	16 бит = 2 байт
LongInt	-2 147 483 648 ...2 147 483 647	32 бит = 4 байт

Мысалы: var i, j: Integer; bma: longint; mnr: Shortint; тәртiп_цифр: Byte; nat_0: word;

Бүтін сандармен **div** (бүтін болу) және **mod** (қалдық) амалдарын орындау орынды болады. Мысалы:

$$25 \text{ div } 4 = 6; 25 \text{ mod } 4 = 1; 49 \text{ div } 7 = 7; 49 \text{ mod } 7 = 0.$$

Нақты сандық мәндер қабылдайтын айнымалылар **нақты сандық айнымалылар** деп аталады. Олардың түрі төмендегі кестеде келтірілген.

Түрі	Мәндер шекарасы	Разряды	Иелейтін жад көлемі
Real	$-2,9 \cdot 10^{39} \dots 1,7 \cdot 10^{38}$	11—12	6 байт
Single	$-1,5 \cdot 10^{45} \dots 3,4 \cdot 10^{38}$	7—8	4 байт
Double	$-5,0 \cdot 10^{324} \dots 1,7 \cdot 10^{308}$	15—16	8 байт
Extended	$-3,4 \cdot 10^{4932} \dots 1,1 \cdot 10^{4932}$	19—20	10 байт
Comp	$-9,2 \cdot 10^{18} \dots 9,2 \cdot 10^{18}$	19—20	8 байт

Мысалы:

```
var бұрышы, жебе_ұзындығы : Real; mab : extended;
дәреже : Single; kub : double; нақты : Comp;
```

Кестедегі «Разряды» санның нақты цифрлар санын білдіреді. Көп жағдайда нақтылы айнымалыларды пайдалану жеткілікті болады.

Символдық тұрақтылар мәнін қабылдайтын айнымалылар **символдық айнымалылар** деп аталады. Олар Паскальдің **Char** арнайы сөзімен сипатталады. Мысалы: var әріп, символ: char;

Жолдық айнымалыларды сипаттау үшін Паскальдің **String** арнайы сөзі қолданылады. Мұндай айнымалылар үшін компьютер жадынан (255 символ үшін) 255 байт орын бөлінеді. Егер жолдық айнымалыны қабылдайтын жолдағы символдар саны бағдарламаның жұмыс барысында белгілі бір шамадан, мысалы 10 символдан аспаса, компьютер жадын үнемдеу үшін, оны String[10] арқылы сипаттауға болады. Мысалы:

```
var қатар : String; {қатар атаулы айнымалыға жадынан 255 байт бөлінеді}
```

```
_жол : String[24]; {_жол атаулы айнымалыға жадынан 24 байт бөлінеді}
```

Логикалық тұрақты мәндерді қабылдайтын айнымалылар логикалық айнымалылар деп аталып, Паскальдің **Boolean** деген арнайы сөзімен сипатталады. Мысалы:

```
var нәтиже : Boolean;
үлкен, кіші : Boolean;
```

Паскаль тілінде құрастырылған бағдарламаға тек сипатталған айнымалылар ғана қатысады. Паскаль трансляторы сипатталмаған айнымалыларды бағдарламада істетуге жол бермей, қате туралы хабарды экранға **«Error 3: Unknown identifier»** (бейтаныс идентификатор, яғни белгісіз айнымалы) жазып бейнелейді. Соны да ұмытпаған жөн, айнымалыларға тек сипаттамада көрсетілген түрдегі мәндерді ғана беру мүмкін.



Сұрақтар мен тапсырмалар

1. Символдық тұрақты дегенде нені түсінесің? Мысалдар келтір.
2. Жолдық тұрақтының символдық тұрақтыдан айырмашылығы неде?
3. Сандық тұрақтылардың қандай түрін білесің?

4. Логикалық тұрақтылар қандай мөндерді қабылдайды?
5. Айнымалылардың тұрақтылардан айырмашылығы неде?
6. Бүтін сандық айнымалылардың түріне мысалдар келтір.
7. Нақты сандық айнымалылардың түріне мысалдар келтір.
8. Символдық айнымалылар қалай сипатталады? Мысалдар келтір.
9. Жолдық айнымалыларды қалай сипаттауға болады? Мысалдар келтір.

Жаттығулар

1. Төмендегі тұрақтылардың түрін айтып бер.
 - а) '7!'; 'информатика'; '-987378'; 'BMA';
 - ә) 99; -200; 101; 87;
 - в) 0.01; 8.909; 132.001; 878887.1;
 - г) 0.07 E-3; -9.8 E6;
 - б) '; 'u'; '0'; '';
 - д) True; False.
2. Төмендегі айнымалылардың түрін анықтап, оларды түсіндір.
 - а) мен : Boolean;
 - ә) өмір : Real;
 - б) бақыт : Integer;
 - в) ержүрек : String[7];
 - г) сан : char;
 - д) мол : Single;

16-сабақ. Тұрақты және айнымалы шамалар тақырыбын қайталау

1. Төмендегі тұрақтылардың түрін айтып бер.
 - а) -9.22 E-2; 0.01 E+5; 1.11 E-4;
 - ә) 21; 21; 7; 7; 19; 19; 63; 63;
 - б) true; true; true; false; false; false;
 - в) '555'; 'aar'; 'mmr'; 'bbj'; 'aga';
 - г) 'Қасиетті'; 'Отан'; 'Тәуелсіз';
 - д) 'a'; 'д'; 'a'; 'm'; 'з'; 'a'; 't';
2. Айнымалыларға ат қойып, оларды түріне қарай сипатта.
 - а) символдық; ә) нақты; б) логикалық; в) жолдық;
 - г) белгілері 7-ден аспайтын жолдық.
3. Әр тарауда бір айнымалының барлық мәні берілген немесе сипаттамасы бейнеленген. Осы айнымалыларға ат қойып, оларды сипатта.
 - а) -5; 0; 7; 58; -15; 9.
 - ә) 'Халық'; 'Отан'; 'Ана'.
 - б) 7.21; 4.2; 50.1902; -1.23.
 - г) true; true; false; true; false.
 - в) бірінші 7 түпкі сан
 - д) '000'; '001'; '002'; '003'.
 - ж) әліпби әріптері.
 - е) 'Жоғары'; руханият, 'жеңілмес'; 'күш'.
4. Барлық мәні берілген бүтін айнымалыларға ат қойып, оны сипатта. Айнымалылар түрін таңдауда жадынан аз орнын алуға тырыс.
 - а) -4; 0; -4; 8; 12;
 - ә) 1; 16; 256; 4096; 65536;
 - б) 0; 2; 4; 6; 8; 10;
 - в) 29350; -2; 8000; 250;
 - г) 5; -32767; 46; 0; 32767;
 - д) 200000; 2000; -20; 99999;

- 5*. Берілген амалдар нәтижесінде шығатын айнымалыларды сипатта.
- а) f : бүтін; g : нақты; $d:=f*g+f+g$;
 - ә) d : бүтін; n : бүтін; $k:=d+2*n$;
 - б) s : логикалық; e : логикалық $q:= \text{not}(s \text{ or } e)$;
 - в) k : тақ; m : жұп; $v:= k+m/2$;

17-сабақ. Кестелік шамалар

Күнделікті өмірде әртүрлі кестелер пайдаланылады: сабақ кестесі, шахмат яки футбол ойындары бойынша жарыс кестесі, Пифагор (көбейту) кестесі, септелу кестесі және т.б. Кестені құрастырушылар оның **элементтері** делінеді. Кестелік шамалар **бір өлшемді** (сызықтық), **екі өлшемді** (тік төртбұрышты), **үш өлшемді** (параллелепипед түрінде) т.с.с. болады. Біз осы оқулықтың аясында сызықтық пен тік төртбұрышты кестелерді қарастырамыз.

Сызықтық кестелер жол немесе баған түрінде бейнеленеді. Мысалы, сыныбындағы оқушылар тізімі сынып журналындағы баған түріндегі кестеге жазылған. Оқушылардың аты-жөні сол кестенің элементтерін құрайды. Олардың әрқайсысының өзіндік реттік нөмірі болады, әр реттік нөмірге бір-ақ оқушының аты-жөні тура келеді.

Екі өлшемді кестелер бағандар мен жолдардан құрастырылады (мәтін процессоры мен электрондық кестелерге қатысты тақырыптарды еске түсір). Олардың элементтері бағандар мен жолдар қиылысқан торларға орналасады. Мұндай кестелердің белгілі бір элементін көрсету үшін, оның нешінші жолда және нешінші бағанда орналасқанын, яғни жол мен баған бойынша реттік нөмірін білу керек болады. Демек, екі өлшемді кестенің әрбір элементіне екі (жол және баған бойынша) реттік нөмірі тура келеді.

Паскаль бағдарламалау тілінде кестелермен жұмыс істеу үшін арнайы ұғым енгізілген. **Массив** – кестелік шама болып, оны нақты сандағы бір түрлі және ретке келтірілген (яғни, реттік нөмірі бар) элементтер кешені құрайды. Массив элементтерінің реттік нөмірі бүтін сандармен бейнеленеді, яғни олар **теріс** сан да болуы мүмкін.

Паскаль тілінде әр массивтің өз аты бар, оларға ат қою – айнымалыларға ат қоюға ұқсайды. Мысалы: `a5`, `сабақ_кестесі`, `түбір_сандар`. Массив элементтерінің реттік нөмірі **индекс** деп

аталып, ол квадрат жақша ішінде жазылады. Мысалы, $a[5]$ жа-
зуы – a атаулы массивтің бесінші элементін білдіреді, яғни
массивтің аты – a , ал индексі – 5 .

1-есеп. 7 элементтен тұратын, A атаулы сызықтық кесте құр.

Реттік нөмірі	1	2	3	4	5	6	7
Мәні	x	A	a	t	r	z	m

Демек, кесте элементтері және оларға тән мәндер мынадай сәйкестікте
болады екен:

Кесте элементі	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
Мәні	x	A	a	t	r	z	m

Екі өлшемді массив элементтері екі индекс арқылы анықталып, олар
бір-бірінен үтірмен ажыратылып жазылады да, бірінші индекс жолдың
реттік нөмірін, екінші индекс бағанның реттік нөмірін білдіреді. Мысалы,
 $S[4,3]$ жазбасы – S атаулы массивтің 4-жолы мен 3-бағаны қиылысқан торда
орналасқан элементін білдіреді.

2-есеп. S атаулы 4×5 (4-ке 5, яғни 4 жолы мен 5 бағаны бар) тік
төртбұрышты кестені бейнеле (тордағы кесте элементтері **көк** түсті етіп
жазылған).

Жол бойынша реттік нөмірі	Баған бойынша реттік нөмірі					
		2	3	4	5	6
	1	3.2 $S[1,2]$	1.37 $S[1,3]$	-1.25 $S[1,4]$	7.12 $S[1,5]$	-11.4 $S[1,6]$
	2	0.5 $S[2,2]$	1.1 $S[2,3]$	1.2 $S[2,4]$	-1,1 $S[2,5]$	4.22 $S[2,6]$
	3	-0.1 $S[3,2]$	1.01 $S[3,3]$	71.2 $S[3,4]$	4.1 $S[3,5]$	-4.11 $S[3,6]$
4	6.3 $S[4,2]$	-7.01 $S[4,3]$	1.5 $S[4,4]$	7.5 $S[4,5]$	-1.09 $S[4,6]$	

Кестеден көрініп тұр, мысалы, $S[1,3]=1.37$, $S[2,2]=0.5$, $S[4,6]=-1.09$.

Айнымалылар сияқты, бағдарламадағы массивтер де сипатталуға тиіс.
Бұл үшін Паскаль бағдарламалау тілінің **Array** көмекші сөзі қолданыла-
ды. Бұл сөзден кейінгі квадрат жақшаның ішіне массивтің бірінші және
соңғы элементтерінің реттік нөмірі бір-бірінен **екі нүктемен** (..) ажыра-
тылып жазылады. Кейін Паскальдің **of** көмекші сөзі жазылады.

Мысалы:

var

A: array[1..7] of char; {1-есептегі элементтері 1-ден 7-ге дейін реттелген char (символдық шама) түрдегі **A** атаулы сызықтық массив};

S: array[1..4,2..6] of real; {2-есептегі жолдары 1-ден 4-ке дейін әрі бағандары 2-ден 6-ға дейін реттелген шын (нақты шама түріндегі) **S** атаулы екі өлшемді массив};

bma: array[-2..100] of integer; {-2 -ден 100 -ге дейін реттелген **бүтін** түрдегі **bma** атаулы сызықтық массив}.



Демек, массив (кестелік шама) дегенде, **бір ғана атпен белгіленген, бірдей ретке келтірілген шамалар кешені** түсініледі.

3-есеп. Бір өлшемді A кестенің бес элементі бар дейік:

Реттік нөмірі	-1	0	1	2	3
Мәні	3	2	12	10	-8

Паскаль тілінде бұл кестенің элементтері былай бейнеленеді:

$$A[-1] := 3; A[0] := 2; A[1] := 12; A[2] := 10; A[3] := -8;$$

Массив элементтерінің индексін бірер бүтін сандық айнымалы, (мысалы, i) арқылы өрнектеуге болады. Мысалы, егер $i = 1$ болса, $A[i] = 12$, егер $i = 3$ болса, $A[i] = -8$ болады.

4-есеп. Екі өлшемді бүтін сандық B массив берілген болсын:

$$B = \begin{bmatrix} 3 & 10 & 5 \\ 2 & 7 & 9 \end{bmatrix}.$$

Массив элементтерінің реттік нөмірін өзіміз беріп, $B[0,0]$, $B[0,1]$, $B[0,2]$, $B[1,0]$, ... сияқты жазып аламыз:

$$B = \begin{bmatrix} 3 & 10 & 5 \\ 2 & 7 & 9 \end{bmatrix} = \begin{bmatrix} B_{00} & B_{01} & B_{02} \\ B_{10} & B_{11} & B_{12} \end{bmatrix} = [B_{ij}],$$

мұндағы $i = 0, 1$ және $j = 0, 1, 2$ (i – жолдың реттік нөмірі, j – бағанның реттік нөмірі) мәндерін қабылдайды. Бұл кесте Паскаль тілінде былай жазылады:

var b: array[0..1, 0..2] of Integer;

Ескерте кету керек, идентификатор атының қай регистрде орналасқаны маңызды емес!

Жалпы жағдайда индекс ретінде айнымалы яки өрнек қолданылады. Мысалы, $I=0, J=2$ болса, 4-есепте $B[I, J] = 5$ және $(I+1=0+1=1$ және $J-2=2-2=0$ болғандықтан) $B[I+1, J-2] = 2$ болады. Біз кестелердің тек сызықтық және тік төртбұрышты түрімен таныстық. Негізінде Паскаль тілінде көп өлшемді (255 -ке дейін) кестелік шамаларды да пайдалануға болады. Мұндай кестелерді сипаттау үшін бірнеше есептер келтіреміз.

- 1) var s: array[1..4, 1..7, 0..10] of Byte; {s – Byte түрлі 3 өлшемді кесте};
- 2) var t, k: array [1..100, 1..80, 1..50] of string; {t және k – 3 өлшемді жолдық кестелер};
- 3) var f: array [- 5..10, 0..10, 2..10] of char; {f – 3 өлшемді символдық кесте}.

Паскаль бағдарламалау тілінде сипатталған массивтер үшін жадында орын сақтап қояды. Сондықтан жадының керегінен артық бөлігін алмау үшін, массивтің түрінен тыс, оның неше элементтен тұратынын білу мақсатқа сай болады. Жалпы айтқанда, сызықтық К -ден S -ға дейін реттелген массив элементтері саны $S-K+1$, екі өлшемді жолдары В-дан М -ға дейін және бағандары А-дан G -ге дейін реттелген массив элементтері саны $(M-B+1) \cdot (G-A+1)$ болады.

Мысалы, 3-есептегі -1 -ден 3 -ке дейін реттелген А массивтің $(3 - (-1)+1=3+1+1=)$ 5 бүтін мәнді элементі, 4-есептегі жолдары 0-ден 1-ге дейін және бағандары 0-ден 2-ге дейін реттелген В массивтің $((1 - 0+1) \cdot (2-0+1)=2 \cdot 3=)$ 6 бүтін мәнді элементі бар болады.



Сұрақтар мен тапсырмалар

1. Тұрмыста кездесетін кестелік шамаларға мысалдар келтір.
2. Сызықтық массивтің қандай өлшемдері болады?
3. Массивтегі индекс не үшін қажет?
4. Массив элементтерінің индекстері қандай мәндер қабылдайды?
5. Кестелік шамалардың түрлерін қалай ажыратуға болады?

Жапғыулар

1. Төмендегі тізбектер (циклдер) неше өлшемді массивтерді өрнектейтінін және неше элементтерден тұратынын анықта.

- а) A[0], A[1], A[2], A[3], ... , A[99];
- ә) B[0,0], B[0,1], B[0,2], ... , B[3,5];
- б) M[0,0,0], M[0,0,1], ... , M[1,1,1].
- в) G[-22,3], G[-22,4], G[-22,5],..., G[-20,5].

2. 100 элементі бар бүтін сандық әрі сызықтық кесте қайсы жауапта дұрыс сипатталған?

- а) var B: array [1..100] of real;
- ә) var M: array [1..100] of char;
- б) var A: array [0..99] of string;
- в) var G: array [5..104] of integer;

3. 2-есептегі массивтердің түрін, өлшемі мен элементтер санын анықта.

4. Біреуден бір өлшемді бүтін және символдық кестелерді, сондай-ақ екі өлшемді нақты түрдегі кестені сипаттап бер.

18-сабақ. Кестелік шамалар тақырыбын қайталау

1. Сегіз жолы мен он бір бағаны бар тік төртбұрышты нақты түрдегі F кестенің қалай жазылатынын анықта.

- а) var A: array [8..11] of real;
- ә) var B: array [1..8,1..11] of integer;
- б) var D: array [8..11,8..11] of real;
- в) var M: array [0..8,0..10] of integer;
- г) var F: array [0..7,0..10] of real; д) var F: array [0..7,0..10] of char;

2. 1-жаптығуда сипатталған массивтердің түрін, өлшемі мен элементтер санын анықта.

3. Пифагор кестесін құр. Кесте элементтерін талда. Массивке ат қойып, оны сипатта.

4. Жанұяң мүшелерінің аты-жөні, туылған жылы мен білімі туралы кесте құр. Массивті сипаттап, оның элементтерін талда.

5. Төмендегі **integer** түріндегі M сызықтық массивтің түріне қарай бос тордағы оның элементтеріне сәйкес мән бер. Массивті сипатта.

M[-7]	M[-6]	M[-5]	M[-4]	M[-3]	M[-2]	M[-1]

6. Төмендегі **char** түріндегі B сызықтық массив түріне қарай бос тордағы оның элементтеріне сәйкес мән бер. Массивті сипатта, элементтерін баған түрінде жаз. Массивті екі өлшемді массив түріне өткізіп, қайтадан сипатта.

B[9]	B[10]	B[11]	B[12]	B[13]	B[14]	B[15]	B[16]

7. Төмендегі **string** түріндегі A массив элементтеріне бос торда түріне сәйкес мән бер. Кестені сипатта. Массивке басқа реттік нөмір беріп, оны G атымен қайта сипатта.

19-сабақ. Стандартты функциялар мен процедуралар, алгебралық өрнектер

Функция ұғымы саған математикадан мәлім. Функциялар қасиеттеріне қарай түрлі топтарға бөлінеді, бұлар: сызықтық, квадратты, тригонометриялық және т.б. Кейбір функциялар Паскаль бағдарламалау тілінде де пайдаланылады. Паскаль тілінде функцияларды пайдалану қолайлы болуы үшін, кейбір функциялар бағдарлама трансляторының қамсыздандыруына енгізілген. Бағдарлама трансляторының қамсыздандыруына енгізілген функ-

циялар **стандартты функциялар** делінеді. Бұл функциялардын көпшілігі саған MS Excel бағдарламасы арқылы таныс. Сондай-ақ Паскальда анық бір амалдарды орындауға арналған **стандартты процедуралар** колданылады.

Төмендегі кестеде Паскальдің кейбір **стандартты функциялардың** сипаттамасын келтіреміз:

Функция аты	Аргумент түрі	Мән түрі	Түсіндірме
Математикалық функциялар			
Abs(x)	бүтін/нақты	бүтін/нақты	x -тің абсолюттік мәні (модулі): $ x $
Sin(x)	бүтін/нақты	нақты	x -тің синусы (радиан ө.б.): sinx
Cos(x)	бүтін/нақты	нақты	x -тің косинусы (радиан ө.б.): cosx
Arctan(x)	бүтін/нақты	нақты	x -тің арктангенсі: arctgx
Sqrt(x)	бүтін/нақты	нақты	x -тің квадрат түбірі (x ≥ 0): \sqrt{x}
Sqr(x)	бүтін/нақты	бүтін/нақты	x -тің квадраты: x²
Exp(x)	бүтін/нақты	нақты	e^x (e = 2.718282...)
Ln(x)	бүтін/нақты	нақты	x -тің натуралды логарифмі (x > 0): ln x
Frac(x)	бүтін/нақты	нақты	x -тің бөлшек бөлігі: {x}
Int(x)	бүтін/нақты	нақты	x -тің бүтін бөлшегі: [x]
Random	—	нақты	[0, 1) аралықтағы кездейсоқ сан
Random(x)	Word	Word	[0, x) аралықтағы кездейсоқ сан
Айнымалылар түрін өзгертетін функциялар			
Trunc (x)	нақты	Long Int	x -тің бүтін бөлшегі
Round (x)	нақты	Long Int	x -ті бүтінге дейін дөңгелектейді
Odd (x)	бүтін	логи- калық	x – тақ сан болса, «рас» мәнін алады
Chr (x)	Byte	Char	x -тің ондық ASCII кодына сай белгі
Ord ('m')	Char	Byte	'm' белгінің ондық ASCII коды
Математикалық процедуралар			
Ins (x)	бүтін	бүтін	x -тің мәнін 1 -ге арттырады (x:=x+1)
Dec (x)	бүтін	бүтін	x -тің мәнін 1 -ге арттырады (x:=x+1)

1-мысал. Кейбір функциялардың пайдаланылуы:

Функция	Мәні	Функция	Мәні	Функция	Мәні
abs(-5)	5	abs(-4.9)	4.9000000000e+00	abs(4.9)	4.9000000000e+00
sqr(4)	16	sqr(2.5)	6.2500000000e+00	Sqrt(16)	4.0000000000e+00
sqr(-4)	16	Sqr(0.0)	0.0000000000e+00	Sqrt (0.16)	4.0000000000e-01
sqr(0)	0	Sin(0)	0.0000000000e+00	Sin(1)	8.4147098481e-01
trunc(5.3)	5	Int(5.3)	5.0000000000e+00	Int(5)	5.0000000000e+00
trunc(-5.3)	-5	Int(-5.3)	-5.0000000000e+00	frac(5.3)	3.0000000000e-01
Round(5.49)	5	frac(-5.3)	-3.0000000000e-01	frac(5)	0.0000000000e+00
Round(5.5)	6	Odd(5)	TRUE	Odd(-5)	TRUE
Round(-5.49)	-5	Odd(4)	FALSE	Odd(-4)	FALSE
Round(-5.5)	-6	Odd(0)	FALSE	Chr(65)	'A'
Chr(97)	'a'	Ord('A')	65	Ord('a')	97

Математикалық формулаларда көп қолданылатын π санын өрнектеу үшін Паскальда арнайы Пи тұрақтысы (константа) ажыратылған ($\text{Пи}=3.1415\dots$).

Паскаль бағдарламалау тіліндегі **алгебралық өрнектер** құрамын арифметикалық амалдар арқылы байланысқан тұрақтылар, айнымалылар мен функциялар құрайды. Алгебралық өрнектер бір жолдық етіп жазылады, яғни жолдан төмен түсіріп не жоғарыға көтеріп жазуға болмайды. Мысалы, $3ab^2$ өрнегі Паскальда

$3*a*sqr(b)$ яки $3*a*b*b$ сияқты, $\frac{a}{b^2}$ өрнегі $a/sqr(b)$ яки $a/(b*b)$ сияқты жазылады.

Өрнектер жазудағы амалдардың орындалу тәртібін **көрсету** үшін тек қарапайым жақшалар қолданылады. Жақшаның ішіндегі амалдар солдан оңға қарай, математикада қабылданған ережеге сәйкес тізбектеп орындалады:

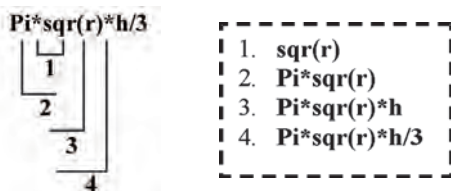
- функциялар мәндері есептеледі;
- көбейту немесе бөлу амалы орындалады;
- қосу немесе азайту амалы орындалады.

Мысалы, $\frac{a+b}{c}$ арифметикалық өрнектің Паскаль тіліндегі жазылуы $(a+b)/c$ сияқты, оны есептеуде әуелі жақша ішіндегі амал, яғни $a+b$ орындалады, кейін нәтижесі c -ке бөлінеді. Амалдарды орындау тәртібі де жақшалар жәрдемімен ретке келтіріледі: $\sqrt{a^2 - b^2}$ өрнегі Паскальда $\text{sqr}(\text{sqr}(a) - \text{sqr}(b))$ сияқты, $|x + \text{tg}x|$ өрнегі $\text{abs}(x + \text{tan}(x))$ түрінде жазылады.

2-есеп. R мен H айнымалылардың белгілі мәндерінде мына өрнектің мәні есептелсін:

$$\frac{1}{3} \pi R^2 H.$$

Бұл өрнек Паскальда $\text{Pi}*\text{sqr}(r)*h/3$ сияқты жазылады. Мұнда амалдар мына тәртіппен орындалады:



Әрине, екі арифметикалық амал тізбектеп келгенде, өрнекті жақшаның ішіне жазуға болады. Мысалы: $5*(-1)$ яки $a+(-b)$.

Кейде Паскаль бағдарламалау тілінде жазылған өрнекті кәдімгі математикалық түрде жазу керек болады. Мысалы, Паскаль бағдарламалау тілінде жазылған $0.5*(\sin(x)+\cos(x))$ өрнегінің математикалық көрінісі мынадай болады:

$$\frac{1}{2}(\sin x + \cos x).$$

Паскаль тіліндегі стандартты функциялар барша математикалық амалдарды қамтымаған. Сондықтан кейбір математикалық амалдарды Паскаль тілінің бірнеше стандартты функциясы арқылы немесе бір стандарт функцияны бірнеше рет қолдану жолымен өрнектеуге тура келеді. Мысалы, Паскаль тілінде санды ерікті дәрежеге өсіру функциясы жоқ. Сондықтан a^3 өрнегін Паскаль тілінде $a*a*a$ яки $\text{sqr}(a)*a$ деп, ал a^4 өрнегін $a*a*a*a$ я $\text{sqr}(\text{sqr}(a))$, яки $\text{sqr}(a)*\text{sqr}(a)$ деп жазуға болады.

Жалпы алғанда, a^b ($a>0$) түрдегі өрнек үшін математикалық $a^b = e^{b \ln a}$ формула орынды болады. Сондықтан a^b ($a>0$) өрнегі Паскаль тілінде $\text{exp}(b*\text{Ln}(a))$ сияқты жазылады.

3-есеп. $\frac{x-y}{x^5-y^3}$ алгебралық өрнегін Паскаль тіліндегі көрінісін жаз.

Шешүі. Бұл өрнекті Паскаль тілінде бірнеше әдіспен сипаттау мүмкін. Солардың бірі мынау:

$$(x-y)/(\exp(5*\ln(x))-sqr(y)*y).$$

Ескерте кету керек, Паскаль бағдарламалау тіліндегі стандартты функциялар аргументі әрдайым жақшаның ішіне жазылады!



Сұрақтар мен тапсырмалар

1. Қандай функциялар стандартты функция деп аталады?
2. Стандарт функцияның өдеттегі жазылуының Паскаль тіліндегі жазылуынан қандай айырмашылығы бар?
3. Алгебралық өрнектер құрамына нелер енеді?
4. Бірер өрнекке бірдей амалдар қатысса, олардың орындалу тәртібі қандай болады?
5. Амалдардың орындалу тәртібін өзгерту үшін нелер пайдаланылады?
6. $\text{Trunc}(4.7)=\text{Round}(4.7)$ дұрыс жазылған ба? Жауабыңды түсіндір.
7. $\sin x-c$ жазуы Паскаль тілінде неліктен қате деп саналады?
8. 2^{*-v} жазуы Паскаль тілінде дұрыс жазылған ба? Жауабыңды түсіндір.
9. $\text{sqr}(\text{abs}(x +\sin(x))-pi)$ өрнектегі амалдардың орындалу тәртібін түсіндір.

Жаттығулар

1. Осы тақырыптың 1-есебіндегі жылжымалы нүктелі сандарды жылжымайтын нүктелі сандарға өткіз.
2. Төмендегі алгебралық өрнекті Паскаль бағдарламалау тілінде жаз.

а) $ax+b$;	ә) x^2-23 ;	б) ax^2+bx+c ;
в) $a^4x^3-(1-y^2)^2$;	г) $\frac{a+5}{2b}$;	д) $8(a+b^2c)$.
3. Төмендегі өрнектерді Паскаль бағдарламалау тілінде жаз.

а) $25^{20}+ 1-y^2 $;	ә) $[5m]+\{100b\}$;	б) $x\sin a+y\cos b-5^2$;
в) $\sin\sin x+\cos\cos y$;	г) $21-\sqrt{2011-b^2}$.	
4. Паскаль тілінде жазылған төмендегі өрнектер арасынан қате жазылғанын тап.

а) $2*a+b$;	ә) $\text{sqr}(x*b^2)$;	б) $\sin(-3*x)$;
в) $\sin((a+b+\cos(x)))$;	г) $2*(-b)+a.2$.	
5. Паскаль тілінде жазылған төмендегі өрнектерді кәдімгі жазу түріне өткіз.

а) $a*(\text{Sqr}(x)+1)$;		
ә) $\sin(x*x*x-\text{sqr}(\text{sqr}(x))+5)$;		
б) $pi*h*(\text{sqr}(r1)+\text{sqr}(r2)+r1*r2)/3$.		

20-сабақ. Стандартты функциялар мен процедуралар, алгебралық өрнектер тақырыбын қайталау

1. Төмендегі өрнектерді Паскаль бағдарламалау тілінде жаз.

а) $\frac{x-y}{x^2-y^3}$

ә) $\frac{x+y}{xyz} + \sin^2 x;$

б) $(5a^2 + 2x) + \frac{3x}{a^3} + tg^5 a^3 ;$

в) $\cos^3 \sin^2 x + \cos a^5;$

г) $\sqrt{5+x} - \sqrt{z} \frac{3x}{a^3} + \sqrt[3]{a}.$

2. Төмендегі Паскаль тілінде жазылған өрнектердің мәнін есепте.

а) $\text{sqr}(\text{trunc}(4.95));$

ә) $\text{trunc}(\text{int}(4.95)+0.7);$

б) $\text{round}(\text{trunc}(3.5)+0.7);$

в) $3+\text{frac}(12.5);$

г) $\text{sqr}(\text{sqr}(16));$

д) $\text{sqr}(\text{sqr}(256)+9);$

е) $\text{sqr}(5-\text{abs}(-5));$

ж) $\text{abs}(-\text{sqr}(6)).$

3. $a = 5, b = 4$ болса, Паскаль тілінде жазылған мына өрнектердің мәнін есепте.

а) $\text{abs}(a+b-a*b);$

ә) $\text{sqr}(a+b-a*b)-110;$

б) $\text{round}(a/b+0.3)+9;$

в) $3+\text{frac}(b/a);$

г) $\text{sqr}(\text{sqr}(a)-b*b);$

д) $\text{sqr}(\text{sqr}(a+b)+6);$

е) $\text{sqr}(a-\text{abs}(b-a));$

ж) $\text{abs}(9-\text{sqr}(a*b+a)).$

4. Паскаль тілінде жазылған мына өрнектердің мәні қандай түрде тұрақты болуын анықта.

а) $\text{abs}(-\text{sqr}(2011));$

ә) $\text{abs}(\text{sqr}(2))+19;$

б) $\text{frac}(\text{abs}(-20));$

в) $\text{int}(1.9)*\text{trunc}(0.2).$

21-сабақ. Меншіктеу және деректерді экранға шығару операторы

Паскаль бағдарламалау тілі, әдетте сипатталған айнымалылар үшін зердеден (жад) орын ажыратып, олардың түріне қарай бастапқы мәндерін жазып қояды:

Айнымалы түрі	Бастапқы мәні	Айнымалы түрі	Бастапқы мәні
барша бүтін санды	0	барлық рас санды	0.0000000000e+00
char	' ' (probel)	boolean	FALSE
string	" (бос жол)	string[7]	" (бос жол)

Меншіктеу операторы. Меншіктеу операторы айнымалыларға мән беру үшін қолданылады. **U :=** белгі арқылы өрнектеледі. Меншіктеу операторының жалпы көрінісі мынадай:

айнымалы := өрнек;

Бұл оператор орындалғанда төмендегі жұмыстар орындалады:

- 1) өрнектің мәні есептеледі;
- 2) өрнектің мәні айнымалыға беріледі, яғни жадының айнымалы үшін ажыратылған бөлігінде (айнымалының «ескі» мәні өшіп кетеді) өрнектің мәні жазылады.

1-есеп. Төмендегі бағдарламаны орындаудың нәтижесінде **a** атты айнымалының мәні **22** санына тең болады.

```
var a: integer;
begin
  a := 22;
End.
```

2-есеп. Төмендегі бағдарламаны орындаудың нәтижесінде **жеміс** атты жолдық айнымалының мәні **«алма»** сөзіне тең болады.

```
var жеміс : string;
begin
  жеміс := 'olma';
End.
```

3-есеп. Бұл есептен **a** мен **b** айнымалылардың мәні қалай өзгертетіні айқын көрінеді.

```
var a,b,m: integer;
begin
  a := 8; {a -ның мәні 8-ге тең болады}
  b := a*5; {b -ның мәні a*5=8*5= 40 -қа тең болады}
  b := b+10; {енді b -нің мәні b+10= 40+10= 50 -ге тең болады}
  m:=m*b; {m -ның бастапқы мәні берілмегендіктен, 0 деп алынады, демек, m-ның мәні 0*50=0 -ге тең болады}
End.
```

Жоғарыда келтірілген есептердегі айнымалылар түрлі мәндерді меншіктейді. Бірақ біз олардың нәтижесін көрмедік. Өйткені олар компьютер жадында қалып, экранға шығарылған жоқ. Деректерді компьютер экранына шығару үшін **шығару операторы** пайдаланылады. Паскаль тіліндегі шығару операторының мынадай екі түрлі көрінісі бар:

Write(шығару тізімі) және **WriteLn(шығару тізімі)**

мұндағы **Write** (ағыл. — жазу) және **WriteLn** — Паскальдің көмекші сөздері; шығару тізімі — өзара үтірмен дараланған және экранға шығаруға тиісті болған өрнек, айнымалы яки тұрақтылар тізбегі. Шығару тізіміне өрнек қатысатын болса, әуелі өрнектің мәні есептеліп, алынған нәтиже экранға шығарылады. Шығару тізіміндегі тұрақтылар символдық немесе жолдық болса, сөзсіз апостроф ішіне алынуға тиіс болады.

Write және **WriteLn** операторларының ерекшелігі сол, **Write** операторының көмегімен деректер экранға шығарылған соң, курсор экранның сол жолында қалады да, экранға шығарылатын кейінгі деректер сол жолға курсор тұрған жерден бастап шығарылады. Ал **WriteLn** операторы бойынша деректер экранға шығарылған соң, курсор кейінгі жолдың басына көшеді.

<p>4-есеп. begin write('Жайна, '); write('гүлде '); write('Отан-анам!'); End.</p>	<p>Бағдарлама орындалған соң компьютер экранында Гүлдей жайна Отан-анам! жазуы пайда болады</p>
<p>5-есеп. begin writeln('Жайна, '); writeln('гүлде '); write('Отан-анам!'); End.</p>	<p>Бағдарлама орындалған соң компьютер экранында Гүлдей жайна Отан-анам! жазуы пайда болады</p>
<p>6-есеп. program меншіктеу_шығару; var a,b:integer; begin a:=23; b:=a+21; write('b -ның мәні ', b, ' -ға тең'); End</p>	<p>Бағдарлама орындалған соң компьютер экранында b -ның мәні 44 -ке тең болады жазуы пайда болады.</p>

Деректер шығаруда **шығару форматын** көрсетуге болады. Өйткені, бұл формат шығарылатын деректердің көрінісін (форматын) белгілейді. Ол үшін шығарылатын айнымалыдан кейін «:» (қос нүкте) белгісі қойылады. Мысалы, а – нақты айнымалы болса, шығару форматында екі параметр – шығарылатын санға ажыратылған бөлмелер саны көрсетіледі.

Мысалы, **WriteLn(a:10:2)**; операторы а-ның мәнін шығару үшін 10 бөлме ажыратады, солардың бір бөлмесі нүкте үшін, ал екі бөлмесі бөлшек бөлімі үшін бөлінеді. Егер сан бүтін болса, шығару форматында бір параметр – шығарылатын санға ажыратылған бөлмелер саны көрсетіледі. Мысалы, **WriteLn(b:6)**; жолдық және символдық айнымалылар үшін шығару форматы олардың мәнін шығаруға ажыратылған орынның (өрістің) ұзындығын анықтайды.

Шығарылатын сан я мәгін оған ажыратылған орынның оң шетіне тегістеліп шығады. Мысалы, a = 3.24; болса, WriteLn('a = ',a:6:2); операторы экранда a = 3.24 жазуын шығарады (= белгісінен кейін екі бос орын (пробел) қалады).

Егер шығару форматында бөлінген жер айнымалы мәнінің «ұзындығынан» кіші болса, шығару форматы жойылып, айнымалының мәні экранға сыйғанша шығарылады. Тек нақты санның бөлшек бөлігінің форматы ғана жойылады. Нақты санды шығарған кезде форматы көрсетілмесе, ол экранға экспоненциал түрінде шығарылады.

<p>7-есеп. var a,b : real; Begin a:=3.24; b:=5; writeln('a=',a); writeln('b=',b); End.</p>	<p>Компьютер экранында</p> <p>a=3.2400000000E+00 b=5.0000000000E+00</p>
<p>8-есеп. var a,b : real; Begin a:=3.24; b:=5.3; writeln('a=', a:6:2); writeln('b=',b:1:0); End.</p>	<p>Компьютер экранында</p> <p>a= 3.24 b=5</p>

Жоғарыда келтірілген екі есептегі экранға шығарылатын *a* мен *b* айнымалылардың мәндері бірдей, бірақ олардың көрінісінде үлкен айырмашылық бар. 8-есептегі экранға шығарылатын ақпарат әрине, 7-есептегіге қарағанда, анық әрі түсінікті болады. Санның бөлшек бөлігіндегі мәнді цифрлар санын анық білмей отырып, экранға қате нәтиже шығармау үшін, шығару форматын абайлап пайдалану керек.

Есте сақта: экрандағы нәтижені қарап шығу үшін **ALT+F5** клавиштері бірге басылады.



Сұрақтар мен тапсырмалар

1. Меншіктеу операторы қандай қызметті атқарады?
2. Меншіктеу операторының жалпы көрінісін мысалдармен түсіндір.
3. Деректерді экранға шығаратын оператордың жалпы көрінісі қандай?
4. Write және Writeln операторлар айырмашылығын мысалдар арқылы түсіндір.
5. Деректерді экранға шығару операторының мүмкіншіліктерін мысалдармен түсіндір.
6. Деректерді шығару операторындағы апостроф ішіне жазылғандар нені білдіреді?
7. Шығару форматы деген не, ол не үшін қолданылады?

Жаттығулар

1. Мына өрнектерді меншіктеу операторының жәрдемімен жаз.
a) $a = 48; b = 51;$ ә) $x = 0; a = 3,6x + \sin x;$

б) $g = 4; g = g + 16;$

в) $a = 9,81; m = 50; F = m a;$

г) $x = 1; y = \frac{x - 63}{21 - 7x};$

д) $z = 25; z = \sqrt{z}.$

2. Шығару операторлары орындалған соң, нәтиженің экранда қалай көрінетінін жаз.

а) `write('a='); write(2+3);`

`write('='); write('2+3');`

ә) `writeln('a='); write(2+3);`

`write('='); writeln('2+3');`

б) `write('a='); writeln(2+3);`

`write('='); writeln('2+3');`

в) `write('a='); write(5);`

`writeln('='); write('2+3');`

г) `write('a='); writeln(5);`

`write('='); write('2+3');`

д) `writeln('a='); writeln(5);`

`writeln('='); write('2+3');`

3. Паскальда жазылған мына бағдарлама үзінділеріндегі барлық айнымалылардың аралық мәнін және экранға шығатын нәтижесін анықта.

а) $a := -\cos(\pi) - \sin(\pi/2); x := x * x + a;$

`writeln('a=', a, 'x=', x);`

ә) $a := 'Мен'; g := 'тәуелсіз';$

$b := 'Өзбекстан'; m := 'перзентімін!'$

`write(a, g, b, m);`

б) $a := 9; b := a + a; a := a * a - b;$

`write('a= ', a); write(' b=', b).`

22-сабақ. Меншіктеу және деректерді экранға шығару операторлар тақырыбын қайталау

1. Мына өрнектерді меншіктеу операторының жәрдемімен жаз.

а) $y = \frac{x - 21}{7 - x^{63}};$

ә) $a = 3,6x + \sin x;$

б) $z = \sqrt{x - 5y + xtgx};$

в) $S = \pi r^2;$

г) $F = ma;$

д) $S = \frac{ah}{2}.$

2. Шығару операторының нәтижесі қандай болатынын анықта.

- а) `a:=123.45;` ә) `a:=123.45;`
 `write('a=', a:2:1);` `write('a=', a:5:1);`
 б) `a:= '2011';` в) `a:= '2011';`
 `writeln(a:3, ' yil':3);` `writeln(a:4, ' yil':5).`

3. Бағдарламадағы айнымалылардың мәні түріне сай болуы үшін, сұрақ белгісінің орнына қажетті стандартты функция жазып, экранға шығарылатын нәтижені анықта.

- а) `var a, b, c: integer;`
 `begin a:=25; b:=?(sqrt(a)); c:=?(a/b);`
 `writeln(a, ' ', b, 'c= ', c);`
 `End.`
- ә) `var x, y, z: word;`
 `begin x:=?(?(-7.21)); y:=?(sqrt(x*x));`
 `z:=?(x+y-100); write(z-x, y);`
 `End.`

23-сабақ. Деректерді жадына диалог әдісімен енгізу операторы

Паскальда айнымалыларға мән берудің меншіктеу операторынан пайдаланудан басқа әдістері де бар. Солардың бірі **деректерді жадына диалог әдісімен енгізу** деп аталып, **енгізу операторының** жәрдемімен жүзеге асырылады. Енгізу операторы бағдарламаның жұмысы барысында айнымалыларға компьютер клавиатурасы мән беру үшін қолданылады. Енгізу операторының мынадай екі көрінісі бар:

Read (енгізу тізімі); және ReadLn(енгізу тізімі);

мұндағы **Read** (ағыл. read — оқу) және **ReadLn** — Паскальдің көмекші сөздері, ал **енгізу тізімі** — бір айнымалы, я бір-бірінен үтірмен ажыратылған бірнеше айнымалылар тізбегі. Мысалы: `Read(a); Read(alfa,betta); ReadLn(_name);`.

Енгізу операторы бағдарламаның жұмысын тоқтатып, тізімдегі айнымалыларға клавиатура арқылы мән беруді күтеді. Егер тізімде бірнеше айнымалы болса, олардың мәндерін бір-бірінен пробелмен ажыратып енгізуге яки **ENTER** клавишін басып енгізуге болады. Екі жағдайда да соңғы айнымалының мәнін енгізген соң **ENTER** клавишін басу керек.

Read және **ReadLn** операторлар арасындағы айырмашылық мынадай: бір немесе бөлек жазылған бірнеше **Read** операторының көмегімен енгізілетін айнымалылар мәнін бір жолға пробелдермен ажыратып енгізуге болады. Ал **ReadLn** операторы тек өзінің тізімінде келтірілген айнымалылар мәнін бір жолға пробелдермен ажыратып енгізуге мүмкіндік береді. Сондықтан **ReadLn** операторындағы тізім аяқталған соң, кейінгі енгізу операторын істету үшін сөзсіз **ENTER** клавишін басу қажет.

1-есеп.

```
Var a,b:Integer;
Begin
Read(a);
Read(b);
WriteLn('a+b=', a+b);
End.
```

2-есеп.

```
Var
a,b:Integer;
Begin
Read(a, b);
WriteLn('a+b=', a+b);
End.
```

Екі есепте де мән енгізу төмендегі екі әдістің біреуімен жүзеге асырылуы мүмкін.

1-әдіс: бағдарлама іске түсірілген соң, экранда жол басында курсор шығады да, бағдарлама **a**-ның мәнін енгізуді күтіп тұрады. Мысалы, **a**-ның мәні ретінде 10-ды енгізген соң, пробелді басып, **b**-ның мәні ретінде 11-ді енгіземіз. Енді **ENTER** клавишін басса, экранда мыналар көрінеді:

```
10 11
a+b=21
```

2-әдіс: бағдарлама іске түсірілген соң, экранда жол басында курсор шығады да, бағдарлама **a**-ның мәні енгізілуін күтіп тұрады. Мысалы, **a**-ның мәні ретінде 10-ды енгіземіз, соңыра **ENTER** клавишін басып, **b**-ның мәні ретінде 11-ді енгіземіз. Енді **ENTER** клавишін басса, экранда мыналар көрінеді:

```
10
11
a+b=21
```

3-есеп.

```
Var a,b,g,m:Integer;
Begin
Read(a, b);
Read(g); m:=a+g+b;
WriteLn('Нәтиже=', m);
End.
```

4-есеп.

```
Var a,b,g,m:Integer;
Begin
ReadLn(a, b);
Read(g); m:=a+g+b;
WriteLn('Нәтиже=', m);
End.
```

<p>3-есепте де мән енгізуді 1–2 әдістің біреуімен жүзеге асыруға болады.</p> <p>4-есепте a және b айнымалылардың мәнін пробелмен немесе ENTER клавиштері арқылы енгізуге болады. Айнымалы g -ның мәнін енгізу үшін, b айнымалының мәнін енгізген соң ENTER клавишін басу қажет. Бұл жағдайда экранда мыналардың бірі болуы мүмкін:</p>	
<p>10 11</p> <p>12</p> <p>Нәтиже= 33</p>	<p>10</p> <p>11</p> <p>12</p> <p>Нәтиже= 33</p>

Айнымалыларға мән беруде меншіктеу операторы қолданылған болса, бағдарлама бір мән үшін орындала береді, яғни айнымалының мәні жадына бір рет енгізіледі. Бірер айнымалының мәнін ауыстыру үшін әр кез бағдарламаға кіріп, өзгерту керек болады. Бағдарламадағы бір айнымалыға мән беруде енгізу операторы қолданылған болса, онда бағдарламаны іске қосып, айнымалының мәнін клавиатурадан енгізеді, яғни **диалог әдісімен** өзгертеді.

Диалог әдісінің біраз қолайсыздығы бар, яғни мәннің қай айнымалыға енгізіліп отырғанын есте сақтауға тура келеді. Бұл қолайсыздықтан қашу үшін **Write** я **Writeln** операторларын пайдалануға болады. Мысалы, `Write('a= '); ReadLn (a); Write('b= '); ReadLn (b);` жазылған болса, мәннің қай айнымалыға енгізіліп отырғаны экранда көрініп тұрады.

`ReadLn` операторының тағы бір мүмкіншілігі бар. Бәріне белгілі, Паскаль бағдарламалау тілі амалдарды өте тез орындайды, сондықтан оның нәтижесін көруге үлгермейсің. Бұрын айтылғанындай, экрандағы нәтижені қарап шығу үшін **ALT+F5** клавиштерін бірге басу керек. Барша нәтиже экранға шығарылған соң, нәтижені қарап шыққанымызша бағдарламаны күттіру үшін, **end.** операторынан алдын әуелі тізімсіз **ReadLn** операторын жазу жеткілікті. Ондай жағдайда бағдарламаның жұмысын аяқтау үшін **ENTER** клавишін басады. Әрине, бұл оператор күтілген нәтижені шығаруы үшін, алдын ала жазылған соңғы енгізу операторы да **LN** қосымшасымен бірге жазылған болу керек.



Сұрақтар мен тапсырмалар

1. Деректерді диалог әдісімен енгізу операторының қызметін түсіндір.
2. Неліктен бұл әдіс деректерді диалог әдісімен енгізу деп аталады?

3. Деректерді енгізетін операторлар бір-бірінен несімен ерекшеленеді?
4. Айнымалыларға мән беруде енгізу операторының қолайлы жағын түсіндір.
5. Енгізу операторындағы мәні беріліп отырған айнымалыны қандай әдіспен білу оңай?
6. Нәтижені қарап шығуда ReadLn операторының қолайлылығын түсіндір.

Жаптықулар

1. N санының квадратын Read операторының жәрдемімен N -ның 10; 11; 12; 13; 14; 15 мәндерінде есептейтін бағдарлама жаса.

2. Төмендегі бағдарламадағы a айнымалының «ӨЗБЕКСТАНЫМ»; «ОТАНЫМ»; «АТА МЕКЕНІМ» мәндерін жадына қолайлы әдіспен енгізіп, нәтижесін шығар.

```
Var a, b, g: string;
```

```
Begin
```

```
  b:= ' – СӘЖДЕ ОРНЫНДАЙ; g:= ' ҚАСИЕТТИ!';
```

```
  write(a, b, g); ReadLn;
```

```
End.
```

3. «Матиз» автомобилі орнынан қозғалып, T секундта S метр жол басып өтті. Оның метрдің секундқа қатынасымен (м/с) өрнектелетін орташа жылдамдығын төмендегі мәндерде есептейтін бағдарламаны деректерді диалог әдісімен енгізу арқылы жаса (бағыт: $V=S/T$).

а) $T= 10; S= 150;$

ә) $T= 12; S= 200;$

б) $T= 20; S=400;$

в) $T= 45; S= 900.$

24-сабақ. Деректерді жадына диалог әдісімен енгізу операторы тақырыбын қайталау

1. Төмендегі бағдарламаға a айнымалының «азат»; «абат» мәндерін компьютер жадына қолайлы әдіспен енгізіп, нәтижесін шығар.

```
Var a, b, g: string;
```

```
Begin
```

```
  b:= 'Бізден'; g:= ' отан қалсын!'; write(b, a, g); ReadLn;
```

```
End.
```

2. Төмендегі бағдарламаның сұрақ белгісінің орнына ағымдағы жылды m айнымалының жәрдемімен енгізіп, тәуелсіздігімізбен құттықтайтын нәтиже шығар.

```
Var a, b, g: string; m: word;
```

```
Begin
```

```
write('Ағымдағы жылды енгіз: '); ?;
```

```
a:= 'Тәуелсіздіктің '; g:= 'жылдығымен '; b:='құттықтаймыз!';
```

```
WriteLn(a); WriteLn(m-1991, g); write(b); ReadLn;
```

```
End.
```

3. Егер денеге әсер ететін күш F , ал оның жылдамдығы a болса, төмендегі мәндерде дененің массасын есептейтін бағдарламаны енгізу операторын пайдаланып жасап, нәтижесін шығар (бағыт $m=F/a$).

а) $F=15, a=55$;

ә) $F = 55, a = 15$;

б) $F=10, a=100$;

в) $F = 100, a = 10$;

4. $a=19, b=2, d=1950$ мәндерді жадына қолайлы әдіспен енгізіп, төмендегі өрнектердің мәнін есептейтін бағдарлама жаса.

а) $y = a + b^2 + ad$;

ә) $t = \sqrt{a+b} - \sqrt[3]{d-a}$;

б) $s = b \cos a + \sin d$;

в) $n = \pi d^2 + ab$.

5. Жақтары a, b, c болған үшбұрыштардың ауданын есептейтін бағдарламаны енгізу операторын пайдаланып жасап, нәтижелерін шығар.

а) $a=5, b=7, c=4$;

ә) $a=8, b=6, c=10$;

б) $a=3, b=4, c=5$;

в) $a=10, b=8, c=10$.

6. x -тің $-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$ мәндерін жадына қолайлы әдіспен енгізіп, $y=23x+1$ функцияның мәнін есептейтін бағдарлама жасап, оның нәтижесін шығар.

7. x -тің $-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$ мәндерін жадына қолайлы әдіспен енгізіп, $y = 21x^2 + 7x + 1963$ функцияның мәнін есептейтін бағдарлама жасап, нәтижесін шығар.

25-сабақ. Мәтін режиміндегі экранмен жұмыс істеу

Біз бұрынғы сабақтарда ақпаратты экранға шығару әдістерімен таныстық. Бұл әдістерде ақпараттың әрқайсысы экранға өзінен алдын шығарылған ақпараттан кейін шығарылады. Бірақ Паскаль бағдарламалау тілінің ақпаратты экранның көрсетілген жерінен шығару мүмкіншілігі де бар.

Бұдан тыс, экранға шығарылатын белгілерді, сондай-ақ олардың фонын түрлі түсті етіп шығаруға да болады. Бұрын айтылғанындай, экранмен жұмыс істеуге арналған Паскальдің процедуралары мен функциялары **Crt** модулінде орналасқан. Сондықтан оларды пайдалану үшін осы модульге жүгіну керек болады. Бұл үшін бағдарламаның басында **Uses Crt**; командасы беріледі.

Егер **Crt** модулі экранға түсті ақпарат шығаруға мүмкіндік беретін болса, қандай түстерді пайдалану мүмкіндігін біліп алайық. Паскаль тілінде, негізінде, 16 түрлі түс қолдануға болады. Олар 0-ден 15-ке дейінгі бүтін сандармен кодтандырылған. **Crt** модулінде бұл сандарға тиісті константалар да белгіленген. Бұл константалардың атаулары оларға тән түстердің ағылшын тіліндегі көрінісімен бірдей болады.

Төмендегі кестеде Паскаль тілінде қолданылатын түстердің кодтары мен оларға тән константалардың аттары келтірілген:

Түс	Коды	Константа аты	Түс	Коды	Константа аты
Қара	0	Black	Төк сұр	8	DarkGray
Көк	1	Blue	Ауа көк	9	LightBlue
Жасыл	2	Green	Аш жасыл	10	LightGreen
Көгілдір	3	Cyan	Аш көгілдір	11	LightCyan
Қызыл	4	Red	Аш қызыл	12	LightRed
Күлгін	5	Magenta	Аш күлгін	13	LightMagenta
Қоңыр	6	Brown	Сары	14	Yellow
Аш сұр	7	LightGray	Ақ	15	White

Белгілі бір түсті таңдау үшін Паскальдің арнайы процедураларымен жақшаның ішінде сол түстің коды яки оған тән константаның аты көрсетіледі. Мәтін мен фонға арналған түстердің түрліше таңдалғаны жөн болады, ондай болмаса, мәтін фон ішінде көрінбей қалады. Мәтін мен фон үшін белгіленген соңғы түстер сәйкесінше мәтіннің ағымдағы түсі мен фонның ағымдағы түсі делінеді. Егер әуел бастан бірер түс таңдалмаған болса, ағымдағы түс ретінде, мәтін үшін ашық ақ, ал фон үшін қара түс саналады.

Енді **Crt** модулі құрамына енген кейбір процедуралармен танысамыз. Мәтінмен және мәтін фоны түсімен істейтін процедуралар мыналар:

Мәтін түсін белгілейтін процедуралар	Мәтін фонының түсін белгілейтін процедуралар
TextColor(түс);	TextBackGround(түс);

мұндағы түс — айнымалы яки тұрақты шама болып, таңдалған түстің яки константаның атын білдіреді.

<p>1-есеп. Uses Crt; Begin TextColor(14); {яки TextColor(yellow)} WriteLn('Бұл мәтін экранға сары түсті болып шығады'); End.</p>	<p>Бұл мәтін экранға сары түсте шығады</p>
--	---

2-есеп.

```
Uses Crt;
Begin
  TextColor(Yellow); TextBackGround(Blue);
  WriteLn('Бұл мәтін экранға сары түсті болып шығады');
End.
```

Бұл мәтін экранға сары түсті болып шығады

Кейде экрандағы мәліметтер көбейіп кеткендіктен қажеттісін табу қиынға соғады. Осындай жағдайда қолданылған **ClrScr** процедурасы экранды тазалап, курсорды экран басына (сол жақтағы жоғары бұрышына) орнатады. Егер бағдарламадағы ClrScr процедурасы мәтін фоны түсінен кейін жазылса, онда экран мәтін фоны түсіне боялады.

3-есеп.

```
Uses Crt;
Begin
  ClrScr; {экран тазаланып, курсор экран басына орнатылады}
  TextColor(14); TextBackGround(2);
  WriteLn('Бұл мәтін экранға көк фонда сары түсті болып шығады');
End.
```

Бұл мәтін экранға көк фонда сары түсте шығады

4-есеп.

```
Uses Crt;
Begin
  TextColor(14); TextBackGround(2);
  ClrScr; {экран тазаланып, экран түсі жасылға боялады, курсор
экран басына орнатылады}
  WriteLn('Бұл мәтін экранға сары түсті болып шығады');
End.
```

Бұл мәтін экранға сары түсте шығады

Бағдарлама нәтижесінің көрінісі «сұлу» болсын десен, түрлі түстерді пайдаланумен бірге оны экранның керекті жерінен шығарудың маңызы зор. Бұл үшін курсорды экранның керекті жеріне орнату қажет. Паскаль тілінде бұл міндетті **GotoXY** процедурасы жүзеге асырады. Оның жалпы көрінісі мынадай: **GotoXY(A,B)**; мұндағы **A** мен **B** – бүтін сандық айнымалы яки тұрақты болып, **GotoXY(A,B)** процедурасы курсорды экранның **A**-баған мен **B**-жол қиылысқан жеріне орнатады. Мәтін жағдайындағы экранның өлшемі, негізінде, 80×25 , яғни 80 баған мен 25 жолдан тұрады (арнайы операторлар көмегімен бұл өлшемді өзгертуге де болады). Сондықтан осы процедурада $1 \leq A \leq 80$ және $1 \leq B \leq 25$ шарты орындалуға тиіс.

5-есеп.

```
Uses Crt;
Begin
  ClrScr; {экран тазаланды}
  GotoXY(22,12); {курсор 22-баған және 12-жол қиылысқан жер-
де орнатылды}
  Write('Бұл мәтін экранның ортасынан шығады');
End.
```

Бұл мәтін экранның ортасынан шығады

Бұл бағдарламада экранға шығарылатын мәтін 36 белгіден тұрады. Мәтінді сол және оң шеттен бірдей орын қалдырып, экранның ортасынан шығару үшін, **GotoXY** процедурасындағы **A** мен **B**-ның мәні былай есептеп шығарылды:

$$A = \lfloor 25/2 \rfloor = 12, B = \lfloor (80-36)/2 \rfloor = 22.$$



Сұрақтар мен тапсырмалар

1. Мәтін режимінде экранмен жұмыс істеу үшін Паскаль тілінің қандай модулі пайдаланылады?
2. Паскаль тілінде негізінен, неше түрлі түс пайдалануға болады?
3. Мәтін түсін қандай процедура арқылы өзгертуге болады?
4. Мәтін фонының түсін қандай процедура арқылы өзгертуге болады?
5. **ClrScr** процедурасы қандай мақсатпен қолданылады? Жауабыңды мысалдармен түсіндір.
6. Мәтін күйіндегі экранның өлшемі неше жол мен бағаннан тұрады?
7. Мәтінді экранның қалаған жеріне шығаруға бола ма? Жауабыңды дәлелде.

Жаттығулар

1. Төмендегі бағдарламалардың нәтижесінде экранның түсі, мәтін фонының түсі, мәтін түстері қандай болатынын, сондай-ақ мәтіндердің орнын анықта.

```
Uses crt;
Begin
  textbackground(yellow); WriteLn('Өзбекстан'); clrscr; textcolor(4);
  write('келешегі '); textbackground(blue); WriteLn('ұлы'); textcolor(2);
  write('MEMЛЕКЕТ!'); ReadLn;
End.
```

2. «Өзбекстан Конституциясы — еркіндік сақшысы» мәтініне қызыл, ал мәтін фонына көк түсті тандап, экранға шығар.

3. Төмендегі бағдарламаға сондай процедуралар қос, барлық мәтін көк түсті, мәтін фондары сары түсті болып экранға шықсын. Бағдарламаның жұмысын түсіндір.

Var a,b: string; m, s : real;

Begin

a:= 'Квадраттың қабырғасын енгіз: '; b:= 'Квадраттың ауданы: ';

Write(a); ReadLn(m); s:=sqr(m); write(b, s:8:2, ' квадрат бірлік);

ReadLn;

End.

4. A мен B айнымалылардың берілген мәнін енгізіп, төмендегі бағдарламаның жұмысын түсіндір.

Uses Crt;

Begin ClrScr; write('A= '); ReadLn(a); write('B= '); ReadLn(b);

GotoXY(A,B); Writeln('Кітап білім ошағы'); ReadLn;

End.

а) A=1, B=1;

ә) A=1, B=8;

б) A=8, B=1;

в) A=8, B=8;

г) A=25, B=25;

д) A=100, B=10;

5. Экранға есімді, фамилиянды, әкенің аты-жөнін түрлі 3 түспен, мәтінді фон түсінің 3 түрімен және экранның түрлі жеріне шығар.

26-сабақ. Мәтін режиміндегі экранмен жұмыс істеу тақырыбын қайталау

1. «Ата-бабалар мұрасын қадірлейік» деген мәтінге көк, ал оның фонына жасыл түсті таңдап, оны экранға шығар.

2. «Отанды сүю имандылық!» мәтінін экранның оң жағындағы 12- жолында қызыл фонда жасыл түсті етіп шығар.

3. «Ең жоғары жол, сол жақтан», «Ең жоғары жол, оң жақтан», «Ең жоғары жол, ортадан», «Ең төменгі жол, оң жақтан», «Ең төменгі жол, оң жақтан», «Ең төменгі жол, ортадан», «Орталықтағы жол, сол жақтан», «Орталықта жол, оң жақтан», «Орталықтағы жол, ортадан» деген мәтіндерді экранның мәтін мазмұнына тиісті жерінде шығуын қамтамасыз ететін бағдарлама жаса.

4. «Су – тіршілік көзі» деген сөйлемді экранға түрлі 5 түсте әрі түрлі жерде шығаратын бағдарлама жаса.

5. Экранға 5 сыныптасыңның атын түрлі түсті етіп, ал экранның түсін сары етіп шығар.

27-сабақ. Сызықтық бағдарламалар жасау

Әдетте, сызықтық алгоритмдердің бағдарлама түріндегі жазылуы **сызықтық бағдарлама** деп аталады.

Демек, сызыктык бағдарламадағы барлық процедуралар рет-ретімен келіп орындалады да, ешқандай шарт тексерілмейді.

1-есеп. Радиусы $R=9$ -ге тең шеңбердің ұзындығын есептеу бағдарламасы жасалсын және $R=9$ бірлік мәні үшін орындалсын.

Шешуі. Шеңбердің ұзындығын есептеу формуласын еске түсіреміз: $L=2\pi R$. Паскаль бағдарламалау тілінде ол $L := 2*\pi*R$ түрінде жазылады. Бағдарламаға бір тұрақты π , екі айнымалы – R мен L қатысады. Есептің шартына қарағанда $R=9$, яғни бүтін сан. Сондықтан R айнымалының түрі Integer болады. Ал шеңбердің ұзындығы – L көбейтіндісіне π қатысқандықтан, оның түрі әрине, нақты (Real) болады. Айтылғандарды ескеріп, бағдарлама құрастырылады:

```
Program шеңбер_ұзындығы;
  Var R:Integer; L:Real;
  Begin
    r := 9; L := 2*pi*R; WriteLn('L=',L,' бірлік.');
```

End.

Бағдарламаны іске қосқан соң (**Ctrl+F9** клавиштерін бірге басқан соң немесе **Run** менюсінен **Run** бөлімін тандаған соң), экранда мынадай нәтиже пайда болады: **L= 5.6548667765E+01 бірлік.**

Бағдарламадағы L – нақты түрлі болғандықтан, нәтиже экспоненциал түрінде шықты. Егер шығару операторында $L:7:2$ форматын пайдаланса, радиусы 9 бірлікке тең шеңбердің ұзындығы 56,54 бірлікке тең екендігі көрінеді. Осы бағдарламаның жәрдемімен ерікті бүтін сандық радиусы бар шеңбердің ұзындығын есептеуге болады. Тек әр кез бағдарламадағы R -дің мәнін өзгертіп тұру керек болады. Өзгерістерді бағдарламаға әр кез енгізбеу үшін, R -дің мәні енгізу операторымен беріледі. Радиустың мәні үнемі бүтін сан бола бермейтінін ескеріп, оны **Real** түрдегі айнымалы ретінде өрнектейді. Бұларды ескеріп, мына бағдарламаны жасаймыз:

Бағдарлама	Экрандағы нәтиже
<pre>Program шеңбер_ұзындығы; Var r, L : Real; Begin Write('Радиусты енгіз : '); ReadLn(r); L := 2*pi*r; WriteLn('L= ',L,'бірлік'); ReadLn; End.</pre>	<p>Радиусты енгіз : 9 L=5.6548667765E+01 бірлік</p>

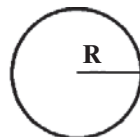
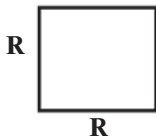
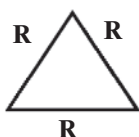
Бұл бағдарламаны іске қосқан соң, экранға «Радиусты енгіз» жазу шығады да, курсор сол жолда қалады. ReadLn операторы бағдарламаның жұмысын тоқтатып, R айнымалыға мән беруді күтеді. Клавиатура арқылы радиустың 9 сандық мәнін 9 енгізіп, **ENTER** клавишін басса, R айнымалының мәні 9-ға тең деп алынып, бағдарламаның жұмысы жалғаса береді. Нәтиже-

де компьютер экранында издәліп отырған нәтиже пайда болады. Бағдарламаны қайта-қайта істетіп, радиустары түрліше шеңберлердің ұзындығын есептеуді ұйымдастыруға болады.

2-есеп. Кабырғалары сәйкесінше a, b, c болған ерікті үшбұрыштың ауданын Герон формуласы бойынша есептейтін бағдарлама жасап, $a = 3, b = 4, c = 5$ мәндерде есепте.

I әдіс	II әдіс
<pre> Program Үшбұрыштың_ауданы; Var a,b,c:Integer; {үшбұрыштың кабырғалары} yp,s:Real; {yp-жарты периметр, s-аудан} Begin a:=3; b:=4; c:=5; yp:=(a+b+c)/2; s:=sqrt(yp*(yp-a)*(yp-b)*(yp-c)); WriteLn('S= ',s,' квадрат бірлігі'); ReadLn; End. </pre>	<pre> Program Үшбұрыштың_ауданы; Var a,b,c:Integer; {үшбұрыштың кабырғалары} yp,s:Real; {yp-жарты периметр, s-аудан} Begin Write('a,b,c -тің мәндерін енгіз '); ReadLn(a,b,c); yp:=(a+b+c)/2; s:=sqrt(yp*(yp-a)*(yp-b)*(yp-c)); WriteLn('S=',s:2:2,' квадрат бірлік'); ReadLn; End. </pre>
S = 6.0000000000E+00 квадрат бірлік	a,b,c -тің мәндерін енгізу 3 4 5 S = 6.00 квадрат бірлік

3-есеп. Кабырғалары R тең кабырғалы үшбұрыш, квадраттың, радиусы R -ге тең шеңбердің ауданын есептеу бағдарламасын жаса және $R = 4$ болғанда есепте.



Бағдарлама	Экрандағы нәтиже
<pre> Program Аудандарды_есептеу; var r: Integer; s1,s2,s3:Real; begin Write('R дiң мәнін енгіз :'); ReadLn(r); s1:=sqr(r)*sqrt(3)/4; s2:=sqr(r); s3:=pi*sqr(r); WriteLn('Үшбұрыштың ауданы = ',s1); WriteLn('Квадраттың ауданы = ',s2); WriteLn('Шеңбердің ауданы = ',s3); ReadLn; end. </pre>	<pre> R -дiң мәні енгiзiлсiн 4 Үшбұрыштың ауданы = = 6.9282032303E+00 Квадраттың ауданы = = 1.6000000000E+01 Шеңбер ауданы = = 5.0265482457E+01 </pre>

Осы бағдарламаны меншіктеу операторының жәрдемімен өзгертіп, нәтижесін шығаруды өз бетіңмен істеуге қалдырамыз.



Сұрақтар мен тапсырмалар

1. *Сызықтық бағдарламалар дегенде нені түсінесің?*
2. *Қандай операцияларды пайдаланып, диалог бағдарламаларын жасауға болады?*
3. *Неліктен бағдарламадағы π санына мән беру шарт емес?*
4. *Шығару форматы туралы өңгіме айт.*
5. *Экрандағы нәтижені қарап шығу үшін қайсы клавиштерді бірге басады?*

Жаттығулар

1. Төмендегі сызықтық бағдарламаның ісін түсіндіріп, нәтижесін анықта.

a) Var a,b:String; Begin a:= 'Өзбекстан'; b:= 'Мемлекет'; WriteLn(a, 'Тәуелсіз ', b); End.	ә) var a,b:Integer; s:Real; Begin a:=4; a:=sqr(a); b:=b-a; s:=2*a+3*b; WriteLn('S= ',s); End.
---	--

2. Трапецияның ауданын есептейтін мына бағдарлама тәртіпсіз жазылған. Операторларды логикалық дұрыс жүйелілікпен орналастыр.

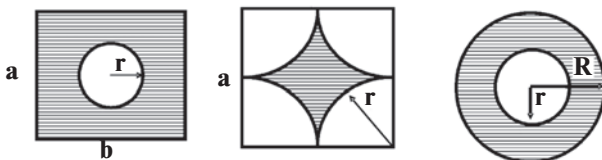
$p := (a+b) / 2$; $s := p \cdot h$; Program трапеция_ауданы; End. WriteLn ('S =', s, 'квадрат бірлік'); ReadLn(a,b,h); Begin Write('A,B,H мәндерді енгіз: '); Var a,b,h: Integer; p,s:Real;

3. Берілгені – үшбұрыштың a , b қабырғалары және олардың арасындағы α бұрыш. Үшбұрыштың ауданын есептейтін бағдарлама жаса:

$$(S = \frac{1}{2} a \cdot b \cdot \sin \alpha).$$

28-сабақ. Сызықтық бағдарламалар жасау тақырыбын қайталау

1. Төменде берілген пішіндердің штрихталған бөлімдерінің аудандарын есептейтін бағдарлама жаса (бағыт: қай пішіннің ауданынан қайсы пішін ауданын айыру керек?).



2. Берілген a және b бүтін сандар мәндерін ауыстыратын бағдарлама жаса, яғни $a = 7$ және $b = 2$ енгізілсе, $a = 2$ және $b = 7$ нәтиже шықсын (бағыт: орын ауыстыру $m := a$, $a := b$, $b := m$).

3. Кедергілері R_1, R_2, R_3, R_4 болған өткізгіштерді параллель ұлағанда шығатын R кернеуді есептеу бағдарламасын жаса (бағыт: $1/R=1/R_1+1/R_2+1/R_3+1/R_4$).

29-сабақ. Өту және тармақталу операторлары

Біз осы күнге дейін сызықтық, яғни командалары тізбектеліп орындалатын бағдарламалармен таныстық. Кейде берілген есепті шешуде операторлардың орындалу тәртібін бұзуға, яғни басқаруды бағдарлама ішінде артқа не алға көшіруге тура келеді. Бұл үшін бағдарламада басқару беріліп жатқан операторға **тамға** қойылады. Тамға, айнымалының аты сияқты, латын әріптері мен цифрлары арқылы шығарылады. Мысалы, 7, N1, тамға2. Оларда қолданылатын сан 0-ден 9999-ға дейін бола алады. Бірінші келген нөлдер есепке алынбайды.

Бағдарламада қолданылатын тамғалар бағдарламаның сипаттама бөлігінде **Label** көмекші сөзі арқылы көрсетілуге тиіс. Тамғалар бағдарламаның **өту операторы** бар болғанда ғана пайдаланылады. Өту операторының мынадай көрінісі бар: **GOTO <тамға>**; мұндағы **GOTO** (ағыл.-ге өтілсін) операторы басқаруды бағдарламаның алдында **тамғасы** бар операторына көшіреді.

<p>1-есеп: Label N1; Var a,b,c:Integer; Begin a:=15; b:=13; c:=a+b; GoTo N1; {Басқару N1 тамғалы операторға берілді} c:=a-b; N1: WriteLn(c); End.</p>	<p>Бұл бағдарлама жұмыс істеуі нәтижесінде экранда пайда болған c-тің мәні 28-ге тең. Себебі басқару N1 тамғалы шығару операторына берілгендіктен, $c:=a-b$ амалды орындамай өткізіп жіберілді.</p>
---	---

Өту операторында ешқандай шарт тексерілмейді, басқару көрсетілген тамғасы бар операторға өтеді. Бірақ көптеген есептердің шешімін шығаруда бірер шарттың орындалуына қарай не ол, не бұл амалдар тізбегін орындау керек болады. Мысалы, квадрат теңдеудің шешімін есептеу үшін, дискриминанттың белгісіне қарай үш бағыттың бірі таңдалады. Бұл түсіндірме **тармақталу алгоритмдері** деген тақырыпты есіңе түсірген болар. Мұндай есептерді шығару үшін Паскальда **тармақталу операторы** пайдаланылады.

Тармақталу операторының жалпы пішіні мынадай:

If <шарт> **Then** <оператор яки операторлар> **Else** <оператор яки операторлар>;

Бұл жердегі **if, then, else** – Паскальдің көмекші сөздері, олардың оқылуы мен мағынасы мынадай: **If** (if) – «егер», **Then** (then) – «ондай жағдайда», **Else** (else) – «ондай болмаса». Әдетте, <шарт> – рас не жалған мәндердің біреуін қабылдайтын логикалық өрнек; <оператор яки операторлар> – Паскальдің ерікті операторы я операторлар тізбегі. Есінде болсын, Else көмекші сөзінен бұрын жазылған оператордан кейін «;» (нүктелі үтір) қойылмайды. Тармақталу операторы былай істейді: әуелі **шарт** тексеріледі; егер оның мәні **рас** болса, **then** -нен кейінгі оператор я операторлар тізбегі, ондай болмаса, **else** -ден кейінгі оператор я операторлар тізбегі орындалады.

2-есеп. Енгізілген сан 25-тен үлкен болса, экранда «үлкен», ондай болмаса, «үлкен емес» деген жазуды шығаратын бағдарлама жаса.

Шешуі. Берілген санның түрі көрсетілмеген, сондықтан оның түрі нақты деп алынады.

```
Program Салыстыру;
Var a:Real;
Begin Write('Ерікті сан енгіз: '); ReadLn(a);
  If a>25 Then WriteLn('Үлкен')
    Else WriteLn('Үлкен емес');
End.
```

Тармақталу операторының бөлімдерін бөлек жолдарға да жазуға болады.

Егер THEN я ELSE сөздерінен соң орындалуға тиіс болған екі яки одан көп операторлар тізбегі жазылған болса, бұл операторлар **begin** көмекші сөзімен басталып, **end;** көмекші сөзімен аяқталуы керек.

3-есеп. Енгізілген a санның b санға көбейткенде және бөлгенде шығатын санды есептейтін бағдарлама жаса.

Шешуі. Берілген сандардың түрі көрсетілмеген, сондықтан олардың түрі нақты деп алынады.

```
Program қатынас;
Label аяқталды;
Var a, b: Real;
Begin
  Write('a санды енгіз: '); ReadLn(a);
  Write('b санды енгіз: '); ReadLn(b);
  WriteLn('Көбейтінді: ', a*b);
```

```
If b=0 Then begin WriteLn('Бөлүдү орындауға болмайды'); goto аяқталды; end;
```

```
WriteLn(' Бөлінді: ', a/b);
```

```
аяқталды: End.
```

Тармақталу операторының **Else** бөлімі керегіне қарай қолданылады. Яғни тармақталу операторын мына формада да пайдалануға болады:

If <шарт> Then <оператор яки операторлар>

Бұл тармақталу операторының **қысқа** формасы делінеді. Бұл жағдайда **шарт**тың мәні рас болса, **Then**-нен кейінгі оператор я операторлар тізбегі орындалады, ондай болмаса, басқару кезектегі (тармақталу операторынан кейінгі) операторға өтеді.

4-есеп. Берілген бүтін сан теріс болса, осы санды оның кубымен ауыстыратын бағдарлама жаса.

Шешуі.

```
var a:Integer; {берілген сан}
```

```
begin Write('Ерікті бүтін сан енгіз: '); ReadLn(a);
```

```
If a<0 Then a:= a*a*a; {сан теріс болса, оның кубымен ауыстырылады}
```

```
WriteLn(a); ReadLn; {ReadLn операторы бағдарлама нәтижесін қарап шығу үшін жазылды}
```

```
End.
```

Тармақталу операторы құрамында басқа тармақталу операторлар қолданылуы мүмкін.

5-есеп. Санның тамғасын анықтайтын бағдарлама жаса.

```
Var a:Integer; b:String; {Бір жолда бірнеше айнымалылар сипаттауға болады}
```

```
Begin
```

```
Write('Ерікті сан енгіз: '); ReadLn(a);
```

```
If a<0 Then b:= 'теріс' Else If a>0 Then b:= 'оң' Else b:= 'нөл';
```

```
WriteLn(b);
```

```
End.
```

6-есеп. Екі санның үлкенін табу (ИКТ) бағдарламасын жаса.

```
Var a,b,үлкен:Real;
```

```
Begin
```

```
Write('Бірінші санды енгіз = '); ReadLn(a);
```

```
Write('Екінші санды енгіз = '); ReadLn(b);
```

```
If a>b Then үлкен:=a Else үлкен:=b; WriteLn('үлкен сан= ', үлкен);
```

```
End.
```

Бұл бағдарламадағы $a > b$ шарты орындалса да, орындалмаса да, WriteLn('үлкен сан= ', үлкен) операторы сөзсіз орындалады. Өйткені, ол

оператор бағдарламаның тармақталу операторымен бір жолда жазылған болса да, оның құрамына енбейді. Неліктен осылай болатынын ойлаңып көрші!



Сұрақтар мен тапсырмалар

1. Тамға не үшін пайдаланылады?
2. Өту операторының жалпы көрінісі қандай?
3. Өту операторы бар бағдарламада тамғаларды пайдаланбауға бола ма?
4. Тармақталу операторы не үшін пайдаланылады?
5. Тармақталу операторына операторлар тізбегі қатысқан болса, олар қандай көмекші сөздер арасында жазылады?
6. Тармақталу операторының қысқа әрі толық көріністері туралы не білесің?
7. Қайсы оператордан кейін нүктелі үтір жазылмайды?

Жаттығулар

1. Төмендегі өту операторлары арасында қате жазылғанын анықта.

- | | |
|--------------|--------------|
| а) Goto 10; | ә) goto 30; |
| б) goto -5; | в) GoTo _5; |
| г) goto sin; | д) goto 2_5; |
| е) GOTO a_5; | |

2. Тармақталу операторына тән төмендегі салыстыру шарттарының ішінен қате жазылғанын тап.

- | | |
|----------------|---------------|
| а) $a < > b$; | ә) $a < -b$; |
| б) $a > < b$; | в) $-a > 0$; |
| г) $-1 > 0$; | д) $a >> b$; |
| е) $a := b$; | |

3. Төмендегілердің арасынан қате жазылған өрнекті тап.

- | |
|-----------------------------------|
| а) IF a=b THEN a:=a+1; ELSE b:=a; |
| ә) IF a:=1 THEN a:=a+1 ELSE b:=a; |

4. x -тің берілген мәнінде $y = \begin{cases} -1, & \text{егер } x > 0, \\ x^2, & \text{егер } x \leq 0 \end{cases}$ функцияның мәнін

есептейтін бағдарлама жаса.

5. Үш сан берілген. Солардың ішіндегі теріс сандардың кубын есептейтін бағдарлама жаса.

6. Берілген бүтін сан теріс болса, оның модулін есептейтін бағдарлама жасалсын. Есептің шешімін 2 әдіспен шығаратын бағдарламаны $abs(x)$ функциясын пайдаланып және пайдаланбай жаса.

7. Егер пароль «информатика» болса, онда парольдің дұрыс енгізілгенін тексеретін бағдарлама жаса.

30-сбақ. Өту және тармақталу операторлар тақырыбын қайталау

- Берілген мәнге сүйеніп, шарттың мәнін анықта.
 - $a:=10; b:=a*3$; шарт: « $a < b/3$ »;
 - $a:=10; b:=a*3$; шарт: « $a <= b/3$ »;
 - $a:=10; b:=a$; шарт « $a+b=2*b$ »;
 - $a:=10; b:=a+3$; шарт: « $a+3 >= b-3$ »;
- Тармақталудың нәтижесінде шығатын мәндерді анықта.
 - $aa:=7; bb:=6.6$; if $aa=\text{round}(bb)$ then $mm:='Иә'$ else $mm:='Жоқ'$;
 - $ag:=\text{true}$; if ag then $aa:=21$ else $aa:=7$; $a:=a+1963$;
 - $ag:=\text{true}$; if ag then $aa:=21$ else begin $aa:=7$; $a:=a+1963$; end;
 - $ms:=50; aa:=10$; if $ms \text{ div } aa = aa*5$ then $ms:=\text{trunc}(ms/3)$ else $aa:=ms \text{ mod } aa$;

3. Екі a мен b сандар берілген. Егер b сан a -дан кіші болса, онда b -ның орнына нөл салатын, ондай болмаса, b -ны өзгеріссіз қалдыратын бағдарлама жаса.

4. Үш a, b, c сандар берілген. Осы сандар арасынан теріс сандардың ғана квадрат түбірін есептеп шығаратын бағдарлама жаса.

5. $ax + b = 0$ тендеуінің түбірін a мен b -ның төмендегі мәндерінде есептейтін бағдарлама жаса.

- | | |
|----------------------|----------------------|
| a) $a = -1, b = 1$; | ә) $a = 0, b = 4$; |
| б) $a = 1, b = 0$; | в) $a = 1, b = -5$. |

6. Берілген A бүтін сан берілген нөлден ерекшеленетін B бүтін санға қалдықсыз бөлінетінін я бөлінбейтінін анықтайтын бағдарлама жаса.

7. Үш a, b, c сандар берілген. Егер $a^2 + b^2 = c^2$ шарты орындалса, осы сандардың қосындысын, ондай болмаса, олардың модульдерінің көбейтіндісін есептейтін бағдарлама жаса.

31-сбақ. Тармақталатын құрылымдық бағдарламалар жасау

Осы күнге дейін біз қарапайым шарттарға сүйеніп, бағдарлама жасауды қарастырдық. Бірақ тармақталатын операторлар күрделі шарттарды да пайдалануға мүмкіндік береді. Күрделі шарттар **NOT** – «емес», **AND** – «және», **OR** – «немесе» логикалық амалдарын қарапайым шарттарға қатысты қолданудың нәтижесінде пайда болады.

Есінде ме, **NOT** – логикалық жою, **AND** – логикалық көбейту, **OR** – логикалық қосу амалдары деп аталады. Осындай амалдармен біз 8-сыныпта танысқанбыз. **NOT** өзінен кейін орналас-

кан шарттың қабылданбаған мәнін береді. **AND** екі жанында орналасқан шарттардың екеуі рас болғанда ғана рас мәнін береді. **OR** екі жанында орналасқан шарттардың ең болмағанда біреуі рас болғанда ғана рас мәнін береді.

Логикалық өрнектерде бірінші кезекте **NOT** амалы, екінші кезекте **AND**, үшінші кезекте **OR** амалы орындалады. Егер логикалық өрнектерге жақшалар қатысса, алдымен олардың ішіндегі өрнек орындалады. Ұқсас амалдар тізбектеліп келгенде, амалдар солдан оңға қарай орындалады. Логикалық амалдар қолданылғанда, шарттар жақшаның ішіне жазылады. Мысалы:

1) $x \in [a, b]$ (яғни $a \leq x \leq b$) Паскальда $(A \leq X) \text{ AND } (X \leq B)$ түрінде жазылады;

2) $\overline{t_1 = t_2}$ Паскальда **NOT**($T1=T2$) түрінде жазылады;

3) $y < -5$ немесе $y > 2$ Паскальда $(Y < -5) \text{ OR } (Y > 2)$ түрінде жазылады.

1-есеп. Төмендегі функцияның мәнін x -тің берілген мәнінде есептейтін бағдарлама жаса.

$$y = \begin{cases} x^2, & \text{егер } x \in (0, 1] \text{ болса,} \\ x, & \text{егер } x \in (0, 1] \text{ болса.} \end{cases}$$

Шешуі.

Var x, y: real;

Begin Write('x= '); ReadLn(x);

If (0<x) And (x<=1) Then y:=Sqr(x) Else y:=x;

WriteLn('y=',y): ReadLn;

End.

Бұл жерде $0 < x$ және $x \leq 1$ шарттардың екеуі де орындалса, яғни x -тің мәні $(0, 1]$ аралықта жататын болса, $y := \text{Sqr}(x)$ меншіктеу операторы, ондай болмаса, яғни x -тің мәні $(0, 1]$ аралықта жатпаса, $y := x$ меншіктеу операторы орындалады.

Тармақталуға қатысы бар төмендегі есептер назар салуға лайықты:

1) If $(A > B) \text{ And } (B > C) \text{ Then } S := B + 7 \text{ ELSE } S := A * B - 1$;

Егер $A > B$ және $B > C$, яғни $A > B > C$ болса, онда $S := B + 7$ операторы орындалады, ондай болмаса, $S := A * B - 1$ операторы орындалады.

2) If $5 * B = M * M \text{ Then Goto } 200 \text{ ELSE Goto } 400$;

Егер $5 * B = M * M$ болса, онда басқару тамғасы «200» операторға, ондай болмаса, «тамғасы 400» операторға өтеді.

3) If $R1 \leq R2 \text{ Then begin WriteLn}(S); R := R1 + R2 \text{ end}$

Else begin WriteLn($S * R1$); $R1 := R2$; $R2 := 0$; end;

Егер $R1 \leq R2$ болса, онда `WriteLn(S)` және $R := R1 + R2$ операторлар орындалады, ондай болмаса, `WriteLn(S*R1)`; $R1 := R2$; va $R2 := 0$; операторлар орындалады.

4) `If SR = 'YASHIL' Then WriteLn('Өту мүмкін) Else WriteLn('Өту мүмкін емес);`

Егер SR (бағдарламаның түсініні) мәні «ЖАСЫЛ» болса, `WriteLn('Өтуге болады')` операторы, ондай болмаса, `WriteLn('Өтуге болмайды')` операторы орындалады.

Көрініп тұрғанындай, егер берілген шарт дұрыс болса, онда THEN көмекші сөзінен кейінгі командалар орындалады, ондай болмаса, ELSE көмекші сөзінен кейінгі командалар орындалады. Бұл жерде үшінші жағдайдың жоқтығын жақсы түсіну керек. Енді тармақталатын бағдарламаларға есептер келтіреміз:

<p>2-есеп. Берілген үш a, b, c сандардың үлкенін табу (UKT) бағдарламасын жаса.</p>	<p>3-есеп. Берілген натуралды санның тақ немесе жұп екенін анықтайтын бағдарлама жаса.</p>
<pre> Program UKT; Var a,b,c,max : Real; Begin Write('a,b,c сандардың мәнін енгіз: '); ReadLn(a,b,c); If a>b Then max:=a Else max:=b; If c>max Then max:=c; WriteLn('Берілген үш санның үлкені= ',max); End. </pre>	<pre> Program Так_жұп; Var n : word; Begin Write('Натуралды сан енгіз:'); ReadLn(n); If Odd(n) Then WriteLn('ТАҚ') Else WriteLn('ЖҰП '); End. </pre>

4-есеп. $ax^2 + bx + c = 0$ квадрат теңдеудің шешімін шығаратын бағдарлама жаса.

```

Program Квадрат_тендеу;
Label Аяқталды;
Var a,b,c,d,x1,x2 : Real;
Begin
Write('a,b,c сандарының мәнін енгіз: '); ReadLn(a,b,c);
d:=Sqr(b) - 4*a*c; {Дискриминант саналады}
If d<0 Then begin WriteLn('Нақты шешімі жоқ'); Goto аяқталды;
end;
If d=0 Then begin
        
```

```

WriteLn('шешімі біреу: ');
WriteLn('x= ', -b/(2*a)); Goto Аяқталды; end;
WriteLn('Шешуі екеу: ');
x1:=(-b-Sqrt(d))/(2*a); x2:=(-b+Sqrt(d))/(2*a);
WriteLn('x1= ',x1); WriteLn('x2= ',x2);
Аяқталды: ReadLn;
End.

```

Жоғарыда келтірілген есептердің шешімінен көрініп тұрғанындай, тармақталатын бағдарламаларда тармақталуды ұйымдастыру қойылған есептің маңызына байланысты болады екен.



Сұрақтар мен тапсырмалар

1. Тармақталу операторының қысқа және толық формалары бір-бірінен несімен ерекшеленеді?
2. Паскаль тілінде қандай логикалық амалдар қолданылады?
3. Паскаль тіліндегі күрделі логикалық шарттарға мысалдар келтір.
4. Логикалық өрнектегі амалдардың орындалу тәртібін түсіндір.
5. Логикалық өрнектегі жақшалар қашан пайдаланылады?

Жаттығулар

1. Төмендегі операторларда жіберілген қателерді анықтап, түсіндір.
 - a) IF d>0 THEN 63 ELSE s:=d+a;
 - ә) IF s1<>s2 THEN ELSE g1:=s1*s2;
 - б) IF i*j THEN goto vo ELSE goto ne;
 - в) IF x<>0 AND x<=5 THEN y=4*sin(x);
2. Төмендегі логикалық өрнектердегі амалдардың орындалу тәртібін анықта.
 - a) a<-6 OR a>=0 AND a<4 ;
 - ә) x*x +y >0 AND a=0.1 OR (b>3.7 AND s<>k4) ;
 - б) v= 'ha' AND x1>0 AND x2>0 ;
 - в) a>0 OR a<1 OR NOT x*x+x*x<=1 ;
 - г) NOT v<=b AND (f<=f1 OR t='.')
 - д) NOT(NOT(NOT(a>b) OR TRUE) AND FALSE) ;
3. Ұзындықтары арқылы берілген үш кесіндіден үшбұрыш шығарып алуға болатынын я болмайтынын анықтайтын бағдарлама жаса.
4. Төменгі сынып оқушысын көбейту кестесі бойынша сынайтын бағдарлама жаса. Онда оның жауабы дұрыс болса, «Жарайсың» деген, дұрыс болмаса, «Қайта істе» деген мәтіндер түрлі түсті болып шықсын.
5. Енгізілген 1-ден 7-ге дейінгі аралықта жататын цифрға сүйеніп, апта күнін экранға шығаратын бағдарлама жаса.

32-сабақ. Тармақталатын құрылымдық бағдарламалар жасау тақырыбын қайталау

1. Күрделі логикалық амалдардың нәтижесін анықта.
 - a:=true; b:=true; m:=false; bb:=NOT(a AND m) AND (a OR b) OR m;
 - ә) a:=77; b:=11; m:=7; ms:= (a div b=m) AND (a mod m=0) AND NOT((a>b) OR (b<m));
2. Тармақталудың нәтижесінде шығатын мәндерді анықта.
 - x:= -1; y:=0; a:= 0.1; IF (x*x +y >0) AND (a=1/10) THEN mm:=true else mm:= false;
 - ә) x1:=sqr(-1); v:= 'и ә'; x2:=sqr(x1); IF (v= 'иә') AND (x1>0) AND (x2>0) THEN x1:=0;
3. Үш a, b, c сан берілген. $a < b < c$ теңсіздіктің орындалатынын яки орындалмайтынын тексеретін бағдарлама жаса.
4. Берілген A бүтін сан B бүтін санға қалдықсыз бөлінсе, осы екі сан қосындысының квадратын, ондай болмаса, көбейтіндісін шығаратын бағдарлама жаса.
5. Берілген бүтін N сан оң әрі 5-ке еселі болса, осы санның квадрат түбірін, ондай болмаса, квадратын есептейтін бағдарлама жаса.
6. M мен N сандар берілген. Егер олар оң және қосындысы 100-ден үлкен болса, M санын N санға бөлгенде, ал егер олар оң және қосындысы 100-ден үлкен болмаса, M -ның N -ге көбейткенде шығатын санды есептейтін бағдарлама жаса.
7. Берілген N санның бүтін бөлігі, бөлшек бөлігінің 1000-ға көбейткендегі саннан үлкен болса, осы санның бүтін бөлігінің, ондай болмаса, бөлшек бөлігінің бірінші 3 цифрын шығаратын бағдарлама жаса.

33-сабақ. Параметрлік қайталау операторы

Көптеген есептерді шешуде белгілі амалдар тізбегін бірнеше рет қайталау керек болады. Өткен тарауда сен осы сияқты есептермен танысып, олардың шешімін табу үшін қайталанатын алгоритмдер жасадың. Енді қайталанатын бағдарламалар жасау әдістерін қарастырамыз.

Қайталанатын бағдарламалар жасау үшін **қайталау операторы** пайдаланылады. Паскальда олар үшеу, тап осы сабақта біз олардың біреуі – **параметрлік қайталау операторын** үйренеміз. Бұл оператордың жалпы көрінісі мынадай:

For I: = N1 To N2 Do <қайталау денесі>;

мұндағы **For** (үшін), **To** (-ге дейін), **Do** (орында) – Паскальдің көмекші сөздері; **I** – бүгін түрлі ерікті айнымалы, оны **қайталау параметрі** деп атайды; **N1** – қайталау параметрінің қабылдайтын бастапқы мәні; **N2** – қайталау операторының қабылдайтын соңғы мәні; **<қайталау денесі>** – қайталануға тиісті оператор я операторлар тізбегі. Қайталану денесін операторлар тізбегі құрастырған болса, олар сөзсіз **begin** командасымен басталып, **end**; командасымен аяқталады. Қайталау параметрінің бастапқы және соңғы мәндері тұрақты, айнымалы яки өрнек көрінісінде болуы мүмкін.

Бұл оператор былайша істейді:

1. Алдымен қайталау параметрі бастапқы мәнін қабылдайды;
2. Егер қайталау параметрінің мәні соңғы мәнінен үлкен болмаса, қайталау денесін құрастырушы операторлар орындалады, ондай болмаса, қайталау тоқтатылады да басқару кезектегі операторға өтеді;
3. Қайталау параметрінің мәні бірге артады (оған 1 қосылады) және 2-пунктке өтіледі.

For операторы, негізінен, қайталанымдар саны бұрыннан белгілі болғанда ғана қолданылады.

1-есеп. «Өзбекстан – Отаным менің!» мәтінін экранға 20 рет шығаратын бағдарлама жаса.

Шешуі. Есептің шартына сәйкес, «Өзбекстан – Отаным менің!» мәтінін 20 рет шығару керек. Яғни, WriteLn('Өзбекстан – Отаным менің!') операторын 20 рет қайталау қажет. Төмендегі бағдарламаны жасаймыз.

```
Program Қайталау;
```

```
Var I : Integer;
```

```
Begin
```

```
  For I:=1 To 20 Do WriteLn('Өзбекстан – Отаным менің!');
```

```
End.
```

Тап осы бағдарламадағы I қайталау параметрінің бастапқы мәні 1-ге, соңғы мәні 20-ға тең болады. Қайталау денесі біреу – WriteLn('Өзбекстан – Отаным менің!') операторы. Бағдарлама орындалғанда, қайталау параметрі кезекпен 1, 2, 3,..., 20 мәндерін қабылдайды да, әр кез WriteLn('Өзбекстан – Отаным менің!'); операторы орындалады. Нәтижесінде экранға «Өзбекстан – Отаным менің!» мәтіні 20 рет жаңа жолдан шығарылады. Бағдарламаның қайталау параметрінің бастапқы мәнін 41-ге, ал соңғы мәнін 60-қа өзгертсе де, нәтиже солай болады, өйткені қайталанымдар саны – $60-41+1=20$ рет.

2-есеп. Экранға 1-ден 20-ға дейінгі бүгін сандарды өсу тәртібімен шығаратын бағдарлама жаса.

Шешуі. Экранға шығарылатын сандар S деп белгіленеді. Әуелі $S:=0$ деп алынады. S -тің мәні $S:=S+1$ көмегімен бірге арттырылып, WriteLn(S) арқылы экранға шығарылады. Бұл амалдарды 20 рет қайталау керек, сондықтан For операторы пайдаланылады.

```
Program Тізбек_телген;
Var I, S : Integer;
Begin
    S:=0;
    For I:=1 To 20 Do begin S:=S+1; WriteLn(S); end;
End.
```

Жалпы алғанда, қайталау параметрін қайталау денесінде де істетуге болады. Бірақ оның мәнін өзгертуге болмайды. 2-есептің бағдарламасын талдай отырып, S айнымалының қайталану барысында I -мен бірдей мәндер қабылдап алғанын көруге болады. Демек, экранға S -тің орнына I -ді шығаруға да болады екен. Онда бағдарламада S айнымалыны қолданбауға да болады. Бұларды ескеріп, бағдарламаға мынадай өзгерістер енгізіледі:

```
Program Тізбек_телген;
Var I : Integer;
Begin
    For I:=1 To 20 Do WriteLn(I);
End.
```

3-есеп. 1 -ден 100 -ге дейінгі бүтін сандарды азаю тәртібімен шығаратын бағдарлама жаса.

Шешуі. Басып шығарылатын сандар санды, қайталау параметрі i деп белгіленеді.

```
Program Сандар;
Var i, сан : Integer;
Begin
    сан:=101;
    For i:=1 to 100 Do Begin сан:=сан-1; WriteLn(сан); end;
End.
```

For операторының қайталау параметрі үлкен мәнінен кіші мәніне қарай азайып баруы мүмкін. Бұл үшін **To** көмекші сөзінің орнына **Downto** көмекші сөзі қолданылады. Осыны ескеріп, жоғарыдағы бағдарламаны былайша ықшамдауға болады:

```
Program Сандар;
    Var i : Integer;
Begin
    For i:=100 Downto 1 Do WriteLn(i);
End.
```

4-есеп. 1-ден 21 -ге дейінгі тақ сандар қосындысын есептейтін бағдарлама жаса.

Шешуі. 1-ден 21 -ге дейінгі сандарды қарап шығу үшін қайталау операторының параметрін қолдануға болады. $S=1+2+3+\dots+21$ қосындысындағы қосылғыштар 255-тен үлкен емес, демек, онда параметр үшін **byte** түрін, S үшін – бүтін әрі теріс емес болғандықтан, **word** түрін тандап алуға болады. Индекстердің тақ екенін тексеру үшін $Odd(x)$ функциясын пайдалану қолайлы.

```
Program Summ;
Var s: word; i: byte;
Begin
  S:=0;
  For i:=1 to 21 Do If odd(i) then S:=S+i;
  WriteLn('S= ', S);
End.
```

5-есеп. Берілген $A[1..21]$ массивінің жұп индексі бар элементтерінің қосындысын есептейтін бағдарлама жаса.

Шешуі. Массивті енгізіп, индекстерді қарап шығу үшін, сондай-ақ қайталау операторының параметрі мен индекстер жұбын тексеру үшін $NOT(Odd(x))$ пайдаланылады.

```
Program Массив;
Var i: Integer; s: real;
a: array[1..21] of real;
Begin
  For i:=1 to 21 Do Begin WriteLn('a[', i, ']= '); ReadLn(a[i]); end;
  S:=0;
  For i:=1 to 21 Do If NOT(odd(i)) then S:=S+a[i];
  WriteLn('S= ', S); ReadLn;
End.
```

Көрініп тұрғанындай, массивтерді енгізу және шығару, олардың индекстерін санау сияқты міндеттерді орындау үшін параметрлік қайталау операторы өте қолайлы екен.



Сұрақтар мен тапсырмалар

1. Қайталанатын алгоритмдерге мысалдар келтір.
2. Параметрлік қайталау операторының көрінісі қандай?
3. Қайталау параметрі қандай мәндерді қабылдайды?
4. Параметрлік қайталау операторының бастапқы және соңғы мәндерін түсіндіріп бер.
5. Қайталау параметрінің мәндері шектелген бе?
6. Қайталау операторының жұмысын түсіндір.
7. Қандай жағдайда *To орнына Downto* көмекші сөзі қолданылады?

Жагыгулар

1. Төмендегі операторлардағы қайталанымдар санын анықта.
 - a) for i:=1 to 88 do b:=1;
 - ә) for i:=73 to 161 do m:=2;
 - б) for i:= -21 to 0 do a:=3;
 - в) a:=5; b:=34; for i:=a+7 to b-1 do s:=s+1;
 - г) a:=5; b:=19; for i:=a*a to 2*b+8 do s:=s+1;
2. $y = 21x^2 + 7x + 1963$ функциясының мәнін x -тің $-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$ мәндерінде есептейтін бағдарлама жаса.
3. $y = 23x + 1$ функцияның мәнін x -тің $[-15, 5]$ аралықтағы бүтін мәндерінде есептейтін бағдарлама жаса.
4. A[1..17] массив берілген. Массивтің нөлге тең элементтерінің индексін шығаратын бағдарлама жаса.
5. $y = 2x + 19$ функцияның мәнін x -тің $[0, 10]$ аралықтағы мәндерінде $0,25$ қадамдап есептейтін бағдарлама жаса (бағыт: $i=0$ да $x=0$; $i=1$ да $x=0,25$; ...; $i=40$ да $x=10$).

34-сабақ. Параметрлік қайталау операторы тақырыбын қайталау

1. Төмендегі операторларда жіберілген қателерді анықтап, түсіндір.
 - a) for I= -15 to 5 do s:=s+I;
 - ә) for kub:=100/10+11 to 1963 do begin a:=7;end;
 - б) for mag:=99 dawnto 1 do readl(aa);
 - в) for bma:= 0.5 to 10 do writeln(k);
2. Төмендегі операторлардағы қайталанымдар санын анықта.
 - a) for k:=trunc(23/5) downto trunc(1/2) do m:=1991;
 - ә) for s:=23 to 1 do m:=1963;
 - б) for J:=2 downto 19 do m:=1950;
 - в) for d:=23 downto 1 do m:=2009;
 - г) for i:=abs(-25) to 25 do s:=s+i*i;
 - д) for h:=round(9.6) downto trunc(3*3) do a:=21;
3. $S=10+12+14+\dots+50$ қосындысын есептейтін бағдарлама жаса.
4. $S = \frac{7}{11} + \frac{17}{21} + \frac{27}{31} + \dots + \frac{2007}{2011}$ қосындысын есептейтін бағдарлама жаса (бағыт: J-ді 10-ға бөлгендегі қалдық 7).
5. $P=1*3*5*...*23$ көбейтіндісін есептейтін бағдарлама жаса.
- 6*. A[1..5] массив берілген. Массив элементтерін теріс тәртіппен шығаратын бағдарлама жаса.

7*. A[1..15] массив берілген. Массивтің тақ индексі бар элементтерінің қосындысынан жұп индексі бар элементтерінің қосындысын айырып, нәтижені экранға шығаратын бағдарлама жаса.

35-сабақ. Шартты қайталау операторлары

Алдыңғы сабақта қарастырылған есептерімізде қайталанымдар саны мәлім еді. Бірақ сондай есептер де кездеседі, олардың белгілі бір амалдар тізбегін мәлім бір шарт бойынша қайталау керек болады, ал сондағы қайталанымдар санын алдын ала айтуға болмайды. Осындай жағдайда берілген шарт бойынша шартты қайталанатын операторлар пайдаланылады. Паскальда мұндай операторлар екеу: **While** мен **Repeat**.

While операторының жалпы көрінісі мынадай:

While <шарт> Do <қайталану денесі>;

мұндағы **While** (ағыл. дейін), **Do** – Паскальдің көмекші сөздері; **<шарт>** – қарапайым я күрделі логикалық өрнек; **<қайталану денесі>** – қайталану денесін құрастырушы оператор я операторлар тізбегі. Егер қайталану денесінде операторлар тізбегі жазылса, олар **begin** -мен басталып, **end;** -пен аяқталады.

Бұл қайталанатын оператор былай істейді:

– алдымен **шарт** тексеріледі. Егер оның мәні **рас** болса, қайталау денесін құрастырушы операторлар істейді де **шарт** тағы тексеріледі. Бұл процесс шарттың мәні **жалған** болғанша жалғасады.

1-есеп. Екі санның ең үлкен ортақ бөлгішін табатын бағдарлама (ЕҮОБ) жаса.

Шешуі. ЕҮОБ табудың ең қолайлы әдісі – Эвклид алгоритмі. Бұл алгоритм саған бұрынғы тараудан таныс. Оның бағдарламасы мынадай:

Program EҮОБ;

Var a, b : Integer;

Begin

Write('Бірінші санды енгіз: '); ReadLn(a);

Write('Екінші санды енгіз: '); ReadLn(b);

While a<>b Do If a>b Then a:=a-b Else b:=b-a; {a=b болғанда

қайталау аяқталады}

WriteLn('ЕҮОБ= ', a);

End.

Repeat операторында шарт бойынша шартты қайталауды жүзеге асырады. Оның жалпы көрінісі мынадай:

Repeat

<қайталану денесі>

Until <шарт>

мұндағы **Repeat** (ағыл. қайталау), **Until** (ағыл. -ке дейін) – Паскаль тілінің көмекші сөздері, **Repeat** – қайталау басы, **Until** – қайталау аяғы; <шарт> – қарапайым я күрделі логикалық өрнек. Қайталау денесі **шарттың** мәні **рас** болғанша орындалады.

2-есеп. $S = 1, 1+1, 5+1, 9+2, 3+...+45, 5$ қосындысын есептейтін бағдарлама жаса.

Шешуі. Есепті талдаған соң, қосылғыштардың арасында 0,4 -ке тең айырма бар екені анықталады. Параметрлік қайталанатын оператор параметрінің мәні нақты бола алмайды. Ондай жағдайда шартты қайталау операторларын пайдалану қолайлы. Есептің төмендегі екі шешімін салыстырып көр.

While көмегімен	Repeat көмегімен
<pre> Program Summ_real; Var J,S: real; Begin S:=0; J:=1.1; While J<= 45.5 do begin S:=S+J; J:=J+0.4; end; WriteLn('S= ',S); End.</pre>	<pre> Program Summ_real; Var J,S: real; Begin S:=0; J:=1.1; Repeat S:=S+J; J:=J+0.4; Until J>=45.5; WriteLn('S= ',S); End.</pre>

3-есеп. Кездейсоқ сандар генераторын (Random) пайдаланып, экранға «А» әрпін түрлі түсті етіп шығаратын бағдарлама жаса. Қызыл түсті «А» белгісі шығысымен бағдарламаның жұмысы аяқталсын.

Шешуі. Паскальдағы түстер 0-ден 15-ке дейінгі бүтін сандармен кодталатыны белгілі. Кездейсоқ сандар генераторы – Random(x) функциясы [0, x) аралықтан кездейсоқ сандарды береді. Сондықтан [0, 15] аралықтағы бүтін сандарды кездейсоқ шығару үшін Random(16) функциясы қолданылады.

Ескерте кету керек, **Random(x)** функциясы, бағдарламаны әр кез іске қосқанда, бірдей сандар тізбегін тудырады. Түрлі сандар шығару үшін **Randomize** операторы пайдаланылады.

Бұл оператор бағдарламада Random функциясынан алдын жазылуға тиіс. Қызыл түстің коды 4 екенін ескеріп, мына бағдарламаны жасаймыз:

```

Program Түрлі түсті_әріптер;
Uses Crt;
Var түс : Integer;
Begin
  Randomize;
  Repeat
    түс:= Random(15); TextColor(түс); Write('A');
  Until түс=4; {түс=4 (қызыл) болса,
```

қайталау операторынан шығады}

End.

Repeat операторының While операторынан бірінші айырмашылығы сол, While операторындағы шарт қайталау басында тексеріледі, ал Repeat операторында – қайталау соңында тексеріледі. Сондықтан While операторындағы қайталау денесін құрастырушы операторлар (шарт алдын ала орындалмаса) бір рет те орындалмауы мүмкін, ал Repeat операторында ең болмағанда бір рет орындалады. Екінші айырмашылығы, While операторындағы қайталаным шарт орындалмағанда («жалған» мәнін қабылдағанда) аяқталса, Repeat операторында қайталау шарты орындалғанда («рас» мәнін қабылдағанда) аяқталады.

Жоғарыда келтірілген 1-есепте алдымен шарт тексерілуге тиіс. Сондықтан онда While операторы пайдаланылады. Ал 3-есепте алдымен түстің мәні анықталып, кейін шарт тексерілуі керек. Сондықтан онда Repeat операторы қолданылады. Жалпы алғанда, 1-есепте Repeat операторын, ал 3-есепте While операторын пайдаланса да болады. Тек бұл жағдайда бағдарламаға кейбір қосымша операторлар енгізуге тура келеді. Бірақ бағдарламаның «сұлу» да «түсінікті» болуы үшін, While және Repeat операторларын өз орнында қолдану дұрыс болады.



Сұрақтар мен тапсырмалар

1. Шартты қайталанатын қандай операторларды білесің?
2. While операторының жұмысын түсіндір.
3. Repeat операторының жұмысын түсіндір.
4. Шартты қайталанатын оператордың параметрлік қайталау операторынан айырмашылығы неде?
5. Қайталау операторларын пайдалануға қолайлы жағдайларға сәйкес етіп түсіндір.

Жаттығулар

1. Төмендегі операторларда жіберілген қателерді анықтап, түсіндір.
 - а) while 5*6 do SH:=sqr(2);
 - ә) WHILE 5>6 do Od:=Od+1;
 - б) Repeat i<j Until s:=0;
 - в) rErEaT s:=0 UntiL s:=0;
2. Төмендегі операторлардағы қайталанымдар санын анықта.
 - а) x:= - 5; while X>0 do x:=x+2;
 - ә) x:= - 5; while X<10 do begin x:=x+2;x:=2*x; end;
 - б) i:=0; while i*i <=1.2 do i:=i+0.1;
 - в) k:=5; while k /5 <= 2.5 do k:=k+1.5;
 - г) t:=100; repeat t:= t/10; until t<=0.1;
 - д) x:=0; repeat x:=x+1/10; until sqr(x)>=6/5;

3. N натуралды сан берілген. Квадраты N -нен үлкен емес барша натуралды сандарды шығаратын бағдарлама жаса.

4. $y = x \cdot \sin x$ функциясының мәнін $[-\pi, \pi]$ аралықта 0,3 қадамдап есептейтін бағдарлама жаса.

5*. Берілгені – N натуралды сан және A_1, A_2, \dots, A_N бүтін сандар тізбегі. Оларды тізбектеп қосып отырып, қосындысы берілген N бүтін саннан асқан сайын оны экранға шығаратын бағдарлама жаса. Егер барлық сандар қосындысы N -нен аспайтын болса, бұл туралы хабар шығарылсын.

36-сабақ. Шартты қайталанатын операторлар тақырыбын қайталау

1. $S=0,5+1,5+2,5+\dots+98,5+99,5$ қосындысын есептейтін бағдарлама жаса.

2. $S=1*2+3*4+5*6+\dots+101*102$ қосындысын есептейтін бағдарлама жаса.

3. N натуралды санның барлық бөлгіштерін шығаратын бағдарлама жаса.

4. Берілген N натуралды санның неше цифрдан тұратынын анықтайтын бағдарлама жаса (бағыт: неше рет $N=N \text{ div } 10$ орындалса, $N=0$ болады?).

5. N натуралды сан берілген. 1-ден N -ге дейінгі натуралды сандар ішіндегі соңғы цифры 3-ке еселі сандарды шығаратын бағдарлама жаса.

6. Екі разрядты натуралды сандар ішінен цифрлар қосындысы жұп сан болған сандарды шығаратын бағдарлама жаса (бағыт: K санның бірлік цифры $K1=K \bmod 10$, ал ондық цифры $K10=K \text{ div } 10$).

7. A_1, A_2, \dots, A_N бүтін сандар тізбегі берілген. Сол тізбектің так элементтері көбейтіндісінен жұп элементтері қосындысын айыратын бағдарлама жаса.

8*. Берілгені – N натуралды сан және $A[1..M]$ массив. Сондай $B[1..N]$ массивін алу керек, оның әрбір K -элементі $B[K] = (A[1] + A[2] + \dots + A[K])/K$ формуласымен анықталсын.

9*. 1-ден үлкен A сан берілген. $7^K > A$ шартына сай келетін, ең кіші де теріс емес бүтін K санын табу бағдарламасын жаса.

37-сабақ. Қайталауға арналған тапсырмалар

1. Төмендегі қосындыны оның мәні берілген M натуралды саннан көп болғанша есептейтін бағдарлама жаса.

$$y = \frac{1}{3} - \frac{1}{10} + \frac{1}{21} - \dots + \frac{(-1)^{j+1}}{j \cdot (2 \cdot j + 1)} - \dots$$

2. 1-жаттығудың шешімін параметрлік қайталау операторының және шартты қайталанатын операторлар көмегімен түрліше шығар.

3. N натуралды сан берілген. $(1/2)$, $(3/4)$, $(5/6)$, $(7/8)$, ... тізбегінің N нақты қосындысын табатын бағдарлама жаса.

4. A оң сан берілген. Егер k -квадраттың қабырғасы $\frac{A}{k}$ болса ($k=1,2,\dots$), k -нің қандай мәнінде барлық квадраттар аудандарының қосындысы бірінші рет A^2 -тан үлкен болатынын анықтайтын бағдарлама жаса.

5. P -ге көбейтілген және оған тең емес үш разрядты санды анықтайтын бағдарлама жаса.

6. Берілгені — N натуралды сан және $A[1..N]$ массив. Массивтің ең үлкен және ең кіші элементін табатын бағдарлама жаса.

7. Берілген M натуралды санның цифрлар қосындысын табатын бағдарлама жаса.

8*. B , M , A натуралды сандар берілген. Тізбек $Y_1=B; Y_i = \sqrt{M} + A \cdot Y_{i-1}, i = 2,3,\dots$ заңдылық бойынша шығарылды. Тізбектің $B \cdot M \cdot A$ санынан кіші болған барлық қадамын басып шығаратын бағдарлама жаса.

9*. Клиент банкіге B сум ақша қойды. Банктегі ақшаға жылына M пайыз үстеме қосылады. Неше жылдан кейін клиенттің ақшасы A сумнан асатынын анықтайтын бағдарлама жаса.

10*. Шағын кәсіпорын 1-күні B дана тауар өндірді. Кейінгі әрбір күнде алдыңғы күнге қарағанда M дана артық тауар өндірді. Өндірілген барлық тауардың саны жоспарланған A данадан бірінші рет неше күннен кейін артқанын анықтайтын бағдарлама жаса.

11*. Саны B -ға тең, берілген реттік нөмірі бар адамдар санап отырған адамды шеңбер тәріздес қоршап тұр. Санаушы адам M -ге дейін санағанша, M - саналған адам шеңберден шығады да, санаушы адам кейінгі адамды 1-ден бастап санайды. Санау 1 адам қалғанша жалғаса береді. Ақырында нешінші реттік нөмірі бар адам қалғанын анықтайтын бағдарлама жаса.

38-сабақ. Символдық және жолдық шамалармен жұмыс істеу

Паскальдағы символдық және жолдық шамалармен істеу үшін арнайы функциялар мен процедуралар енгізілген. Төмендегі кестеде солардың кейбіреулері келтірілген.

Ескерте кету керек, бағдарламаға функциялар қатысқанда олардың мәні бірер айнымалыға меншіктенеді, ал процедуралар қолданылғанда меншіктеу операторынсыз жазылады.

Енді қарапайым есептерді қарастырамыз.

Жазылуы	Міндегі
Стандарт функциялар	
Concat(S1,S2, ..., SN)	S1, S2,...,SN жолдық (символдық) айнымалы (тұрақтыларды) тізбектеп ұлайды
Length(S)	S жолдың ұзындығын (белгілері санын) анықтайды.
Pos(b,S)	S жол ішінен b сөзді іздейді
Copy(S,n1,n2)	S жолдың n1 -символынан бастап n2 символының нұскасын алады
Ord(B)	B символдың ASCII кодын анықтап береді
Chr(a)	ASCII коды a -ға тең символды анықтайды
Стандартты функциялар	
Delete(S,n1,n2)	S жолды n1 -символынан бастап n2 символын алып тастайды
Insert(S1,S,n)	S жолдан n -орыннан бастап S1 жолды орналастырады
Str(a,S)	S жолдық айнымалының мәні a санның жолдық түріндегі өрнегіне тең болады
Val(S,a,c)	a сандық айнымалының мәні S жолдық айнымалының сан түріндегі өрнегіне, ал c нөлге тең болады (егер берілген жолды сан көрінісінде өрнектеп болмаса, a -ның мәні 1-ге тең болады, ал c -тің мәні нөлден ерекшеленеді)

1. $a='Сау денедә '$, $b='сау ақыл.'$ болса, $c:= \text{Concat}(a,b)$; операторы орындалғанда, c -тің мәні 'Сау денедә сау ақыл.' -ға тең. Бірақ $c:= \text{Concat}(a,b)$; орнына $c:=a+b$; деп жазу да керекті нәтижені береді.

2. $a='информатика'$ болса, $n:=\text{Length}(a)$; операторы орындалғанда, n -нің мәні 11-ге тең болады.

3. $a:=\text{Pac}('m', 'информатика')$; операторы орындалса, a -ның мәні b -ға, ал $a:=\text{Pac}('ma', 'информатика')$; операторы орындалса да, a -ның мәні b -ға, $a:=\text{Pac}('sn', 'сынып')$; операторы орындалса, a -ның мәні 0 -ге, ал $a:=\text{Pac}('v', 'сынып)$; операторы орындалса да, a -ның мәні 0 -ге тең болады.

4. $a:=\text{Copy}('информатика',3,5)$; операторы орындалса, a -ның мәні 'форма' сөзіне тең болады

5. $a:='A'$ болса, $\text{Ord}(a)$ функцияның мәні 65-ке тең болады. Өйткені 'A' (латын) әрпінің ASCII коды 65. Ord функциясының аргументі тұрақты болса, ол апостроф ішіне жазылады. Мысалы, $\text{Ord}('A')$.

6. $\text{cod}=65$ болса, **Chr**(cod) функциясының мәні латын 'A' әрпіне, ал **Chr**(66) функциясының мәні латын 'B' әрпіне тең болады.

7. $a:=\text{'орындалмады'}$ болса, **Delete**(a,8,2); процедурасын орындаған соң, нәтиже $a:=\text{'орындалды'}$ болады. Мұны схема түрінде былай бейнелеуге болады:

($a:=\text{'орындалмады'}$ → **Delete**(a,8,2); → 'орындалмады'
 $a:=\text{'орындалды'}$)

8. $a:=\text{'орындалды'}$, $b:=\text{'ма'}$ болса, **Insert**(b,a,8); процедурасын орындаған соң, нәтиже $a:=\text{'орындалмады'}$ болады. Бұны схема түрінде былай жазуға болады:

($a:=\text{'орындалды'}$, $b:=\text{'ма'}$ → **Insert**(b,a,8); → $\text{'орындал'+\text{'ма'+\text{'ды'}}$ → $a:=\text{'орындалмады'}$)

9. $a:=765$ болса, **Str**(a,s); процедурасын орындаған соң, $s:=\text{'765'}$ болады.

10. $s:=\text{'123'}$ болса, **Val**(s,a,c); орындалған соң, $a:=123$ және $c:=0$ болады; $s:=\text{'34BMA5'}$ болса, **Val**(s,a,c); орындалған соң, $a:=0$ және $c \neq 0$ болады.

Енді қарастырылған функциялар мен процедуралардың іс жүзінде пайдаланылуына есептер келтіреміз:

1-есеп. Берілген 'тәуелсіз', 'Өзбекстан', 'мемлекет' сөздерінен 'Өзбекстан тәуелсіз мемлекет' деген сөз тіркесін шығаратын бағдарлама жаса.

Шешуі. Берілген сөздер бағдарламаның жұмыс барысында өзгермейді. Сондықтан олар константа ретінде өрнектеледі.

```
Program Сөз_жасау;
Const a='тәуелсіз'; b='Өзбекстан'; c='мемлекет';
Var d : String;
Begin
    d:=Concat(b, ' - ' a, c); WriteLn(d);
End.
```

2-есеп. Енгізілген сөзді теріс аударып беретін бағдарлама жаса. Мысалы, 'ота' сөзінен 'ато' сөзі шықсын.

Шешуі. Енгізілген сөзді a деп, жасалатын сөзді b деп белгілейміз. b -ның мәні бас жолға теңеледі ($b:=\text{' '}$). a -ның ұзындығы анықталып, сол жағынан бастап бір-бірлеп белгісін алып, b -ға сол жақтан біріктіреміз.

```
Program Керісінше;
Var a, b, belgi: String; i, len : Integer;
Begin
    Write('Сөз енгіз : '); ReadLn(a); Len:=Length(a); b:=\text{' '};
    For i:=1 To len Do begin
```

```
belgi:=Copy(a,i,1); {a нын i-белгісінің көшірмесі алынды}
b:=belgi+b;        {a дан нұсқасы алынған белгі b-ның сол
                   жағына қосылды}
end;
```

```
Write(b); ReadLn;
```

End.

3-есеп. Берілген сөзде берілген белгі бар немесе жоқ екенін анықтайтын бағдарлама жаса.

```
Program Іздеу;
```

```
Var suz : String; belgi : Char;
```

```
Begin
```

```
Write('Сөз енгіз : '); ReadLn(сөз);
```

```
Write('ізіліп жатқан белгіні: енгіз '); ReadLn(белгі);
```

```
If Pos(белгі,сөз)>0 Then WriteLn('БАР') Else
```

```
WriteLn('ЖОК');
```

```
End.
```

Бағдарламаны орындаған соң, компьютер экранында 'БАР' не 'ЖОК' сөздерінің бірі көрінеді.

4-есеп. 'саодат' және 'кадок' сөздерінен 'садокат' сөзін тудыратын бағдарлама жаса.

```
Program Сөз_жасау;
```

```
Var a,b : String;
```

```
Begin
```

```
a:='саодат'; b:='кадок';
```

```
Delete(a,3,2); {a='саат' болды}
```

```
Delete(b,1,2); {b='док' болды}
```

```
Insert(b,a,3); {a='садокат'}
```

```
WriteLn(a);
```

```
End.
```

5-есеп. Берілген 'Ұлы ақын Әлішер Науаи', 'жылы туылған' сөз тіркестері және 1441 санынан 'Ұлы ақын Науаи 1441 жылы туылған' деген жолды шығаратын бағдарлама жаса.

```
Program Әлішер_Науаи;
```

```
Const a=' Әлішер Науаи '; b=' – жылы туылған'; c=1441;
```

```
d='Ұлы ақын';
```

```
Var жыл, s : String;
```

```
Begin
```

```
Str(c, жыл; {c=1441 дан жыл='1441' жасалды}
```

```
s:=Concat(d,a,жыл,b);
```

```
WriteLn(s);
```

```
End.
```

6-есеп. Берілген бүтін санның цифрлар қосындысын есептейтін бағдарлама жаса.

Program Цифрлар;

```

Var сан, цифр, len, i, c, нәтиже : Integer; _сан, _цифр : String;
Begin Write('Бүгін сан енгіз: '); ReadLn(сан);
      Str(сан, _сан); len:=Length(_сан); нәтиже:=0;
For i:=1 to len Do begin
      _цифр:=Copy(сан, i, 1); Val(_цифр,цифр,c);
      нәтиже:=нәтиже+цифр; end;
WriteLn(сан, 'ның цифрлар қосындысы=', нәтиже);
End.

```

Енгізілген сан 32767 -ден үлкен болса, бұл бағдарлама бұрыс нәтиже береді. Бұның себебін өзің ойланып көр. Бағдарламаға қандай өзгерістер енгізілсе де, 2147483647 дейінгі сандарды да қолдануға болатынын ойлап көр!

Паскальдағы жолдардың «массив қасиеті» бар. Яғни жол – массив деп, ал жолдағы белгілер – массивтің элементтері деп қарастырылады. Мысалы, s – жолдық айнымалы болса, s[3] – сол жолдың 3-белгісін білдіреді. Яғни s:='компьютер' болса, s[3]='m'.

7-есеп. Берілген жолдағы белгілердің ASCII кодтарын экранға шығартып бағдарлама жаса.

```

Program Код;
Var s, b : String;
i, L, cod : Integer;
Begin
      Write('Жол енгіз: '); ReadLn(s);
L := Length(s); { енгізілген жол ұзындығы есептеліп,
L -ге мекшіктенді }
      For i:=1 to L Do WriteLn(Ord(s[i]));
End.

```

Жолдардың «массив қасиеті» арқылы көптеген есептерді оңай шешуге болады. Мысалы, 6-есепті жолдардың сол қасиетін пайдаланып, өзің шешіп көр.



Сұрақтар мен тапсырмалар

1. *Concat* функциясының қызметі қандай? Мысалдар келтір.
2. *Concat* функциясының орнына басқа амалды пайдалануға бола ма?
3. Қайсы функция жол ұзындығын анықтайды? Мысалдар келтір.
4. *Pos* функциясы қандай қызметті атқарады?
5. *Pos* функциясының мәні қашан нөлге тең болады? Мысалдар келтір.
6. *Сору* функциясы не үшін қолданылады?
7. *Сору* функциясының мәні сандық бола ала ма? Мысалдар келтір.
8. Паскаль тілінде берілген жолдың бөлігін өшіретін қандай процедура бар?

9. Паскаль тилинде ерікті символдық яки жолдық шаманы сандық шамаға өткізуге бола ма? Жауабыңды түсіндір.
10. Ord және Chr функцияларының қызметін айт.
11. Жолдық айнымалылардың «массив қасиеті» нені білдіреді?

Жаттығулар

- Төмендегі функциялардың орындалу нәтижесін анықта.
 - Concat('а', 'на');
 - Concat('жа', 'са', 'сын');
 - a:='әлем'; Length(a);
 - Pos('о', 'көктем');
- Жолдық S айнымалының мәні 'Информатика' болсын. Төмендегі амалдар орындалған соң, оның мәнін анықта:
 - Delete(s,5,7);
 - Delete(s,1,2); Delete(s,6,4);
- Енгізілген сөзден «ана» сөзін жасау мүмкіншілігін анықтайтын бағдарлама жаса.

39-сабақ. Символдық және жолдық шамалармен жұмыс істеу тақырыбын қайталау

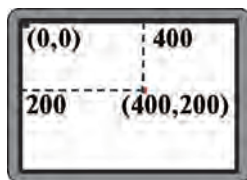
- Берілген сөз белгілерінің арасына бір-бірлеп бос орын қосып шығатын бағдарлама жаса.
- S жолы берілген. Ондағы «b» әріптерінің санын анықтайтын бағдарлама жаса.
- A[1..N] жолдық массив берілген. Массив элементтері ішінен «m» әрпінен басталатындарын экранға шығаратын бағдарлама жаса.
- *. S жолы тек цифрлардан тұрады. Жолдағы цифрлардың ең үлкен санын шығаратын бағдарлама жаса.
- *. A жолдағы белгілердің тек орнын ауыстырып, B жолды жасауға болатынын не болмайтынын анықтайтын бағдарлама жаса.

40-сабақ. Экранды Паскаль тилиндегі графикалық күйге өткізу

Қазірге дейін біз компьютер экранына тек мәтіндік ақпаратты шығарумен таныстық. Бірақ компьютер экранында түрлі суреттер де жасауға болады. Бұл үшін Паскальдің **Graph** (граф) модулінде нүкте, түзу сызық, тік төртбұрыш, шеңбер сияқты бірнеше пішіндер сызуға арналған операторлар бар.

Пішін сызатын операторларды пайдалану үшін бағдарлама басында **Uses Graph**; командасы беріледі. Бұл команда тек оның құрамындағы операторлар мен функцияларды пайдалануға мүмкіндік береді. Ал осы операторларды іске қосу үшін экранды графикалық күйге өткізу керек.

Графикалық күйдегі компьютердің экранын ұсақ нүктелер (пиксельдер) түзеді. Мәтіндік күйдікі сияқты, графикалық күйдің де **курсоры** бар, ол нүктеден тұрады. Экранда пайда болатын барлық сызбалар курсордың экранда із қалдырып яки із қалдырмай жылжуының нәтижесінде жаратылады. Курсордың экранда тұрған орны оның координатасымен анықталады. Координата басы – (0,0) нүктесі экранның сол жақтағы жоғары бұрышында орналасқан. Экранды графикалық күйге өткізгенде курсор координата басында орналасады. X пен Y координата осьтері координата басынан сәйкесінше оңға және төменге қарай бағытталған, яғни мәндер сол бағыттарға қарай өседі. Экранның курсор тұрған нүктесі **ағымдағы нүкте** делінеді. Экрандағы нүктелердің ең көп саны **640x480** (0..639x0..479) болады.



Экранды графикалық күйге өткізу үшін Graph модулінің **InitGraph(GD,GM,<жол>);** процедурасын пайдаланады, мұндағы **GD** (GraphDriver) және **GM** (GraphMode) – бүтін сандық айнымалылар. Олардың мәні компьютердің графикалық мүмкіншілігіне және тандалған графикалық күйге байланысты. Егер GD:=0; (яки GD:=Detect;) деп алынса, бағдарлама ең тиімді графикалық күйді автоматты түрде анықтайды. <жол> – графикалық күйдің жұмысын қамтамасыз ететін, BGI кеңейтуі бар арнайы файл орналасқан **BGI** каталогының адресі, қазіргі заманғы компьютерлерде EGAVGA.BGI файлы қолданылады. Бұл файл ағымдағы каталогта орналасқан болса, <жол> орнына бос жол жазылады.

Графикалық күйден шығу, яғни мәтін күйіне оралу үшін **CloseGraph** процедурасы пайдаланылады. Графикамен байланысты бағдарламалардың негізгі көрінісі мынадай болады:

```
Uses Graph;
  Var gd, gm:integer;
  {Графикамен байланысты есепке тән айнымалылар сипаттамасы }
  Begin
```

```
Gd := 0; {Графикалық драйверді автоматты түрде анықтау}
InitGraph(Gd,Gm, ""); {Графикалық күйге өту}
{Графикамен байланысты есептің шешімі жазылатын бөлім}
ReadLn; CloseGraph; {Графикалық күйден шығу}
End.
```

Паскаль тілінде түрлі 16 түсті пайдалануға болады. Бұл түстер 0-ден 15-ке дейінгі бүтін сандармен кодтандырылған. Graph модулінде әрбір түс үшін арнайы тұрақты (константа) ажыратылған, олар мәтін күйінде істегілетін түстерге ұқсайды (экранмен істеу тақырыбындағы түстер кестесіне қара).

Graph модулінде 80-ге жуық операторлар мен функциялар бар. Енді олардың кейбіреулерімен танысамыз.

PutPixel(X,Y,Түс) процедурасы экранның (X,Y) координатасы бар нүктесін «Түс» параметрімен анықталған түске бояйды. Мысалы, PutPixel(400,200,Red) процедурасы экранның (400,200) координатасына тән қызыл түсті нүкте орналастырады (95-беттегі суретке қара).

GetPixel(X,Y) функциясы экранның (X,Y) координатасы бар нүктесінің түсі қандай екенін анықтайды. Мысалы, Түс – бүтін сандық айнымалы болса, Түс:= GetPixel(40,50); операторы орындалған соң, Түс айнымалының мәні (40,50) координатасы бар нүкте түсінің мәніне тең болады.

GetMaxX және **GetMaxY** функциялар сәйкесінше экранның горизонталь және вертикаль бағытындағы ең жоғары деңгейдегі координатасын анықтайды. Бұл функциялар компьютердің графикалық адаптеріне, сондай-ақ қолданылатын графикалық күйге байланысты болмаған бағдарламалар жасауға пайдалы.

1-есеп. Нүктелер көмегімен экранның (0,0) нүктесін (639,479) нүктесімен қосатын қызыл түсті кесінді сызатын бағдарлама жаса.

Шешуі. Төмендегі бүтін айнымалылардың мәніне қарай мына алынады: bx:=0; ox:= GetMaxX; (ox:=639; сияқты алу мүмкін). Тексеріп көрсен болады, [bx, ox] аралықта y=[GetMaxY·x/GetMaxX] бүтін сандық сызықтық функцияның бастапқы мәні 0, ал соңғы мәні 479 болады (GetMaxY орнына 479 жазуға болады). Енді параметрлік қайталау операторы жәрдемімен құрылған төмендегі бағдарлама есептің жауабын береді.

```
Uses Graph;
Var gd, gm:integer;
    bx, y, ox: Integer; x: LongInt;
```


Begin

Gd := 0; InitGraph(Gd, Gm, "");

bx:=0; ox:= GetMaxX;

For x:= bx to ox do begin

y:= trunc(GetMaxY*x/GetMaxX); putpixel(x, y, red); end;

ReadLn; CloseGraph;

End.

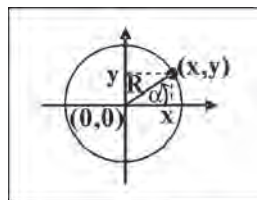
Неликтен x айнымалы LongInt түрінде алынғанын ойланып көрші!

2-есеп. Радиусы R - тең сары түсті шеңберді нүктелер көмегімен экранның ортасына сызатын бағдарлама жаса.

Шешуі. Экранның ортасын анықтайтын бүтін айнымалылар енгізіледі: орталықX:=trunc(GetMaxX/2); орталықY:=trunc(GetMaxY/2); (немесе орталықX:=639; және орталықY:=479). Математикадан белгілі, анықта-

маға сәйкес $\sin \alpha = \frac{y}{R}$ және $\cos \alpha = \frac{x}{R}$. Бұдан шығатыны, ортасы $(0,0)$

нүктеде, радиусы R -ге тең болған шеңбердің нүктелерін мына екі формуланың жәрдемімен анықтауға болады: $x = r \cdot \cos(\alpha)$, $y = r \cdot \sin(\alpha)$, мұндағы α бұрыш 0 -ден 2 -ге дейін өзгереді. Шеңбер нүктелерін тығыздап сызу үшін бағдарламада α бұрыштың өзгерісі $0,01$ қадаммен алынады. Соларға сүйене отырып, мынадай бағдарлама шығарылады:



Uses Graph;

Var gd, gm:integer;

x, y, орталықX, орталықY: Integer; R, alfa: real;

Begin

Gd := 0; InitGraph(Gd, Gm, "");

Write("R= "); readln(R);

орталықX:=trunc(GetMaxX/2); орталықY:=trunc(GetMaxY/2); alfa:=0;

while alfa<=2*pi do begin

x:= орталықX +trunc(R*cos(alfa));

y:= орталықY +trunc(R*sin(alfa));

putpixel(x, y, 14); alfa:= alfa+0.01;

end;

ReadLn; CloseGraph;

End.

3-есеп. Экранның ортасын координата басы деп есептеп, $y=x^2$ функциясының графигін нүктелер көмегімен $x \in [-5, 5]$ аралықта көк түсті етіп сызатын бағдарлама жаса.

Шешуі. Бұл мәселе былай шешіледі:

```
uses Graph;
var gD, gm: Integer;
    орталықX, орталықY: integer; x, y: real;
begin
gD := 0; InitGraph(gD, gm, "");
орталықX:=trunc(getmaxx/2); орталықY:=trunc(getmaxy/2); x:=−5;
    while x<=5 do begin
        y:=x*x; putpixel(trunc(10*x+орталықX), trunc(−5*y+орталықY), blue);
        x:=x+0.01;    end;
    Readln;    CloseGraph;
End.
```

Бағдарламадағы масштабты үлкейту үшін x -ті 10 -ға, y -ті 5 -ке көбейтеді. Парабола бұтақтары төменнен жоғарыға бағытталған болу үшін «-» таңбасы қойылған.

Масштабтын үлкейтілуін, сондай-ақ таңбаны алып тастап, бағдарламаны жүргізіп көруді өзің істе.



Сұрақтар мен тапсырмалар

1. Паскаль тілінің *Graph* модулі қандай мақсатпен пайдаланылады?
2. Графикалық күйде экрандағы нүктенің орны немен анықталады?
3. Экранды графикалық күйге өткізу үшін Паскаль тілінде қандай нұсқау беріледі?
4. Графикалық күйден «шығу» үшін қайсы процедура қолданылады?
5. *PutPixel*, *GetPixel*, *GetMaxX*, *GetMaxY* функциялардың қызметін айт.

Жаттығулар

1. Экранды графикалық күйге өткізіп, **Enter** клавишін басқанда, қайтадан мәтін күйіне қайтаратын бағдарлама жаса.
2. Экранның төрт бұрышына сары түсті нүкте жасайтын бағдарлама жаса.
3. Нүктелердің көмегімен экранды ортасынан бөлетін горизонталь сызық жаса.
4. *Random* функциясын пайдаланып, түрлі түсті нүктелер жасайтын бағдарлама жаса.
5. $y=3x+5$ функциясының графигін $x \in [-10, 10]$ аралықта сызатын бағдарлама жаса.

41-сабақ. Экранды Паскаль тіліндегі графикалық күйге өткізу тақырыбын қайталау

1. Қабырғаларының түсі түрліше болған тік төртбұрыш сызатын бағдарлама жаса.

2. Графикалық координатасымен берілген нүкте ұштары графикалық координаттар арқылы берілген түзу сызыққа тиесілі я тиесілі емес екенін анықтайтын бағдарлама жаса (бағыт: нүктенің түсі түзу сызықтың түсіне тең екенін анықтау үшін GetPixel функциясын пайдалан).

3. Түрлі түсті 15 параллель кесінділер сызатын бағдарлама жаса (бағыт: кесіндінің координаттары мен түсін өшіру үшін қайталау операторын пайдалан).

4. Random функциясының көмегімен «жұлдызды аспан» көрінісін шығаратын бағдарлама жаса.

5. Экранның ортасын кесіп өтетін координаталар осін, координаталар осінің атын тиісті жерде жазатын, сондай-ақ $y = |x|$ функциясының графигін $x \in [-7, 7]$ аралықта сызатын бағдарлама жаса.

6*. Бірінің ішінде екіншісі орналасқан 7 шеңбер сызатын бағдарлама жаса (бағыт: радиусты өшіру үшін қайталау операторын пайдалан).

7*. 7 рет өшіп-жанатын шеңбер сызатын бағдарлама жаса (бағыт: шеңберді сыз да, процесті баяулату үшін бос қайталауды орында, алдыңғы шеңберді фон түсінде сыз да, процесті баяулату үшін бос қайталауды орындап, қайталауды 7 рет орындат).

42-сабақ. Паскаль тілінде пішіндер сызу мүмкіншіліктері

Бұрын көргеніміздей, нүктелердің жәрдемімен де онай болмағанмен, түрлі пішіндер жасауға болады екен. Бірақ Паскаль тілінің дайын пішіндер жасайтын операторлары да бар. Осы операторлар көмегімен алдын ала тандалған бірер түспен түрлі пішін сызуға болады. Паскаль тілінде сызықтың түсін таңдауға **SetColor(түс)** процедурасы, ал фон түсін таңдауға **SetBkColor(түс)** процедурасы орындалады. Бұл жердегі **түс** – бүтін сандық айнымалы я тұрақты шама, ол тандалған түстің кодын яки оған тән константаның атын өрнектейді. Тандалған түс **ағымдағы түс** делінеді. Егер алдын ала бір түс тандалмаған болса, онда ақ түс ағымдағы түс деп танылады.

Line(X1,Y1,X2,Y2) процедурасы экранның (X1,Y1) координатасы бар нүктесін (X2,Y2) нүктесімен біріктіретін ағымдағы түсі бар кесінді сызады.

Circle(X,Y,R) процедурасы ортасы (X,Y) нүктеде және радиусы R -ге тең шеңбер сызады. Енді шеңбермен байланысты есептерді оп-онай шешуге болады.

1-есеп. Ұштарының координаттары $(10,200)$, $(630,200)$ болған көк түсті кесінді, сондай-ақ ортасы $(300,200)$ координаталы нүктеде, радиусы 100 жасыл түсті шеңбер сыз. Фон түсі сары болсын.

```
Uses Graph;
Var gd, gm : Integer;
Begin Gd:=Detect; InitGraph(gd, gm, "");
      SetBkColor(Yellow);
      Setcolor(Blue);   Line(10,10,630,200);
      Setcolor(Green);  Circle(300,200,100);
ReadLn;  CloseGraph;
End.
```

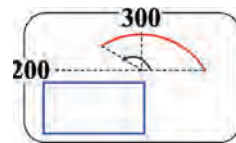


Ellipse(X,Y,BB,OB,XR,YR) процедурасы ортасы (X,Y) нүктеде, x және y осьтері бағытындағы радиустары сәйкесінше XR мен YR -ге тең эллипстің BB бұрышынан OB бұрышына дейінгі осін сызады. Бұрыш градус өлшем бірлігінде беріледі. $XR=YR$ болса, шеңбердің доғасы сызылады.

Rectangle(X1,Y1,X2,Y2) процедурасы экранда жоғарғы сол жақтағы бұрышы $(X1,Y1)$ координаталы, ал төменгі оң жақтағы бұрышы $(X2,Y2)$ координаталы нүктеде болған тік төртбұрыш сызады.

2-есеп. Ортасы $(300,200)$ координаты бар нүктеде, x осі бойымен радиусы 100, y осі бойымен радиусы 50, бастапқы бұрышы 0° , соңғы бұрышы 135° болған қызыл түсті ось, сондай-ақ солдан жоғары және оңнан төменгі ұштары сәйкесінше $(10,220)$ және $(300,400)$ координаталы нүктелерде болған көк түсті тік төртбұрыш сыз.

```
Uses Graph;
Var gd, gm : Integer;
Begin
      Gd:=Detect; InitGraph(gd, gm, ""); Setcolor(4);
      Ellipse(300,200,0,135,100,50);
      Setcolor(1); Rectangle(10,220,300,400);
      ReadLn;  CloseGraph;
End.
```



DrawPoly(BS,KM) процедурасы сынық сызық сызады. BS — сынық сызықтың сыну нүктелерінің саны, KM — сынық сызық сызу нүктелерінің координаттары берілген массивтің аты. Егер сынық сызықтың бастап-қы нүктесінің координатасы мен соңғы нүктесінің координатасы бір-бірінің үстіне түссе, жабық сала — көпбұрыш пайда болады.

Паскаль түрлі түрлі түспен әрі әдіспен боялған пішіндер сызатын процедуралары да бар. Осы пішіндердің шекаралық сызықтар түсі біз жоғарыда көрген **SetColor** процедурасымен тандап алынады. Олардың ішін бояуға

арналған түс пен әдіс Паскальдің **SetFillStyle(әдіс, түс)**; операторымен белгіленеді, мұндағы **түс** – тандалып отырған түстің коды, **әдіс** – бояу әдісі.

Паскаль тілінде пішіндер ішін тандалған түспен бірнеше әдіспен бояуға болады. Бояу әдістері түстер сияқты бүтін сандармен кодталған. Graph модулінде әрбір бояу әдісіне арнайы константалар да ажыратылған. Төмендегі кестеде бояу әдістері, олардың кодтары және тиісті константалардың аттары келтірілген:

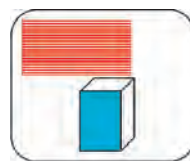
Бояу әдісі	Коды	Константа аты
Фон түсіне бояу 	0	EmptyFill
Берілген түске бояу 	1	SolidFill
Қалың горизонталь сызықтармен толтыру 	2	LineFill
Жіңішке оңға қарай еңіс сызықтармен толтыру 	3	LtSlashFill
Қалың оңға қарай еңіс сызықтармен толтыру 	4	SlashFill
Қалың солға қарай еңіс сызықтармен толтыру 	5	BkSlashFill
Жіңішке солға қарай еңіс сызықтармен толтыру 	6	LtBkSlashFill
Тор сызықпен толтыру 	7	HatchFill
Еңкіш тор сызықпен толтыру 	8	XHatchFill
Қою еңіс сызықтармен толтыру 	9	InterLeaveFill
Сирек нүктелермен толтыру 	10	WideDotFill
Қою нүктелермен толтыру 	11	CloseDotFill
Пайдаланушы белгілеген әдіспен бояу	12	UserFill

Bar(X1,Y1,X2,Y2) процедурасы экранда жоғарғы сол жақтағы бұрышы (X1,Y1) және төменгі оң жақтағы бұрышы (X2,Y2) координаталы нүктелерде, іші ағымдағы түске және әдіспен боялған тік төртбұрыш сызады.

Bar3D(X1,Y1,X2,Y2,a,b) процедурасы ағымдағы түске және әдіспен боялған параллелепипед сызады. Бұл жердегі **a** – параллелепипед қабырғасының ұзындығы, ал **b** – логикалық өрнек, егер оның мәні «рас» болса, параллелепипедтің жоғары ұшы сызылады, «жалған» болса, сызылмайды.

3-есеп. Іші қызыл түсті қалың горизонталь сызықтармен толтырылған тік төртбұрыш және көк түсті параллелепипед сыз.

```
Uses Graph;
Var gd, gm : Integer;
Begin
    Gd:=Detect; InitGraph(gd,gm, "");
    SetFillStyle(2, 4); Bar(10,10,400,200);
```



```
SetFillStyle(1,9); Bar3D(100,200,350,400,50,True);
ReadLn; CloseGraph;
```

End.

FillEllipse(X,Y,XR,YR) процедурасы ортасы (X,Y) координаталы нүктеде, X және Y осьтері бағытындағы радиустары (ені мен бойы) сәйкесінше XR және YR -ге тең, ағымдағы түске және әдіспен боялған эллипс сызады.

4-есеп. Іші көк түсті еңіс тор сызықтармен толтырылған эллипс сыз.

```
Uses Graph;
```

```
Var gd, gm : Integer;
```

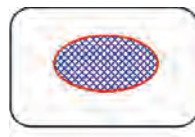
```
Begin
```

```
Gd:=Detect;InitGraph(gd,gm, ""); SetColor(Red);
```

```
SetFillStyle(8,1); FillEllips(320,240,200,100);
```

```
ReadLn; CloseGraph;
```

End.



FillPoly(BS,KM) процедурасы ағымдағы түске және әдіспен боялған көпбұрыш сызады. BS – көпбұрыштың бұрыштар саны, KM – көпбұрыш ұштарының координаттары берілген массив. Бұл оператордың DrawPoly операторынан айырмашылығы сол, ол көпбұрыштың бірінші ұшы мен соңғы ұшын өзі біріктіріп алады.

5-есеп. Ұштарының координаттары KM массив арқылы берілген және іші қызыл түсті торкөзді сызықтармен толтырылған алтыбұрыш сыз.

Шешуі. Ұштарының саны 7 болған сынық сызық сызу операторын пайдаланып, 7-ұшының координатасы 1-ұшы координатасымен бірдей етіп берілсе, алтыбұрыш пайда болады.

```
Uses Graph;
```

```
Const bs=6;
```

```
Var gd, gm : Integer;
```

```
km : Array[1..bs,1..2] of Integer;
```

```
Begin gd:=0; InitGraph(gd,gm, ""); Setcolor(4);
```

```
SetFillStyle(7,4); {әдіс және түс тандалады}
```

```
km[1,1]:=300; km[1,2]:=10;
```

```
km[4,1]:=400; km[4,2]:=190;
```

```
km[2,1]:=200; km[2,2]:=80;
```

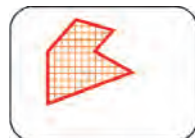
```
km[5,1]:=300; km[5,2]:=80;
```

```
km[3,1]:=200; km[3,2]:=200; km[6,1]:=400; km[6,2]:=40;
```

```
FillPoly(bs,km); {eger DrawPoly(bs,km); жазылса іші бос
```

```
алтыбұрыш сызады}
```

End.





Сұрақтар мен тапсырмалар

1. Экранда бір пішін сызу үшін оның түсін қайсы процедура арқылы таңдайды?
2. Паскальдің кесінді сызу мүмкіншілігін іс жүзінде көрсет.
3. Шеңбер қайсы процедураның жәрдемімен сызылады?
4. Эллипс процедурасының көмегімен қандай пішіндер сызуға болады?
5. Тік төртбұрыш сызу процедурасындағы $x1$, $y1$, $x2$, $y2$ нені білдіреді?
6. DrawPoly процедурасы көмегімен қандай пішіндер сызуға болады?
7. Фон түсін өзгертуді іс жүзінде көрсет.
8. SetFillColor көмегімен қандай пішіндер сызуға болады?
9. Іші боялмаған және боялған көпбұрыш сызу әдістерін түсіндір.

Жаттығулар

1. Экранның ортасын кесіп өтетін горизонталь және вертикаль сызық сызатын бағдарлама жаса.
2. Экранның ортасында радиусы 100-ден кіші сары түсті 4 шеңбер сызатын бағдарлама жаса.
3. Экранды сары горизонталь сызықтармен толтыр.
4. Бағдарламашам суретін сызатын бағдарлама жаса.
5. Қызыл түсті дұрыс бесбұрыш сызатын бағдарлама жаса.

43-сабақ. Паскальдің пішіндер сызу мүмкіншіліктері тақырыбын қайталау

1. Өзбекстан жалауының суретін сызатын бағдарлама жаса.
2. Экранның төрт бұрышына ені 60, ал биіктігі 40-қа тең қызыл түсті төртбұрыштар сызатын бағдарлама жаса.
3. Экранды бірдей төрт бөлікке бөліп, оларды сәйкесінше қызыл, сары, жасыл, көк түстерге бояйтын бағдарлама жаса.
4. Экранның ортасында радиусы 100 -ге тең сары түсті шеңбер сызатын бағдарлама жаса.
5. Ай мен жұлдыздарды бейнелейтін түнгі аспан көрінісін сызатын бағдарлама жаса.
6. Теңіз үстінен сәулесін шашып тұрған Күннің суретін сызатын бағдарлама жаса. Теңіз сызғанда, ось сызу операторын пайдалан.
7. Іші сирек қызыл нүктелермен толтырылған дұрыс алтыбұрыш сызатын бағдарлама жаса.
- 8*. 12 түрлі бояу әдісін көрсететін, өлшемі 40x40 болған 12 квадрат сызатын бағдарлама жаса.
- 9*. Бағдарламашам шырақтарын тізбектеп жандыратын бағдарламаның суретін сызатын бағдарлама жаса.

44-сабақ. Файлдармен жұмыс істеу

Біз енгізу-шығару операторларымен таныстық. Енгізу операторының жәрдемімен деректерді клавиатура арқылы енгізуге болатын болса, шығару операторы деректерді экранға шығарады екен. Кейде сондай есептер кездеседі, шығарылатын деректер экранға сыймай қалады. Кейбір есептердің нәтижелерін келешекте пайдалану үшін сақтап қою керек болады. Мұндай жағдайда қажетті деректерді файл түрінде сақтау жөн. Сен файлдар және олардың (мәтіндік, графикалық және т.б.) форматтары туралы жеткілікті мәлімет алдың. Паскаль бағдарламалау тілі түрліше форматталған файлдармен істеуге мүмкіндік береді. Паскальда файлдармен істеу үшін арнайы **файл түріндегі айнымалылар** (файлдық айнымалылар) енгізілген. Файлдық айнымалылар арқылы сыртқы жадында орналасқан файлдар өрнектеледі. Сандық айнымалылардың мәні сан, жолдық айнымалылардың мәні жол болғаны сияқты, файлдық айнымалылар мәні файл болады. Файлдық айнымалылар да сипатталуға тиіс.

Біз төменде мәтіндік файлдармен ғана жұмыс істеуге үйренеміз. Мәтіндік файлдар түрлі ұзындықтағы жолдардан тұрады. Мұнда жол ұзындығы 256 белгіден аспауы керек. Мәтіндік файлдарды бейнелейтін айнымалылар Паскаль тілінің **text** көмекші сөзімен сипатталады, мысалы:

var fmatn : Text; {fмәтін – мәтінді файл}

Файлдармен жұмыс істеу үшін Паскальда төмендегі амалдарды орындау қажет:

1. Файлдық айнымалыны сыртқы жадындағы файлмен байланыстыру;
2. Файлды «оқу» я «жазу» үшін ашу;
3. Файлдағы деректерді оқу я оларды файлға жазу;
4. Файлды жабу.

Файлдық айнымалы сыртқы жадындағы файлмен **Assign(f, <файл аты>)**; операторы арқылы байланысады. Бұл жердегі **f** – файлдық айнымалы; **<файл аты>** – сыртқы жадында орналасқан файлдың атын өрнектейтін жолдық айнымалы яки тұрақты. Егер файл ағымдағы каталогта болмаса, оның **толық аты** көрсетіледі. Мысалы, «Наурыз.txt» файлы «D» дискінің «Мейрам» атты каталогына орналасқан болса, оның толық аты мынадай болады:

d:\ Мейрам\Наурыз.txt

Осы файлды f – файлдық айнымалымен байланыстыру үшін Assign операторы былай жазылады:

```
Assign(f, 'd:\ Мейрам\Наурыз.txt');
```

Assign процедурасы файлдық айнымалының мәнін, яғни сыртқы жадындағы нақты бір файлды белгілейді. Бұл файлға жүгіну (ондағы деректерді оқу немесе оған деректер жазу) үшін оны «ашу» керек. Мәтіндік файлды бір уақытта әрі оқу, әрі жазу үшін ашуға болмайды. Оқуға арналып ашылған файлдағы деректерді тек оқуға болады. Ал жазу үшін ашылған файлға тек деректерді жазуға болады. Паскаль тілінде файлдарды «жазу үшін» мына екі түрлі әдіспен ашуға болады:

1) жаңа файл жасау және оны жазу үшін ашу;

2) бар файлды, деректердің оған енгізілуін одан әрі жалғастыру үшін ашу.

Rewrite(f) процедурасы сыртқы жадында жаңа файл жасайды және оны жазу үшін ашады. Ол Assign процедурасының жәрдемімен белгілі бір файлмен байланысқан болуы керек. Мысалы, Assign(f, 'сүмелек.txt'); Rewrite(f); процедурасы орындалған соң, ағымдағы каталогта «сүмелек.txt» атаулы жаңа мәтіндік файл жаратылып, оған деректер жазу үшін ашылады. Егер ағымдағы каталогта дәл осындай атты файл бұрыннан бар болса, ол өшіп, оның орнына жаңа файл жазылады.

Файлға керекті мәліметтер жазып болған соң, оны сөзсіз жабу керек. Паскальда ашылған файл **Close(f)** процедурасының жәрдемімен жабылады. Бұл процедура оқуға арналып ашылған файлдарды да, жазуға арналып ашылған файлдарды да жабады. Сыртқы жадындағы файлдық айнымалыға сай файл, Close процедурасын қолданудан бұрын ашылған болу керек.

Деректерді файлға жазу (деректерді файлға шығару) үшін мына операторлар қолданылады:

```
Write(f, <шығару тізімі>); және  
WriteLn(f, <шығару тізімі>);
```

мұндағы **f** – файлдық айнымалы, **<шығару тізімі>** – бір-бірінен үтірмен ажыратылған бір я бірнеше айнымалы яки тұрақты. Тап осы операторлар **<шығару тізімінде>** көрсетілген айнымалылар мен тұрақтылар мәндерін файлға жазады.

1-есеп. Ағымдағы каталогта «Әнұран.txt» атаулы файл жасап, клавиатурадан енгізілген әнұранның 4 -жолын файлдың арнайы жолдарына жазып қоятын бағдарлама жаса.

Шешуі. Енгізілген мәліметтерді файлдың арнайы жолдарына жазу үшін WriteLn операторы пайдаланылады..

```
Program Әнұран1;
Var _әнұр : Text;   жол : String; m:integer;
Begin
    Assign(_әнұр, 'Әнұран.txt'); Rewrite(_әнұр);
    For m:=1 to 4 do      begin
        Write('Әнұранымыздың ', m, 'жолын енгіз: '); ReadLn(жол);
        WriteLn(_әнұран, жол); end;
    Close(_әнұр);
End.
```

Біз сыртқы жадында жаңа файл жасауды көрдік. Кейде сыртқы жадына сақтап қойылған бір мәтіндік файлды жалғастыру, яғни оған жаңа деректер қосу керек болады. Сонда Rewrite процедурасының орнына **Append(f)** процедурасы пайдаланылады. Бұл процедура көрсетілген сыртқы жадындағы файлды «жазу үшін» ашады. Көрсетілген файл сыртқы жадында болмаса, қате болып қалады.

Демек, Append процедурасының көмегімен ашылуға тиісті файл сыртқы жадында сөзсіз бар болу керек.

2-есеп. 1-есепте құрастырылған «Әнұран.txt» файлы аш, әнұранымыздың 4 жолына қайырман кіргізетін бағдарлама жаса.

Шешуі. Файлды жалғастыру үшін оны Append процедурасымен ашады.

```
Program Әнұран2;
Var f : Text;   қайырма: String; m:integer;
Begin
    Assign(f, 'Әнұран.txt'); Append(f);
    For m:=1 to 4 do      begin
        Write('қайырманьң ', m, 'жолын енгіз: '); ReadLn(қайырма);
        WriteLn(fқайырма); end;
    Close(f);
End.
```

Reset(f) процедурасы дискідегі файлдарды оқу үшін ашады. Ол Assign процедурасы арқылы сыртқы жадындағы бір файлға байланыстырылған болу керек. Көрсетілген файл сыртқы жадында болмаса, қате болып қалады.

Оқу үшін ашылған файлдың деректерін мына операторлар көмегімен оқиды:

Read(f, <айнымалылар тізімі>); және
ReadLn(f, <айнымалылар тізімі>);

Мұндағы **f** – файлдық айнымалы; <**айнымалылар тізімі**> – бір немесе бір-бірінен үтірмен ажыратылған бірнеше айнымалылар. Тап осы операторлар көрсетілген айнымалылар мәндерін файлдан оқып алады.

Read операторы, негізінде, файлға сандық деректер жазылған болса, яғни файлдың жолы бір-бірінен бос орынмен ажыратылған сандардан тұратын болса ғана қолданылады. Read операторы бұл сандардың әрқайсысын бөлек оқып шығады. Бір жолдағы деректер аяқталған соң, кейінгі жолға өтіледі.

3-есеп. Кабырғаларының ұзындықтары «үшбұр.in» файлында келтірілген үшбұрыштың ауданын есептейтін бағдарлама жаса. «үшбұр.in» файлының бір-ақ жол түзеді, ондағы үшбұрыш кабырғалары ұзындықтарын өрнектейтін үш сан бір-бірінен пробелмен ажыратылып жазылған.

Шешуі. «үшбұр.in» файлындағы деректерді оқу үшін Read операторын пайдаланамыз. Үшбұрыштың ауданын Герон формуласы бойынша есептейміз.

```
Program үшбұрыш_ауданы;
Var f : Text;   a, b, c, yp, s : Real;
Begin
  Assign(f, 'үшбұр.in');
  Reset(f); {«үшбұр.in» файлы оқу үшін ашылды}
  Read(f, a); Read(f, b); Read(f, c);
  {a,b,c мәндері «үшбұр.in» файлынан оқып алынды}
  Close(f);   {« үшбұр.in» файлы жабылды }
  yp:=(a+b+c)/2; s:=sqrt(yp*(yp-a)*(yp-b)*(yp-c));
  WriteLn('үшбұрыш ауданы=', s);
```

End.

ReadLn операторы файлдағы жолды толығымен оқиды. Мәтіндік файлдардағы жолдарды реттік нөміріне қарай кезек-кезек оқуға болады. Мысалы, файлдың 10-жолын оқу үшін алдағы 9-жол әрине, оқып шығылған болу керек.

4-есеп. «сынып.txt» файлында 9-сынып оқушыларының тізімі (әрбір жолға бір оқушының тегі) берілген. Сол тізімдегі 12-оқушының тегін экранға шығаратын бағдарлама жаса.

Шешуі: 12-оқушының тегі «сынып.txt» файлының 12-жолында жазылған. Оны оқу үшін алдағы 11-жолды оқып шығу керек. Бұл үшін параметрлік қайталау операторы пайдаланылады.

```
Program Сынып;
var fio : Text;   i : Integer; fam : String;
Begin
  Assign(fio, 'sinf.txt'); Reset(fio);
  For i:=1 to 11 Do ReadLn(fio, fam);
  ReadLn(fio, fam); Close(fio);
```

WriteLn('12-оқушының фамилиясы:', fam);

End.

Осы есептегі «сынып.txt» файлында берілген барша оқушылардың фамилияларын экранға шығару керек болса, ділгірлік туындайды. Өйткені, «сынып.txt» файлы неше жолдан тұратыны белгісіз. Мұндай жағдайда Паскаль тілінің **Eof(f)** функциясы қолданылады. Eof – логикалық функция, оның мәні, егер файлдағы оқуға арналған деректер қалмаса, «Рас», ондай болмаса, «Жалған» болады.

5-есеп. «сынып.txt» файлында 9-сынып оқушыларының тізімі берілген. Тізімдегі барша оқушылардың фамилияларын экранға шығаратын бағдарлама жаса.

Шешуі. Шартқа сәйкес қайталау операторы – While пайдаланылады.

```
Program Сынып;
var fio : Text;   fam : String;
Begin
    Assign(fio, 'sinf.txt'); Reset(fio);
    While Not(eof(fio)) Do begin
        ReadLn(fio, fam);
        WriteLn(fam);      end;
    Close(fio);
```

End.

Бағдарламаның қайталау операторының шарты – Not(eof(fio)) «жалған» мәнді қабылдамағанша, яғни eof(fio) функциясы «рас» мәнін қабылдамағанша, қайталанымдар жалғаса береді. Eof(fio) функциясы «рас» мәнін қабылдап алысымен, яғни «сынып.txt» файлында оқу үшін деректер аяқталысымен, қайталаным тоқтайды. Керек болғанда, бір бағдарламада бірнеше файлдар ашуға болады. Егер олар кезекпен, яғни біріншісі жабылған соң екіншісі ашылатын болса, бір файлдық айнымалыны пайдалануға болады. Ондай болмаған жағдайда олардың әрқайсысына бөлек файлдық айнымалыны сипаттауға тура келеді.



Сұрақтар мен тапсырмалар

1. *Файлдық айнымалы дегенде нені түсінесің?*
2. *Файлдық айнымалы сыртқы жадындағы файлмен қай процедураның көмегімен байланысады?*
3. *Rewrite процедурасының қызметін айт.*
4. *Ашылған файлды жабу үшін қайсы процедура қолданылады?*
5. *Деректерді мәтіндік файлға жазу үшін қайсы операторлар пайдаланылады?*
6. *Файлға жаңа мәліметтер қосу үшін оны қайсы процедураның көмегімен ашу керек?*
7. *Деректерді оқу үшін файлды қайсы процедураның көмегімен ашады?*

8. Деректерді мәтіндік файлдан оқу үшін қайсы операторлар пайдаланылады?
9. Read операторының ReadLn операторынан айырмашылығы неде екенін айт.
10. ReadLn операторына бірнеше айнымалы қатысса, олар қалай жазылады?
11. Eof функциясының қызметін айт.

Жаттығулар

1. Апта күндерінің атауларын енгізіп, «АПТА.ТХТ» файлына сақтап қоятын бағдарлама жаса.
2. 1-жаттығудағы «АПТА.ТХТ» файлын ашып, оның артынан апта күндерінің орыс тіліндегі атауларын жазатын бағдарлама жаса.
3. «АПТА.ТХТ» файлында берілген апта күндерінің атауларын экранға шығаратын бағдарлама жаса.

45-сабақ. Файлдармен жұмыс істеу тақырыбын қайталау

1. Сыныптастарының фамилиялары мен есімдерінен құрастырылған «СЫНЫП.ТХТ» атаулы мәтіндік файл жасайтын бағдарлама жаса.
2. 9-сынып оқушыларының «сынып.txt» файлында берілген фамилиясы арасынан «М» әрпімен басталатындарды экранға шығаратын бағдарлама жаса.
3. «сынып.txt» файлында берілген 9-сынып оқушылары фамилиялары арасынан «Б» әрпінен басталатындарын айырып алып, олардан «bсынып.txt» файлын шығаратынын бағдарлама жаса.
- 4*. $y = \sin^2 x$ функциясының $[-\pi, \pi]$ аралықтағы мәндерін 0,01 қадаммен есепте. Нәтижесін «sinus.out» файлына сақтап қой.
- 5*. «sinus.out» файлына түсіндірме беретін бағдарлама жаса.

46-сабақ. Процедуралар мен функциялар

Көбінесе белгілі амалдар кешенін бағдарламаның түрлі бөлігінде қайталауға тура келеді. Паскальда көп қайталанатын амалдар кешенін негізгі бағдарламадан ажыратып алып, олардан арнайы блоктар — **процедуралар** мен **функциялар** құруға болады. Осылай құрылған әрбір процедура мен функцияға сөзсіз атау беріледі. Керек процедура мен функцияға оның **атауы** арқылы жүгінеді. Процедуралар мен функцияларды тиімді пайдаланып, құрастырылған бағдарлама әрине, қарапайым да түсінікті болады.

Процедуралар да, функциялар да белгілі амалдар кешенінен тұратын болса, олардың бір-бірінен айырмашылығы неде?

Жалпы алғанда, функциялар белгілі бір мәнді есептеуге арналып түзіледі. Ең ақырында есептелген мән функцияның атына меншіктеледі.

Ал процедуралар белгілі бір амалдар тізбегін орындау мақсатымен түзіледі. Мұнда белгілі бір нәтижелі мәнді есептеу емес, берілген амалдар тізбегін орындаудың өзі маңызды болады.

Мысалы, бағдарламада қабырғалары берілген үшбұрыштың ауданын бірнеше рет есептеу керек болса, **функцияны** пайдалану керек болады. Өйткені, бұл жерде әр кез бір-ақ мән — үшбұрыштың ауданы есептеледі. Егер үшбұрыштың берілген қабырғаларына сүйеніп, экранда бірнеше үшбұрыш сызу керек болса, онда әрине бір-ақ мәнді есептеуге арналған функцияны емес, үшбұрыш сызу процесін экранда ұйымдастыратын процедура жасау керек.

Процедуралар мен функцияларды оларға жүгіну үшін берілетін белгілі бір мәндерге байланыстырып ұйымдастыруға да болады. Бұл мәндер **параметрлер** деп, ал осылай ұйымдастырылған процедура мен функция **параметрлік процедура** мен **функция** деп аталады.

Процедуралар мен функциялар тақырыптан басталады. Процедура тақырыбының жалпы көрінісі мынадай:

Procedure <процедура аты> (**параметрлер**);

Функция тақырыбының жалпы көрінісі мынадай:

Function <функция аты> (**параметрлер**) : <функция мәнінің түрі> ;

Параметрлік процедура мен функциядағы параметрлердің түрі де сипатталады. Мысалы,

Function дәреже (a, n : Integer):Integer; яки Procedure shifr(satr:String);

Осыдан көрініп тұрғанындай, процедуралар мен функциялардың құрылысы ұқсас. Олардың құрылысы бағдарламаның құрылысынан айтарлықтай ерекшеленбейді. Ескерте кету керек, негізгі бағдарламада сипатталған тұрақтылар мен айнымалылар **жалпы (ғаламдық)** тұрақтылар мен айнымалылар деп аталады. Оларды бағдарламаның қалаған бөлігінде, мысалы процедура яки функцияның ішінде пайдалануға болады. Бір процедура яки функцияның ішінде сипатталған тұрақтылар мен айнымалылар **жергілікті (локальный)** тұрақтылар мен айнымалылар деп атала-

Процедураның жалпы құрылысы:	Функцияның жалпы құрылысы
<p>Procedure <процедураның аты> (параметрлер);</p> <p>Label <таңбалар>;</p> <p>Const <константалар сипаттамасы>;</p> <p>Var <айнымалылар сипаттамасы>;</p> <p>«Ішкі» процедуралар мен функциялар</p> <p>begin <процедура денесі> {бағдарлама}</p> <p>end;</p>	<p>Function <функцияның аты> (параметрлер):түрі;</p> <p>Label <таңбалар>;</p> <p>Const <константалар сипаттамасы>;</p> <p>Var <айнымалылар сипаттамасы>;</p> <p>«Ішкі» процедуралар мен функциялар</p> <p>begin <функция денесі> {бағдарлама}</p> <p>end;</p>

ды. Оларды тек өзі сипаттаған процедура мен функцияның ішінде пайдалануға болады. Паскаль тілінде жалпы және жергілікті айнымалылардың аты бірдей болуы да мүмкін. Бұл жағдайда жергілікті айнымалыны сипаттаған процедура яки функция ішіндегі жергілікті айнымалының мәні, басқа процедуралар мен функциялардағы, сондай-ақ негізгі бағдарламадағы жалпы айнымалының мәні ескеріледі.

1-есеп. [20, 83], [178, 391], [211, 746] аралықтардағы барша бүтін сандар қосындысын табатын бағдарлама жаса.

Шешуі. Берілген үш аралықтағы бүтін сандардың қосындысын есептеу үшін параметрлік қайталау операторын үш рет пайдалануға тура келеді. Есепті берілген аралықтың бастапқы мен соңғы мәндерін параметр ретінде алып, сол аралықтағы бүтін сандардың қосындысын есептейтін функцияны пайдаланып та шешуге болады.

```

Program Сумма;
Var i, s, s1, s2, s3 : Integer;
Function Sum(n1, n2:Integer):Integer;
Begin
    s:=0; For i:=n1 To n2 Do s:=s+i; Sum:=s;
End;
Begin
    S1:= Sum(20,83); s2 := Sum(178,391); s3:=Sum(211,746);
    S:=s1 + s2 + s3; WriteLn('S= ', s)
End.
```

2-есеп. Үшбұрыштың берілген қабырғаларына сүйеніп, оның биіктігін есептейтін бағдарлама жаса.

```

Program үшбұрыш_биіктігі;
Var a, b, c, ha, hb, hc: real;
```

```

Function H_UB(a, b, c: real): real; { a, b, c – үшбұрыш
кабырғалары }
  Var yp, s: real;
  Begin
    yp:= (a+b+c)/2; {Периметрдің жартысын есептеу }
    s:= Sqrt(yp*(yp-a)*(yp-b)*(yp-c)); {Ауданын есептеу}
    H_UB:= 2*s/a; {Функцияға мән берілді }
  End;
Begin
  Write(үшбұрыш аттары (a,b,c) енгізілсін '); Readln(a,b,c);
  ha:= H_UB(a, b, c); hb:= H_UB(b, a, c); hc:= H_UB(c,
b, a);
  Writeln(үшбұрыштың биіктіктері: ');
  Writeln('ha= ', ha:10:4, 'hb= ', hb:10:4, 'hc= ', hc:10:4);
Readln
End.

```

3-есеп. Төменде үш үшбұрыштың ұштарының координаттары берілген:
 1) (120,20), (80,170), (140,150); 2) (200,97), (500,156), (210,180);
 3) (300,190), (200,390), (415,222).

Сол үшбұрыштарды сәйкесінше қызыл, сары, жасыл түстерге бояп сызатын бағдарлама жаса.

```

Uses Graph;
  var gd, gm: Integer;
  Procedure Үшбұрыш(x1,y1,x2,y2,x3,y3,col: Integer);
  begin
    SetColor(col); Line(x1,y1,x2,y2); Line(x2,y2,x3,y3);
    Line(x3,y3,x1,y1);
  end;
  Begin gd:=0; InitGraph(gd,gm, "");
  Үшбұрыш(120,20,80,170,140,150,4);
  Үшбұрыш(200,97,500,156,210,180,14);
  Үшбұрыш(300,190,200,390,415,222,2); ReadLn;
  CloseGraph;
  End.

```

Жоғарыда келтірілген бағдарламаларға назар аударып, мынадай қорытынды шығаруға болады: процедуралар мен функцияларға жүгінген кезде функцияның аты әрқашан белгілі бір оператор құрамында, ал процедура-ның аты бөлек (дербес) жазылады екен.



Сұрақтар мен тапсырмалар

1. Бағдарламадағы процедура мен функцияны қандай мақсатпен пайдаланады?
2. Процедуралар мен функциялар арасындағы айырмашылық неде?
3. Параметрлік процедуралар мен функциялар туралы айт.

4. Процедураңыз жалпы құрылысын айт.
5. Функцияның жалпы құрылысын айт.
6. Бағдарламадағы жалпы әрі жергілікті айнымалылар туралы әңгімелеп бер.
7. Қандай жағдайда функцияның орнына процедураны қолдануға болады?

Жаттығулар

1. Үш тік төртбұрыштың бір-бірлеп диагональ бағытындағы ұштарының координаттары берілген: 1) 20,20 және 80,200; 2) 200,97 және 500,156; 3) 300,120 және 400,420. Осы тік төртбұрышты сәйкесінше қызыл, сары, жасыл түстерге бояп сызатын бағдарлама жаса.

2. Берілген n натуралды сан үшін $S=1 \cdot 5+2 \cdot 6+3 \cdot 7+\dots+n \cdot (n+4)$ қосындысын есептейтін бағдарламаны процедураның жәрдемімен жаса.

3. Үш санның үлкенін табатын бағдарлама жаса. Бұл үшін екі санның үлкенін шығаратын функция құрастырып, оны пайдалан.

47-сабақ. Процедуралар мен функциялар тақырыбын қайталау

1. Берілген мәтіндегі 'a' белгіні 'g' белгіге, 'm' белгіні 's' белгіге, 'f' белгіні 'h' белгіге ауыстыратын бағдарламаны процедураның жәрдемімен жаса.

2. $y = x^5 + 3x$ функцияның мәнін x -тің $-9, -5, -2, 2, 5, 7$ мәндерінде есептейтін бағдарлама жаса. Дәрежесіні көбейту арқылы есептеуге арналған функция құрастыр.

4. Процедура көмегімен экранның ортасында бірінің ішіне екіншісі орналасқан түрлі түсті 15 шеңбер сызатын бағдарлама жаса.



48–49-сабақ. Қайталауға арналған тапсырмалар

1. Қабырғасының ұзындығы a -ға тең квадрат, сондай-ақ оның ішіне сызылған шеңбер сызатын бағдарлама жасалсын. a -ның мәні клавиатурадан енгізілсін.

2. Радиусы R -ге тең шеңбер және оған суреттеліп жазылған квадрат сызатын бағдарлама жаса. R -дің мәні клавиатурадан енгізілсін.

3. Қабырғаларының ұзындығы a мен b болған тік төртбұрыш және оған жазылған эллипс сызатын бағдарлама жаса. a және b -ның мәні клавиатурадан енгізілсін.

4. Көк түсті фоны бар экранды 20 пиксель болған вертикаль ось сызықтарымен толтыр.

5. Түрлі түсті 15 шеңбер сыз.

6. Үстелдің суретін сыз.

7. Үйдің суретін сыз.

50-сабақ. HTML туралы түсінік

Интернеттің WWW қызметі, негізінде, web-бетпен яки web-бетке байланысты, сондықтан, әрине, «Web-беттер қалай дайындалады?» деген сұрақ туындайды. Web-беттер арнайы бағдарламалар бойынша дайындалады. Бұлар — Microsoft FrontPage, Macromedia HomeSite, Adobe Dreamweaver сияқты редакторлар, PHP, ASP, JavaScript сияқты сервер скриптері (сценарий тілі), XML, HTML және басқалар.

Бұл бағдарламалардың бәрі HTML (Hypertext Markup Language — Гипермәтіндік маркерлеу тілі) тіліне негізделеді. HTML бағдарламалау тілі деп саналмайды. Бұл тілдегі құжат (web-бет) дайындау үшін Windows-тың «Блокнот» қарапайым мәтін редакторының өзі де жеткілікті. HTML тілінің нұсқаулары «<» және «>» белгілері арасында жазылып, дескриптор (ағыл. сипаттаушы) яки тег (ағыл. тамға, белгі, таңба) деп аталады. Мысалы, <HTML> жазбасы HTML тіліндегі құжаттың басталуын білдіреді. Тегтер латын әліпбиінің әріптерімен жазылады. Мұндағы әріптерді жоғары я төменгі регистрде жазудың ешқандай айырмашылығы жоқ, яғни <HTML> және <html> — бірдей тегтер.

Жалпы алғанда, тегтер екі түрге бөлінеді:

1. Жұп тегтер (яки контейнер-тегтер): тегі үшін тегі бар, солардың біріншісі бір амалдың басталғанын білдірсе, екіншісі сол амалдың аяқталғанын білдіреді.

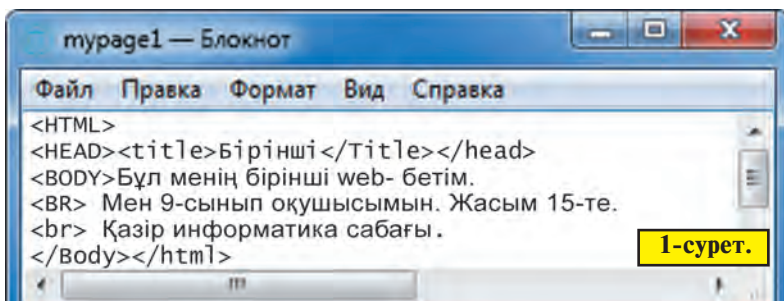
2. Жұп емес тегтер: <D> түріндегі тег ашылады, жабылуы шарт емес, мысалы, өзінен кейінгі мәтінді жаңа жолға өткізетін
 тегіне ұқсас.

HTML-құжат — «html» яки «htm» кенейтуі бар мәтіндік файл, сондай-ақ қарапайым мәтін редакторында тегтер жәрдемімен жазылған мәтін. HTML-құжат <HTML> және </HTML> тегтер арасына жазылған болады. HTML-құжат зердеге жүктелгенде, ол экранда web-браузер жәрдемімен web-бет көрінісінде бйнеленеді.

HTML-құжат әдетте екі бөлімнен тұрады. Солардың біріншісі **HEAD** (ағыл. бас бөлім я тақырып) бөлімі болып, `<HEAD>` және `</HEAD>` тегтері арасында орналасады. Екіншісі **BODY** (дене) бөлімінде құжаттың мазмұны бейнеленеді де, `<BODY>` және `</BODY>` тегтері арасында орналасады. Егер HTML-құжат фрейм-құрылымды өрнектеуге (деректер web-браузер терезесіндегі бөлек салаларда көрінуге) тиіс болса, онда BODY бөлімінің орнына FRAMESET бөлімі (FRAME SET - құрылымдар (рамкалар) құрамасы (жинағы), `<FRAMESET>` жұп тегі жәрдемімен) пайдаланылады. Әдетте, HTML-құжатта `<HEAD>` және `<BODY>` жұп тегтерін жазу ұсынылады, бірақ бұл міндетті емес.

Web-бетке енгізілетін тағы бір элемент – web-беттің аты, оны енгізу үшін `<TITLE>` жұп тегі қолданылады. Web-беттегі бұл тег бір рет қолданылады. Web-беттің аты web-браузер жолында кескінделіп, web-беттің ішінде көрінбейді. Сондықтан оны web-беттің қалаған жеріне (әдетте, HEAD бөлімінде) жазады. Web-бетке кез келген атау, мысалы, өз атыңды беруге болады.

HTML тілі жедел өріс алуда. Өз кезінде web-браузерлер де жаңалануда. Қазіргі таңда web-бет дайындау үшін негізінен HTML-4 тілі пайдаланылады. Оның кейбір нұсқауларын «ескі» web-браузерлер (Internet Explorer-4 я Internet Explorer-6) орындай алмайды. Бәріне белгілі, мысалы, Internet Explorer, Opera, FireFox, Mozilla , Netscape сияқты түрлі web-браузерлер бір-бірінен ерекшеленеді. Сондықтан бір HTML-құжат түрлі web-браузерлерде өзгеше болып бейнеленеуі мүмкін.

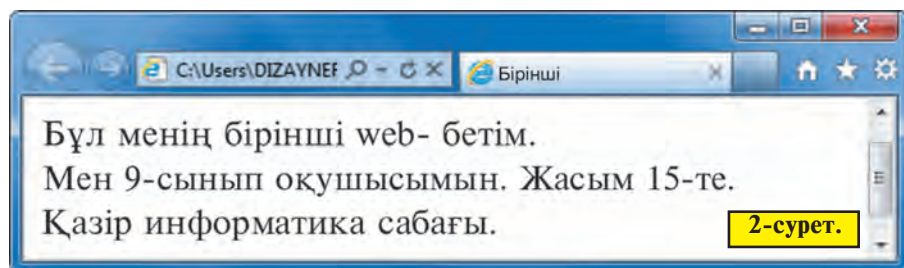


Ең қарапайым web-бетті тек мәтін түзеді. Біз де web-бет дайындауды мәтін орналастырудан бастаймыз. Ол үшін Windows-тың Блокнот мәтін редакторын іске қосамыз (басқа

мәтін редакторын да пайдалануға болады). Әдетте, мәтіндер сияқты, Web-бет те тақырыптан басталады. Бұл біздің бірінші web-бетіміз болғандықтан, оған «Бұл менің бірінші web-бетім» деген тақырып жазамыз. Бұл үшін 1-суреттегі мәтін Блок-нот бағдарламасының жұмыс өрісіне енгізіледі.

Бұл мәтіндері <HTML>, </HTML>, <HEAd>, </head>, <title>, </Title>, <BODY>, </Body> мен
 – HTML тілінің тегтері болып, <HTML> – web-беттің басталатынын, </HTML> – web-беттің аяқталғанын, <HEAd> – тақырып бөлімінің басталатынын, <HEAd> – тақырып бөлімінің аяқталғанын, <title> – ат енгізу басталатынын, </Title> – ат енгізу аяқталатынын, <BODY> – деректер бөлімінің басталғанын, </BODY> – деректер бөлімінің аяқталғанын,
 – мәтіннің жалғасын жаңа жолдан жазу керектігін білдіреді.


Жоғарыда келтірілген мәтінді «**mypage1.html**» атымен бір каталогқа, мысалы, «Мои документы» папкасының ішіндегі «Гүлнар» папкасына сақтап қоямыз. Енді бұл файлдың белгісі web-браузерге бейімделгенін көруге болады. Пайда болған web-бетті Internet Explorer браузерінде ашқанда, 2-суретте келтірілген көрініс пайда болады:



Енгізілген мәтіндік файл мен web-бетті салыстырып, мынадай қорытынды шығарамыз:

- 1) файл атауы (жоғарыдағы мысалда, **mypage1**) web-бетте көрінбейді;
- 2) web-бет атауы (жоғарыдағы мысалда, **Бірінші**) web-браузердің тақырып жолында көрінеді;
- 3) web-бет мәтініндегі арнайы нұсқаусыз жазылған сөз (жоғарыдағы мысалда, «web-бетім.» сөзі, «Жасым 15-те.» мәтіні) өзінен бұрынғы сөзге жалғасып кетеді.

HTML-да енгізілген мәтіннің web-браузердегі көрінісі де біз күткен нәтижені әрқашан бере бермейді. Оны біз қалаған көрініске келтіру үшін бірнеше рет редакциялауға тура келеді. Бұл үшін Блокнот мәтін редакторын іске қосып, қажетті HTML-құжатты дискіден іздеп табу және зердеге жүктеу, оны редакциялау және қайта сақтау, web-браузерді іске түсіріп, редакцияланған HTML-құжатты тағы дискіден тауып, web-браузерде қарау керек болады. Әдетте, web-бетті біз ойлаған көрініске келтіру үшін жоғарыда санап өтілген амалдарды бірнеше рет орындауға тура келеді. Бұл амалдарды өте қолайлы әдістермен да орындауға болады:

1-әдіс. Internet Explorer-де ашылған web-бетті редакциялау керек болса, жүгірткіні сол web-беттің үстіне әкеліп, оң түймесін басады. Ашылған контекст-менюден «HTML-кодын қарау» (**Просмотр HTML-кода**) тарауын тандап алса, экранға web-беттің HTML-құжаты жүктелген Блокнот мәтін редакторы шығады. HTML-құжатты редакциялап, сақтап қояды. Internet Explorer аспаптар панеліндегі «Жаңалау» () түймесін басқанда, деректер өрісіндегі web-бет жаңаланады (редакцияланған web-бетке ауысады).

2-әдіс. Көрініс менюсінен «HTML-кодын қарау» бөлімін тандап алып, 1-әдістегідей редакциялайды.



Сұрақтар мен тапсырмалар

1. HTML тілінің нұсқаулары қалай аталады?
2. HTML-құжат деген не? HTML-құжаттың файлдық атының кеңейтуі қандай?
3. Тегтердің түрін айт.
4. HTML-құжат қандай тегпен басталады?
5. Web-бетке ат қоюды мысалдар арқылы көрсетіп бер.
6. Web-беттің аты web-браузердің қай жерінде бейнеленеді?
7. Web-бет атын HTML-құжаттың қайсы бөлімінде жазуға болады?

Жаттығулар

1. «Менің Отан-анам» атаулы web-бетке арналған мәтін дайында. «Менің Отаным» деген атымен сақта.
2. «Менің Отаным» атаулы мәтінді web-құжат түріне өткіз. Web-бетті web-браузерде ашып, редакцияла.
3. «Біздің мектеп» деген тақырыпқа қарапайым мәтіндік web-бет дайында.

51-сабақ. Мәтінді Web-бетке енгізу

Web-беттегі мәтінде бірнеше тақырыпты баяндауға болады. Бұл жағдайда web-бетте бірнеше тақырып істетуге тура келеді.

Мысалы, жеке web-бетке өзін туралы, қызығушылығың, ұнаған фильмдер туралы мәлімет енгізсең, сәйкесінше «Өзім туралы», «Қызығуларым», «Ұнатқан фильмдерім» сияқты тақырыптар қоюға болады. Кейде бір тақырып мәтінін ірі әріптермен жазу керек болса, қалғандарын ұсақ шрифтпен жазуға тура келеді.

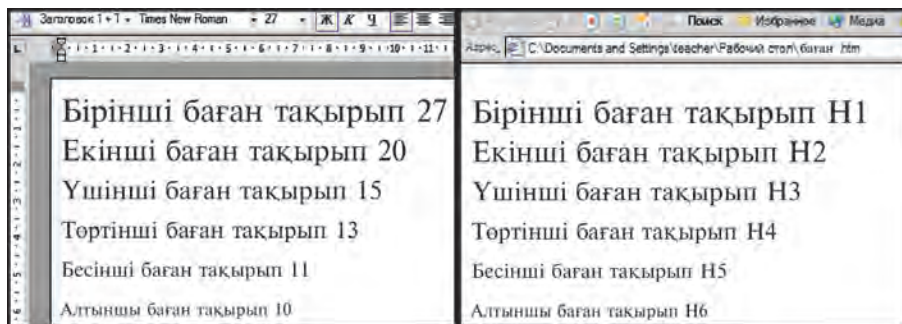
HTML тілі 6 бағаннан тұратын тақырып қоюға мүмкіндік береді. Бұл үшін HTML тілінде мынадай жұп тегтер бар: <H1>, <H2>, <H3>, <H4>, <H5>, <H6>. Демек, бұл тегтердің әрқайсысына сәйкесінше жабылатын (аяқталушы) (</H1>, ..., </H6>) тегтер де бар («H» белгісі – «Heading», яғни ағылшынның тақырып сөзінің бірінші әрпі).

Файл	Правка	Формат	Вид	Справка
<code><HTML></code>				
<code><h1> Бірінші баған тақырып H1 </h1></code>				
<code><h2> Екінші баған тақырып H2 </h2></code>				
<code><h3> Үшінші баған тақырып H3 </h3></code>				
<code><h4> Төртінші баған тақырып H4 </h></code>				
<code><h5> Бесінші баған тақырып H5 </h5></code>				
<code><h6> Алтыншы баған тақырып H6 </h6></code>				
<code></html></code>				

MS Word бағдарламасымен істегенінде, сен шрифт өлшемі (мысалы, 27), шрифт жазу формасы (мысалы, Times New Roman), жазу стилі (Мысалы, Тақырып 1), түске қаныққандығы (мысалы, қалың) сияқты түсініктермен танысқансың. Соларды еске түсіріп, MS Word-пен дайындалған мәтін тақырыптарын HTML тілінде дайындалған web-беттегі тақырыптармен салыстыруға болады (кейінгі беттегі сурет):

Мәтінді Web-бетке енгізу мәтін редакторында сияқтыдай жүзеге асырылады. HTML-құжатқа енгізілген мәтінді web-браузер деректер өрісінің өлшеміне бейімдеп, форматтап алады. Сондықтан енгізілген мәтін браузерде пішіні біраз өзгергендей болып көрінуі мүмкін. Мәтінді web-бетте дәл біз

ойлағандай орналастыру үшін, HTML тілінде мәтін форматына әсер ететін арнайы тегтер бар.



Web-беттегі абзацтарды белгілеу үшін **<P>** жұп тегі қолданылады. Бұл тег абзацтың басында жазылады да, өзінен кейін жазылған мәтін алдында бос жол қалдырады. Бұрын айтылғанындай, мәтіннің жалғасын жаңа жолға өткізу үшін жұп емес **
** тегі қолданылады. Бұл тегті мәтінде бос жол қалдыру үшін де пайдалануға болады.



HTML тегтерін параметрлерімен жазуға болады. Параметрлер тегтің атынан кейін пробелмен ажыратылады. Параметрлерге мән беру керек болса, ол теңдеу белгісінен кейін тырнақшаның ішіне немесе тырнақшасыз жазылады.

Шрифт жазу формасын белгілеу үшін **** (шрифт) жұп тегі **FACE** параметрімен бірге қолданылады: **** мәтін ****. Саған мәлім, шрифт жазудың формасына **Times New Roman**, **Verdana**, **Elephant** сияқтылар енеді. Бір web-бетте бірнеше тақырып формасын қолдануға болады. Бірақ web-браузер шрифт жазудың керекті формасын көрсете алмауы да мүмкін. Сондықтан, керек болған жағдайда осы тег параметрінің бірнеше мәндерін үтірмен ажыратып жазу жөн болады:

****.

Web-браузер тег параметрі мәндерін сол жақтан бастап оқиды, солардың бірі болмаса кейінгісін іздейді. Төменде шрифтін қолданылатын жазу формасына мысал келтірілген.

Кейде мәтіннің белгілі бір бөлігіне назар аудару үшін, кейбір сөздер бірер қасиетімен ерекшеленіп көрсетіледі. MS Word құжатында бұған мәтіндегі сөздерді қалың, еніс я асты сызылған

<pre><HTML> <title>Шрифт жазу түрі</title> Жазу түрі - Times New Roman
 Жазу түрі - Verdana
 Жазу түрі - Elephant </html></pre>	 <p>Жазу түрі - Times New Roman Жазу түрі - Verdana Жазу түрі - Elephant</p>
--	--

түрде жазу арқылы жетуге болатын еді. HTML тілінде де сондай мүмкіндік бар болып, мына жұп тегтер пайдаланылады:

 – қалың шрифт (Bold)	немесе орнына	
<I> – еңіс шрифт (Italik)		 немесе <Cite>
<U> – асты сызылған шрифт (Underline)		–

Жоғарыда көрсетілген тегтерді бірге қолданып, web-беттегі мәтінді қалың да еңіс, еңіс те асты сызылған және басқа көрініске келтіруге болады:

```
<HTML><H1>Мәтіндегі сөздерді ажыратып көрсету</H1>
<p>
Мәтіннің бірер бөлігіне назар аудару үшін кейбір сөздер ерекшеленіп көрсетіледі.
<BR>Бұған мәтіндегі мәлім сөздерді <B>қалың,</B><I> еңіс</I> яки <U>асты сызылған</U> көріністе жазуға болады.
<BR>Тегтерді бірге қолданып, мәтінді <B><I>қалың және еңіс,</B> <U> еңіс және асты сызылған,</I></U><BR>және басқа көрініске келтіру мүмкін.</HTML>
```

Мәтіндегі сөздерді ажыратып көрсету

Мәтіннің бірер бөлігіне назар аудару үшін кейбір сөздер ерекшеленіп көрсетіледі.

Бұған мәтіндегі мәлім сөздерді **қалың**, *еңіс* яки асты сызылған көріністе жазуға болады.

Тегтерді бірге қолданып, мәтінді **қалың және еңіс**, *еңіс және асты сызылған* және басқа көрініске келтіру мүмкін.

MS Word бағдарламасында  аспаптар жәрдемімен мәтінді түрліше (яғни, беттің сол шетіне, ортасына, оң шетіне)

туралауға мүмкін еді. HTML тiлiнде де мәтiндi web-бетте түрлiше орналастырудың мүмкiндiгi бар, бұл амал **<P>** жүп емес тегтерге (яки **<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>**, **<H6>** жүп тегтерге) **ALIGN** (ағыл., туралау) параметрiн қосу арқылы жүзеге асырылады.

<P ALIGN= "орналастыру параметрiнiң мәнi">.

Орналастыру параметрi мәнiнiң орнына **"Left"** (сол), **"Right"** (оң), **"Center"** (орталық) және **"Justify"** (енi) сөздерiнiң бiрi жазылады. Web-бетке орналастырылған абзацтарға мысал келтiреміз:

```
<html>
<h2>Мәтiндi web-бетке орналастыру</h2>
<p align="left">Бұл жол беттiң сол жағына орналасқан.
<h5 align="right">Бұл жол беттiң оң жағына орналасқан.</h5>
<p align="center">Бұл жол беттiң ортасына орналасқан.
<h4 align="justify">Бұл абзац екi жақтан тураланған. Сен мәтiндi бетте орналастыру әдiстерi мен Microsoft Word мәтiн процессоры тақырыбында толық танысқансың.
HTML-да бұл iстердi орындау бiраз өзгеше жүзеге асырылады.
</html>
```

Мәтiндi web-бетке орналастыру

Бұл жол беттiң сол жағына орналасқан.

Бұл жол беттiң оң жағына орналасқан.

Бұл жол беттiң ортасына орналасқан.

Бұл абзац екi жақтан тураланған. Сен мәтiндi бетте орналастыру әдiстерi мен Microsoft Word мәтiн процессоры тақырыбында толық танысқансың. HTML-да бұл iстердi орындау бiраз өзгеше жүзеге асырылады.

Енгiзiлген мәтiн оның web-браузердегi көрiнiсiнен бiраз ерекшеленедi. Өйткенi, әдетте web-браузерлер артықша бос орындарды (пробелдердi) жоққа шығарады (тастап жiбередi). Кейде мәтiн қалай енгiзiлсе, web-бетте де солай орналастыруға тиic болады. Мысалы, web-бетке өлең орналастыру яки қарапайым белгiлер көмегiмен сурет сызу керек болса, мәтiн форматтауды web-браузерге тапсыруға болмайды. Мұндай жағдайда HTML-ның **<PRE>** жүп тегiн пайдаланады. Бұл тег арқылы мәтiн web-бетке, HTML-құжатта қалай жазылған болса, солай шығарылады. **<PRE>** тегiнiң қалай iстейтiнi мына мысалда түсiндiрiлген:

```

<html>
<pre>
Көктем!   Өте сұлу маусым!
  Көктем!   Өте сұлу маусым!
    Көктем!   Өте сұлу маусым!

</pre>
<p>
<br>Көктем!   Өте сұлу маусым!
<br>  Көктем!   Өте сұлу маусым!
      Көктем!   Өте сұлу маусым!

```

<P>,
 тегтерін <PRE> және </PRE> тегтері арасындағы мәтінге қатысты қолданудың пайдасы жоқ. Бұл аралықтағы осы тегтерді web-браузер қабылдамайды. Мәтінді Web-бетке <PRE> тегін пайдаланып, оп-оңай енгізуге болады, бірақ бұл тегті лажсыз жағдайда ғана істету жөн болады. Өйткені, web-браузер мәтінді деректер өрісінің өлшеміне бейімдеп форматтайды. Ал <PRE> тегін қолданатын мәтін web-браузер тарапынан форматталмайды, сондай-ақ оның деректер өрісіне сыймаған бөлігі көрінбей қалады.



Сұрақтар мен тапсырмалар

1. HTML-4 -те неше бағандық тақырыптарды қолдануға болады?
2. Тақырып бағандары бір-бірінен несімен ерекшеленеді?
3. HTML-дағы абзац қай тегтің жәрдемімен белгіленеді?
4. HTML-дың қайсы тегі мәтінді жаңа жолға өткізеді?
5. Мәтінді қалың, еңіс және асты сызылған шрифтпен жазуға мысалдар келтір.
6. Web-беттегі мәтінді туралау әдістеріне мысалдар келтір.
7. Мәтінді форматтамау үшін Web-браузер қандай тегті пайдаланады?

Жаттығулар

1. Республикамыз әнұранының мәтіні жазылған web-бет дайында. Онда жолдардың шрифт жазу формасын түрліше таңда.

2. Мектебің орналасқан республика, облыс, қала, аудан, көшенің аттарын сәйкесінше 1-, 2-, 3-, 4-, 5-бағандық тақырыптар көрінісінде бейнелейтін web-бет дайында.

3*. «Біздің сынып» атаулы web-бет дайында. Онда форматтаудың түрлі тегтерін пайдалан.

52-сабақ. Мәтінді Web-бетке енгізу тақырыбын қайталау

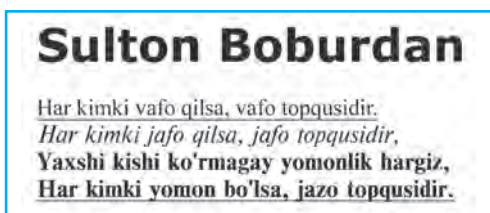
1. «Менің жанұям» атаулы web-бет дайында. Онда түрлі форматтау тегтерінен басқа белгілердің көмегімен жаратылған қарапайым сурет (үй, шырша және басқалар) да болсын.

2. Пәндердің аттарын қызығуыңа қарай түрлі бағандық тақырыптар түрінде бейнелейтін web-бет дайында.

3*. «Біздің сынып» деп аталып, дайындалған web-беттегі барлық сөйлемдер түгел қалың және еңіс, үзілмелі еңіс және асты сызылған түрде болсын.

4*. «Ұлы бабаларымыз» атаулы web-бет дайында. Онда беттің ортасында жайғасқан бабаларымыздың аттары 1-бағандық тақырыпта, ал атқарған еңбектері оң шетіне тураланып, 3-бағандық тақырыпта көрінсін.

5*. Көрінісі оң жақтағыға ұқсас, «Рубай» атаулы web-бет дайында.



53-сабақ. Шрифт өлшемі мен түсі, web-беттің фоны

Бұрынғы тақырыптарда шығарылған web-беттерде біз бірдей шрифт пайдаландық. Шрифт өлшемін де тақырып тегтерінің көмегімен өзгертуге болады. Бірақ Интернет web-беттерінде түрлі шрифтпен, түрлі өлшеммен, түрлі түстермен жазылған мәтіндерді кездестіруге болады.

Шрифт өлшемін белгілеу үшін HTML тілінде **** жұп тегі **SIZE** (өлшем) параметрімен бірге қолданылады. Осы тегті пайдаланғанда, енгізіліп жатқан мәтіннің шрифт өлшемі өзгермейді. Шрифт өлшемінің өзгергенін тек web-браузерде ғана көруге болады.

Web-бетте түрлі 7 өлшемі бар шрифтерді пайдалануға болады. Олар 1-ден 7-ге дейінгі параметр мәнімен белгіленіп, 1 – ең кіші, ал 7 – ең үлкен шрифт өлшемін білдіреді. Мысалы, шрифт өлшемін 5-ке тең етіп алу үшін, **** түріндегі жұп тегі жазылады. Негізгі шрифт өлшеміне оралу үшін, **** тегі пайдаланылады.

```

<html>
<h1>Шрифттиң өлшемі</h1>
<p>Мәтіннің бұл жолы негізгі шрифтпен жазылған<br>
<font size=7>7-шрифт<font size=6>6-шрифт<br>
<font size=5>5-шрифт<font size=4>4-шрифт</font>
</font></font></font>
<br>Мәтіннің бұл жолы негізгі шрифтпен жазылған<br>
<font size=3>3-шрифт</font>
<font size=2>2-шрифт<font size=1>1-шрифт
</font></font></font>
<br>Мәтіннің бұл жолы негізгі шрифтпен жазылған
</html>
    
```

Шрифттиң өлшемі

Мәтіннің бұл жолы негізгі шрифтпен жазылған

7-шрифт 6-шрифт

5-шрифт 4-шрифт

Мәтіннің бұл жолы негізгі шрифтпен жазылған

3-шрифт 2-шрифт 1-шрифт

Мәтіннің бұл жолы негізгі шрифтпен жазылған

HTML-код пен web-бетке назар аударсақ, шрифт өлшемі неше рет тандалған болса, сонша рет аяқталған болады.

Web-бетте пайдаланылған түрлі түстер оны одан да әдемі де көрікті етіп көрсетеді. Web-беттегі шрифт пен мәтінді таңдау үшін **** жұп тегі **COLOR** параметрімен бірге қолданылады. Бұл нұсқаудан кейін "=" белгісі және тырнақша ішінде **түс коды** жазылады. Түс коды "#" белгісімен басталады. Мәтіндегі оның жалпы көрінісі мынадай:

****.

Түс коды RGB (Red-қызыл, Green-жасыл, Blue-көк) түстер жүйесіне негізделген. Бұл жүйеде керекті түс негізгі үш түстің түрлі мөлшердегі қоспасынан жасалады. Негізгі үш түстің әрқайсысы 16-лық санақ жүйесіндегі 00 -ден FF -ке дейінгі (256) сандармен белгіленеді. Негізгі түстерді қолданған-

Ақ	#FFFFFF	White
Қара	#000000	Black
Қызыл	#FF0000	Red
Жасыл	#00FF00	Green
Көк	#0000FF	Blue
Сары	#FFFF00	Yellow
Қызғылт	#FF00FF	Magenta

да олардын ағылшын тилиндегі өрнегін де пайдалануға болады. Кестеде кейбір түстердің кодтары мен ағылшын тилиндегі бейнесі келтірілген.

Шрифт түсін таңдауды мына мысалдан көруге болады:

```
<html>
<font color="red"> Бұл</font>
<font color="blue"> web-беттің</font>
<font color="magenta"> барлық</font>
<font color="green"> сөздері</font>
<font color="yellow"> түрлі</font>
<font color="black"> түсті</font>
</html>
```

Бұл web-беттің
барлық сөздері
түрлі түсті

Web-беттегі мәтіннің я фонның түсін өзгерту үшін, **<BODY>** жұп тегі сәйкесінше **Text** (мәтін) яки **Bgcolor** (background color, яғни фон түсі) параметрлерімен бірге қолданылады. Бұл параметрлерден кейін "=" белгісі және тырнақша ішінде түс коды яки түстің ағылшын тилиндегі аты жазылады. Түс коды "#" белгісімен басталады. Ескерте кету керек, мәтін түсін өзгертетін тег, шрифт түсі басқаша етіп таңдалған бөлімге әсер етпейді. Төмендегі мысалда web-беттегі мәтінге және фонға түс беру көрсетілген:

Бұл {Бұл бөлім мәтін түсіне сәйкес қызғылт түсті болып көрінеді}
web-беттің барлық сөздері {Бұл бөлім мәтін түсіне сәйкес қызғылт түсті
болып көрінеді} түрлі түсті {Бұл бөлім мәтін түсіне сәйкес қызғылт түсті
болып көрінеді}

```
<html><body bgcolor="#ffddd0" text="#ff00ff">
<font color="red"> Бұл </font>{Бұл бөлім мәтін түсіне сәйкес қызғылт
түсті болып көрінеді}
<font color="blue"> web-беттің </font>
<font color="magenta"> барлық</font>
<font color="green"> сөздері </font>
{Бұл бөлім мәтін түсіне сәйкес қызғылт түсті болып көрінеді}
<font color="yellow"> түрлі</font>
<font color="black"> түсті</font>
{Бұл бөлім мәтін түсіне сәйкес қызғылт түсті болып көрінеді}
</body></html>
```

MS Word бағдарламасында фонға түс беру я фонға сурет орналастыру мүмкіншілігі бар еді. HTML тилинде фонға сурет орналастыру үшін **<BODY>** жұп тегі **Background** параметрімен бірге қолданылады. Бұл жағдайда теңдеу белгісінен кейін

суреттің толық адресі тырнақшасыз жазылады. Егер сурет web-құжаты бар каталогта болса, онда сол суреттің атын (мысалы, Gul.jpg) жазудың өзі жеткілікті.

```
<html>
<body background=Nastarin.jpg>
<body text="#ffffff">
<p>HTML тілінде фонға сурет орналастыру үшін BODY
<br>мен бірге BACKGROUND нұсқауы бірге қолданылады.
<br>Онда теңдік белгісінен соң суреттің толық мекенжайы
<br>тырнақшасыз жазылады.
</body></html>
```

HTML тілінде фонға сурет орналастыру үшін BODY мен бірге BACKGROUND нұсқауы бірге қолданылады. Онда теңдік белгісінен соң суреттің толық мекенжайы тырнақшасыз жазылады.

Әдетте, суретті Web-бетке тез жүктеу үшін, оның кенейтуі JPEG я GIF форматына сай келуі керек, бірақ форматы BMP суретті де пайдалануға болады. Бұл туралы кейінгі сабақтарда мәлімет береміз.



Сұрақтар мен тапсырмалар

1. HTML -да неше түс пайдалануға болады?
2. Web-беттегі шрифт өлшемі қай тегтің жәрдемімен белгіленеді?
3. Web-беттегі шрифт түсі қай тегтің жәрдемімен белгіленеді?
4. Web-беттегі мәтін түсі қай тегтің жәрдемімен белгіленеді?
5. <BODY> тегі параметрлерінің жұмысына мысалдар келтір.
6. Web-бет фонын түрлі түспен белгілеуді мысалдар көмегімен көрсет.
7. Суретті Web-бет фонына орналастыру туралы әңгіме айт.

Жаттығулар

1. «Ұлы бабаларымыз» атаулы web-беттегі әрбір сөз басқаларынан түсімен ерекшеленсін.
2. «Менің жанұям» атаулы web-беттегі сөздердің, мәтін мен фонның түсін өзгерт.
3. «Біздің сынып» атаулы web-беттің фонына сурет орналастыр.

54-сабақ. Шрифт өлшемі мен түсі, web-беттің фоны тақырыбын қайталау

1. Фамилияң, есімің мен әкеңнің аты-жөні web-бетте түрлі өлшемді шрифтпен көрінуі үшін Html-кодын жаз.

2. Ауданың, мектебің, сыныбың, фамилияң мен атың web-бетте түрлі түсті және өлшемді болып көрінуі үшін Html-кодын жаз.

3. Html-кодта әуелі web-бет фонын түске енгізіп, кейін сурет орналастыратын тег жаз. Шығарылған web-беттің көрінісін түсіндір.

4. «Отанымыздың тарихи қалалары» атты web-бет дайында. Онда қалалардың атаулары түрлі түс пен шрифт өлшемдері бойынша өзгеше болсын. Қалалар туралы жазған мәліметің абзацтан, ортасы бойынша тураланып шықсын.



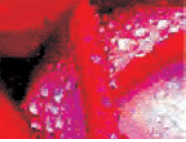
5. «Маған ұнаған мамандық» атты web-бет дайында. Web-бет фонына арналған, ұнаған мамандығыңа қатысты суретті MS Word коллекциясынан ал (бағыт: алдымен тиісті суретті MS Word жұмыс өрісіне орналастыр, кейін суреттің көшірмесін Paint бағдарламасының жұмыс өрісіне орналастыр, сурет бөлігін арнайы файлға сақта).

55-сабақ. Web-беттегі графика

Web-бетті «жандандырудың» ең тиімді әдісі – оған түрлі суреттер орналастыру. Тек мәтіннен тұратын web-беттің мәліметі мол болса, ол жалықтырып жібереді. Алайда web-бетте орналасқан шамадан тыс сурет web-бетті көріксіз етіп жіберіп, web-бет файлының көлемін үлкейтіп жібереді. Файл көлемі қаншама үлкен болса, оны Интернет желісінен оқып алу үшін соншама көп уақыт жұмсалады. Демек, web-бетке файл көлемі кіші болған суреттерді орналастыру жақсы екен. Интернет желісінде, негізінде, форматы **jpeg** және **gif** болған суреттер файлдары қолданылады. Мұның себебі сол, біріншіден, web-беттің көлемі кіші болады, екіншіден, web-браузер сол файлдарды қосымша бағдарламасыз көрсете алады.

Шынында, **bmp** форматы бар суреттер файлын **jpeg** форматына өткізсе, файл көлемі бірнеше есе азаяды. Суретті файлды бір форматтан екіншісіне өткізу үшін арнайы бағдарламалар (ACDSee, Photoshop,...) пайдаланылады. Салыстыру үшін төмендегі мысалды келтіреміз:

Суретті web-бетте орналастыру үшін **** жұп емес тегі (ағыл. image – бейне) қолданылады. Сурет файлын көрсету үшін

Суреттер	Форматы	Сурет өлшемі	Графикалық файлдың көлемі
	BMP	130 x 100 пиксель	38,3 килобайт
	JPEG	130 x 100 пиксель	4,44 килобайт
	GIF	130 x 100 пиксель	6,5 килобайт

аталмыш тегке **SRC** (source – дереккөз) параметрін қосу керек. Мысалы, web-бетке орналастырылатын суреттер файлының аты «mypic.jpg» болса, HTML-құжатқа мынадай жол қосылады:

****,

мұндағы **mypic.jpg** – SRC параметрінің мәні.

Алдыңғы тақырыптарда сен мәтінді web-бетте **ALIGN** параметрі көмегімен орналастыруға үйрендің. Аталмыш параметрді web-беттегі суретті сол я оң жаққа орналастыру үшін де қолдануға болады. Шынында бұл параметр мәтінді суреттің жанына түрлі жағдайда орналастыру үшін қолданылады, бірақ сурет орналасқан жолда мәтін болмаса, ол суреттің жайғасуына әсер етеді. Мысалы, **** немесе **** жазуы "mypic.jpg" суретті web-беттің оң жағына орналастырады.

Web-беттегі мәтін мен суретті **ALIGN** параметрінің төмендегі кестеде келтірілген мәндеріне тән жағдайдың бірінде орналастыру мүмкін.

Мысалы:

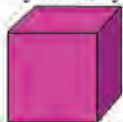
1. **** жазбасы web-бетке "kub.bmp" суретін мәтіннің ағымдағы жолының ең үлкен элементіне туралайды:

MIDDLE	Суреттің ортасы ағымдағы (суреттен кейінгі) жолдың астына тураланады
ABSMIDDLE	Суреттің ортасы ағымдағы жолдың ортасына тураланады
BOTTOM	Суреттің төменгі шекарасы ағымдағы жолдың астына тураланады
TOP	Суреттің жоғары шекарасы ағымдағы жолдың ең үлкен элементіне тураланады
LEFT	Сурет шет жағасына тиіп тұрады, мәтін суреттің оң жағына жазылады
RIGHT	Сурет оң жағасына тиіп тұрады, мәтін суреттің сол жағына жазылады

2. `` жазуы web-бетке "kub.bmp" суретін оң шетке тигізіп, суреттің жоғары шекарасын ағымдағы жолдың үлкен элементіне туралайды да, ағымдағы жол суреттің сол жағына жазылады:

```
<html>
Кубтың суреті<br>
 Кубтың 6 қабырғасы, 12 қыры, 8 ұшы
бар.
</html>
```

Кубтың суреті



Кубтың 6 қабырғасы, 12 қыры, 8 ұшы бар.

Ескерте кету керек, web-беттегі мәтін мен суреттің жайғасу күйі мәтін форматына тән тегтерге де байланысты, бұған көз жеткізу үшін жоғарыдағы мысалдардың екіншісінде `
` тегін алып тастаса жеткілікті.

```
<html>
Кубтың суреті<br>
 Кубтың 6 қабырғасы, 12 қыры, 8 ұшы
бар.
</html>
```

Кубтың суреті

Кубтың 6 қабырғасы, 12 қыры, 8 ұшы бар.



Суретті web-бетке орналастыруда оның өлшемдерін де таңдауға болады. Бұл үшін **WIDTH** (ені, кендігі) және **HEIGHT** (бойы, биіктігі) параметрлер қолданылады. Аталмыш нұсқаулар арқылы Web-бетке орналастырылып жатқан суреттің ені мен бойы пиксельдер яки суреттің негізгі өлшемі бойынша пайыз есебімен беріледі. Суреттің өлшемдерін пайыз есебімен беру қолайсыз (кейбір браузерлер пайызды қабылдамайды). Мысалы,

жазуы mypic.jpg файлындағы суреттің (жоғарғы кестедегі 130x100 пиксельді сурет) негізгі өлшемдері қандай болуына қарамастан, оны 50x100 пиксель өлшемді етіп орналастырады:

```
<html>
<p align="justify">
Web-беттегі сурет айналасында бос жер қалдыру (суреттен шегіну) үшін
HSPACE және VSPACE бұйрықтары қолданылады. Қалдырылған бос
жер пиксельдермен беріледі.
<br>HSPACE – суреттің
сол және оң жағынан бос жер қалдырады.
<br>VSPACE – суреттің төбесі мен астынан бос жер қалдырады.
</html>
```

Web-беттегі сурет айналасында бос жер қалдыру (суреттен шегіну) үшін HSPACE және VSPACE бұйрықтары қолданылады. Қалдырылған бос жер пиксельдермен беріледі.



HSPACE — суреттің сол және оң жағынан бос жер қалдырады.

VSPACE — суреттің төбесі мен астынан бос жер қалдырады.



Суреттің үлкейтілген өлшемдері оның сапасын нашарлатуға әкелетінін ұмытпа! Суретті бір графикалық редактор жәрдемімен керек өлшемге келтіріп алып, содан кейін web-бетке орналастырған дұрыс!

Суретті web-бетке орналастырғанда, оның терезесінде бос орын болмауы, яғни экрандағы суретке мәтін не басқа сурет «тиіп» қалуы мүмкін (жоғарыдағы 1-мысал). Web-беттегі сурет айналасында бос орын (бостық — space) қалдыру (суреттен шегіну) үшін **HSPACE** (суреттің сол және оң жағынан бос орын қалдыру) және **VSPACE** (суреттің төбесі мен астынан бос орын

қалдыру) нұсқаулары қолданылады. Қалдырылған бос орын пиксельдермен беріледі. Мысалы,

**<IMG SRC="lola.jpg" ALIGN=left
HSPACE=15 VSPACE=15>**

жазуы web-беттегі (лала.jpg) суреттің оң, сол, жоғары және төменгі жақтарынан кендігі 15 пиксельге тең (рамка тәрізді) бос орын қалдырады:

```
<html>
```

```
<p align="justify">
```

Web-беттегі сурет айналасында бос жер қалдыру (суреттен шегіну) үшін HSPACE және VSPACE бұйрықтары қолданылады. Қалдырылған бос жер пиксельдермен беріледі.

```
<br> HSPACE – суреттің сол және оң жағынан бос жер <br> қалдырады. <br>VSPACE – суреттің төбесі мен астынан бос жер <br>қалдырады.</html>
```

Web-беттегі сурет айналасында бос жер қалдыру (суреттен шегіну) үшін HSPACE және VSPACE бұйрықтары қолданылады. Қалдырылған бос жер пиксельдермен беріледі.



HSPACE – суреттің сол және оң жағынан бос жер қалдырады.

VSPACE – суреттің төбесі мен астынан бос жер қалдырады.

Сурет айналасында (қара) рамка шығару үшін BORDER (ше-кара) параметрі пайдаланылады. Мұнда параметр мәні ретінде рамка қалыңдығы пиксель есебімен алынады, мысалы:

.

Сурет айналасында орналасқан рамканы өз бетінмен жа-сап көр.



Сұрақтар мен тапсырмалар

1. Қандай графикалық форматтарды білесің?
2. BMP форматы бар суретті JPEG пен GIF форматына PAINT бағдарламасы жәрдемімен өткіз.
3. Суретті web-бетке қалай орналастыруға болады?
4. тегінің қандай параметрлері бар?
5. Суретті web-беттің сол не оң жағына қалай орналастыруға болады?
6. Web-беттегі суреттің өлшемдерін өзгерте отырып, оны қалай орналастыруға болады?

7. Web-беттегі суреттің айналасында бос орын қай параметрлер көмегімен қалдырылады?



Жаптығулар

1. «Ұлы бабаларымыз» атты web-бетке бабаларымыз табысқа жеткен салаға қатысты суреттерді түрлі өлшемдермен орналастыр.
2. «Маған ұнаған мамандық» атты web-бетке мамандыққа тиесілі суреттер орналастырып, суреттер маңында 15 пиксельдік рамка жаса.
3. «Менің жанұям» атты web-бетке жанұяң мүшелерінің мамандығы жөніндегі суреттерді орналастыр. Мұнда орналастырудың түрлі әдістерін пайдалан.

56–57-сабақ. Web-беттегі графика тақырыбын қайталау

1. «Біздің сынып» атаулы web-бетке 3 сурет орналастыр.
2. «Отанымыздың тарихи қалалары» атты web-бетке қалаларға қатысты суреттер орналастыр. Суреттер түрлі жағдайларда және өлшемдермен орналасқан болсын (бағыт: суреттерді Интернеттен шығаруға болады).
3. «Үй жануарлары» атты web-бет дайында. Онда жануарлар суреттері орналасқан және сурет жанында жануардың аты жазылған болсын (бағыт: суреттерді MS Word коллекциясынан алуға болады).
4. «Компьютер құрылғылары» атты web-бет дайында. Оған қажетті суреттерді графикалық редактордың көмегімен өзің жаса.
5. «Менің достарым» атты web-бет дайында. Онда достарыңды қызықтырып жүрген мамандыққа я салаға тиісті сурет пен мәлімет орналастыр.

58-сабақ. Web-бетке тізім орналастыру

MS Word мәтін процессоры жәрдемімен дайындалған құжаттың тізім жасаудың 2 әдісі бар, солардың бірі –  түймесінің жәрдемімен) маркерленген тізім, екіншісі ( түймесінің көмегімен) реттелген тізім, мысалы:

Маркерленген тізім	Маркерленген тізім	Маркерленген тізім	Реттелген тізім	Реттелген тізім
•Информатика	○ Информатика	◆Информатика	1. Информатика	A. Информатика
•Математика	○Математика	◆Математика	2. Математика	B. Математика
•Тарих	○Тарих	◆Тарих	3. Тарих	C. Тарих

Тізімді web-бетке енгізу үшін (unordered list – реттелмеген, яғни маркерленген тізім) я (ordered list – рет-

телген тізім) жұп тегтері пайдаланылады. Осыдан көрінгеніндей, HTML-құжатта және тегтерінен кейін орналасқан деректерді web-браузер оң жаққа белгілі бір қашыққа шегініп бейнелейді. Тізім элементтерін белгілеу үшін (list item – тізім элементі) жұп емес тегі қолданылады. Тізімнің тегімен басталған элементі үнемі жаңа жолда көрінеді. Мысалы, кестедегі бірінші бағанда маркерленген тізімді бейнелейтін HTML-құжаттың үзіндісі былай жазылады:

```
<html><ul>
<li>Информатика<li> Математика<li> Тарих
</ul></html>
```

Мұнда, әрине, мына сұрақ туындайды: маркер көрінісін қалай өзгертуге болады?

Маркерленген тізімді белгілейтін тегін **TYPE** параметрімен бірге жазуға болады. Бұл параметр мәндері **disk** (дөңгелек), **circle** (шеңбер), **square** (боялған квадрат) болады, мысалы, <UL TYPE = **square**>. Егер тегі, жоғарыда көрсетілгеніндей, параметрсіз жазылса, Web-браузер тізім маркерін диск мәні деп есептейді. Тізім элементін көрсетуші тегін де TYPE параметрімен жазуға болады:

```
<html><ul>
<li type=disk> информатика
<li type=circle> Математика
<li type=square> Тарих
</ul></html>
```

- Информатика
- Математика
- Тарих

Кестедегі реттелген тізімді web-бетте көрсету керек болса, HTML-құжаттың үзіндісі мен web-браузерде көрінісі мынадай болады:

```
<html><ol>
<li> информатика<li> Математика<li> Тарих
</ol></html>
```

1. Информатика
2. Математика
3. Тарих

Егер тізімнің реттік нөмірі 1-ден басқа саннан (мысалы, 3-тен) басталуы керек болса, тегі **START** параметрімен бірге істетіледі, мысалы: <OL start = 3>. Егер тізімді латын әріптері яки рим цифрлары арқылы ретке келтіру керек болса,

онда тегі TYPE параметрімен істетіледі. Соған ұқсас, тегін де TYPE және VALUE параметрлерімен бірге қолдануға болады.

Төмендегі мысал түрліше ретке келтірілген тізімнің web-бетте қалай орналасатынын айқын көрсетеді:

```
<html>
<ol start=3><li> Информатика<li> Математика<br>...
<li value=17> Тарих</ol>
<ol type=A><li> Информатика<li> Математика</ol>
<ol type=a><li> Информатика<li> Математика</ol>
<ol type=I start=5><li> Информатика <li> Математика</ol>
<ol type=i><li> Информатика<li type=A> Математика</ol>
</html>
```

Кейде, web-беттердің көрінісі көрікті де әсем болуы үшін, маркерленген тізімде маркер ретінде графикалық бейнелердің пайдаланылғанын байқауға болады. Бірақ, web-беттің графикалық элементтерінің бар болуы жіберілетін мәлімет көлемін асырып жібереді. Суреттік тізімді ұйымдастыруда тегінің қажеті болмайды. Суреттік тізім элементтерін бір-бірінен ажырату үшін <P> я
 тегтерін пайдалануға болады. Графикалық маркерленген тізімнің қалай жасалатынын мына мысалдан көруге болады:

```
<html><ul>
<h2>Ағаштар</h2><br>
<h3> Шырша <br>
 Тал<br>
 Шие</h3>
</ul></html>
```

Ағаштар



Шырша

Тал

Шие

Ескерте кету керек, бұл жерде «Barglar.gif» – графикалық файл, ол осы web-бет сақталған каталогта орналасқан болады.

Интернет желісінде өте көп үйретуші web-сайттар бар, солардың жәрдемімен белгілі бір пәнді, пән бөлімін немесе бағдарлама құралдарын пайдаланып үйренуге болады. Бәріне белгілі, әрбір пән, әр бағдарламалық құрал қажетті атаулар мен ұғымдар жүйесін қамтиды. Бұл атаулар мен ұғымдарды web-бетте сипаттау үшін <DL> (definition list – сипаттау тізімі) жұп тегі қолданылады. Осы жұп тегінің ішінде <DT> (definition term – атауды сипаттау) және <DD> (definition description – сипаттау нәтижесі) жұп емес тегтер қолданылады. Төмендегі мысалдан осы тегтердің міндетін оңай түсінуге болады:

```
<html><dl><dt>Информатика
<dd>Компьютер техникасын қолдана отырып, адам
қызметінің түрлі салаларындағы ақпараттарды іздеу, жинау,
сақтау, қайта өңдеу, пайдалану мәселелерімен
шұғылданатын ғылым.
<dt>Дескриптор яки<dd> HTML тілінің бұйрықтары "<" жаңа ">"
белгілері арасына жазылады, ағылшын тілінен тег – жарлық,
белгі, ерекше деп аударылады.
</dl></html>
```

Информатика

Компьютер техникасын қолдана отырып, адам қызметінің түрлі салаларындағы ақпараттарды іздеу, жинау, сақтау, қайта өңдеу, пайдалану мәселелерімен шұғылданатын ғылым.

Дескриптор яки

HTML тілінің бұйрықтары "<" жаңа ">" белгілері арасына жазылады, ағылшын тілінен тег – жарлық, белгі, ерекше деп аударылады.

Кейде сондай жағдайлар болады, тізімнің кейбір элементіне тізім қосу керек болады. Онда бірінің ішіне екіншісі орналасқан, яғни қабаттасқан тізім құрастырылады. Төмендегі мысалда саған белгілі болған тегтер көмегімен қабаттасқан тізім жасау көрсетілген:

```
<html><ul>
<b>Кейбір планеталардың серіктері </b>
<li> Жер
<ol><li> Ай </ol>
<li> Марс
<ol><li> Фобос <li> Деймос </ol>
</ul></html>
```

Кейбір планеталардың серіктері

- Жер
 1. Ай
- Марс
 1. Фобос
 2. Деймос

Мұндай тізімдерді ұйымдастыру қиын емес, бірақ тегтердің тәртібін жазуға ерекше назар аударуға тура келеді.



Сұрақтар мен тапсырмалар

1. MS Word-тың қандай тізім түрлері бар?
2. Web-бетте неше түрлі тізім пайдалануға болады? Мысалдар келтір.
3. Реттік нөмірі бар тізім қалай ұйымдастырылады? Мысалдар келтір.
4. Маркерленген тізім қалай ұйымдастырылады? Мысалдар келтір.
5. Бейне-графикалық тізімді ұйымдастыру процесін түсіндір.
6. Графикалық маркерленген тізім қалай ұйымдастырылады?
7. Сипаттамалар тізімі туралы әңгіме айт.

Жагыгулар

1. Рим цифрлары қатысқан тізім жаса.
2. Реттік нөмірі 9-дан басталған тізім жаса.
3. Қабаттасқан тізім жасап, оны түсіндір.

59-сабақ. Web-бетке кесте орналастыру

Кестені web-бетке енгізу үшін **<TABLE>** (table – кесте), **<TR>** (table row – кесте жолы), **<TH>** (table header – кестедегі тақырып), **<TD>** (table data – кесте мәліметтері) жұп тегтері қолданылады. **<TABLE>** тегі – кестенің басын, **</TABLE>** тегі – кестенің аяғын, **<TR>** жұп тегі – кесте жолын, **<TD>** жұп тегі – кесте бағанын білдіреді. **<TH>** жұп тегі тақырып торларын білдіріп, сол торлардағы деректер web-бетте қара шрифтпен тор ортасына тураланған күйінде (яғни, **ALIGN=Center** және **VALIGN=Middle**, мұндағы **V** – вертикаль бағытты білдіреді) бейнеленеді. **<TH>** тегі **<TR>** тегінсіз қолданылмайды. Егер web-беттегі кестенің шекара сызықтары да көрінуге тиісті болса, онда **BORDER** параметрі қолданылады. **BORDER** параметрінің мәндері пиксельдермен өлшеніп, кесте рамкасының сыртқы сызықтарының қалыңдығына ғана өзгеше әсер етеді.

Ай аты	Маусым	Айдың реттік нөмірі
Январь	Қыстың 2-айы	Жылдың бірінші айы
Декабрь	Қыстың 1-айы	Жылдың соңғы айы

Кестені жолдар түзеді. Әрбір жол өз кезегінде бағандарға бөлінеді. **<TR>** және **<TD>** тегтері сәйкесінше **</TR>** және **</TD>** тегтерімен жабылады, мысалы, кестені бейнелейтін HTML-құжаттың көрінісі былай жазылады:

```

<TABLE>
<TABLE BORDER>
<TR> <TH> Ай аты </TH><TH> Маусым</TH><TH> Ай
төргібі </TH></TR>
<TR> <TD>Январь</TD><TD>қыстың 2-айы</TD><TD>
Жылдың бірінші айы</TD> </TR>
<TR> <TD>Декабрь</TD><TD>қыстың 1-айы</TD><TD>
Жылдың соңғы айы</TD> </TR>
</TABLE>
    
```


Кестенің шекара сызықтарынсыз және шекара сызықтарымен бейнеленген HTML-құжаттың және оған тән web-беттегі кестенің орналасуын мына мысалдан көруге болады:

```
<HTML><TABLE>
<TR><TH>Ай аты</TH><TH>Маусым</TH><TH>Ай тәртібі
</TH></TR><TR><TD>Январь</TD><TD>Қыстың 2-айы</TD>
<TD>Жылдың бірінші айы</TD></TR>
<TR><TD>Декабрь</TD><TD>Қыстың 1-айы</TD>
<TD>Жылдың соңғы айы</TD></TR></TABLE>
<TABLE BORDER=7>
<TR><TH>Ай аты</TH><TH>Маусым</TH>
<TH>Ай тәртібі </TH></TR><TR><TD>Январь</TD>
<TD>Қыстың 2-айы</TD><TD>Жылдың бірінші айы</TD></TR>
<TR><TD>Декабрь</TD><TD>Қыстың 1-айы</TD >
<TD>Жылдың соңғы айы</TD></TR>
</TABLE></HTML>
```

Ай аты	Маусым	Ай тәртібі
Январь	Қыстың 2-айы	Жылдың бірінші айы
Декабрь	Қыстың 1-айы	Жылдың соңғы айы

Ай аты	Маусым	Ай тәртібі
Январь	Қыстың 2-айы	Жылдың бірінші айы
Декабрь	Қыстың 1-айы	Жылдың соңғы айы

Web-бетте бейнеленетін кестеге тақырып беру керек болса, **<CAPTION>** (тақырып) жұп тегі қолданылып, оны бірінші **<TR>** тегінен бұрын жазу керек болады. Бұл тегтің **ALIGN** және **VALIGN** параметрлері бар болып, оларды қалай пайдалануға болатыны төмендегі кестеде көрсетілген:

ALIGN	VALIGN	Түсіндіру
TOP	жазылмайды	Тақырып кестеден жоғары кестенің ортасы бойынша тураланады
BOTTOM	жазылмайды	Тақырып кесте астында кестенің ортасы бойынша тураланады
LEFT	TOP	Тақырып кестеден жоғары кестенің сол жақ шекарасына тураланады

LEFT	BOTTOM	Тақырып кесте астында кестенің сол жақ шекарасына тураланады
CENTER	TOP	Тақырып кестеден жоғары кестенің ортасы бойынша тураланады
CENTER	BOTTOM	Тақырып кесте астында кестенің ортасы бойынша тураланады
RIGHT	TOP	Тақырып кестеден жоғары кестенің оң жақ шекарасына тураланады
RIGHT	BOTTOM	Тақырып кесте астында кестенің оң жақ шекарасына тураланады

Мына мысалда ALIGN және VALIGN параметрлерінің істетілуін көруге болады:

```
<html><table><table border=7>
<caption align=right valign=bottom>КЕСТЕ</caption>
<tr><th>Ай аты</th><th>Маусым</th><th>Ай тәртібі </th></tr>
<tr><td>Январь</td><td>Қыстың 2-айы</td><td>Жылдың бірінші айы</td></tr>
<tr><td>Декабрь</td><td>Қыстың 1-айы</td><td>Жылдың соңғы айы</td></tr>
</table></html>
```

Ай аты	Маусым	Ай тәртібі
Январь	Қыстың 2-айы	Жылдың бірінші айы
Декабрь	Қыстың 1-айы	Жылдың соңғы айы

КЕСТЕ

Кестенің ішкі сызықтарының қалыңдығын (екі торды айыратын параллель сызықтар арақашықтығын) **CELLSPACING** параметрімен (мысалы, **CELLSPACING =5**) пиксельдер есебінде таңдауға болады. Ал торға енгізілген деректер мен тордың арақашықтығы **CELLPADDING** параметрімен (мысалы, **CELLPADDING=9**) пиксельдермен анықталады.

Біз бұрын, суретгі web-бетке орналастырғанда, оның айнала-сындағы мәтіннің орналасуын айтып өткенбіз. Сол сияқты, мәтінді кестенің де айналасына орналастыру мүмкіншілігі бар, бірақ мәтін кестенің тек сол не оң жағына орналастырылады.

Бұл үшін <TABLE> тегі ALIGN параметрімен бірге істетіледі, мысалы: <TABLE ALIGN=LEFT>. Егер кесте web-беттің сол жағына тураланса, мәтін кестенің оң жағында көрінеді және керісінше болады.

Ескерте кету керек, тор ішіндегі деректерді форматтау үшін, тақырыптарды, мәтіндер мен суреттерді форматтауда қолданылатын барша тегтерді пайдалануға болады. Кестелер, жолдар мен торлардың шекара сызықтарының түсін өзгерту BORDERCOLOR параметрі арқылы іске асырылады.

Жоғарыда келтірілген мәліметтерге сүйеніп, төмендегі HTML-құжат пен web-бетті жасауға болады:

```
<html>
<table><table align = right Border=7 bordercolor=RED
CELLSPACING=5 CELLPADDING=3>
<TR bordercolor=BLUE> <TH>Ай аты </TH><TH>Маусым </TH>
<TH>Ай тәртібі </TH></TR>
<TR><TD>Январь</TD><TD bordercolor=MAGENTA>Қыстың 2-айы
</TD><TD>Жылдың бірінші айы</TD>
<TR bordercolor=black><TD>Декабрь</TD><TD>Қыстың 1-айы
</TD><TD bordercolor=green>Жылдың соңғы айы</TD></TR>
</table>
<BR>Ескерте кету керек, тор ішіндегі мәліметтерді форматтау үшін,
тақырып, мәтін мен суреттерді форматтауда істетілген барша
тегтерді пайдалануға болады. Кесте, жол мен торлар шекара
сызықтарының түсін өзгерту BORDERCOLOR параметрі арқылы
жүзеге асырылды.
</html>
```

Ескерте кету керек, тор ішіндегі мәліметтерді форматтау үшін, тақырып, мәтін мен суреттерді форматтауда істетілген барша тегтерді пайдалануға болады. Кесте, жол мен торлар шекара сызықтарының түсін өзгерту BORDERCOLOR параметрі арқылы жүзеге асырылды.

Ай аты	Маусым	Ай тәртібі
Январь	Қыстың 2-айы	Жылдың бірінші айы
Декабрь	Қыстың 1-айы	Жылдың соңғы айы

MS Word бағдарламасында құжатқа орналастырылған кестенің торларын біріктіруге немесе торды қайтадан торларға бөлуге болатын еді.

1-тор	2 баған біріккен тор	
2 жол біріккен тор	4-тор	2 жол біріккен тор
	6-тор	

HTML тілі де сондай мүмкіншілік береді. Бұл үшін <TH> я <TD> тегтері **COLSPAN** (column spanning – бағанды орналастыру) және **ROWSPAN** (row spanning – жолды біріктіру) параметрлерімен бірге істетіледі. Логика тұрғысынан қарағанда, бөлінген торы бар кестені жасау үшін, тор бөлудің орнына, басқа торларды біріктіру арқылы мақсатқа жетуге болады. Жоғарыдағы кестенің HTML-құжаты мынадай болады:

```
<HTML><TABLE><TABLE BORDER CELLPADDING=2><TR>
<TD align=middle><font color=red>1-тор</font></TD>
<TD COLSPAN=2 bgcolor=#00ffD0><B><font color=white>2 баған біріккен
тор</font></B></TD>
<TR><TD ROWSPAN=2 bgcolor=yellow><B><I><U>2 жол біріккен тор
</U></I></B></TD>
<TD bg color=magenta><U>4-тор</U></TD>
<TD ROWSPAN=2 bgcolor=black><B><font color=white>2 жол біріккен
тор</font></B></TD></TR><TR><TD
bgcolor=#808080><I>6-тор</I></TD>
</TABLE></HTML>
```

Кестелермен әлі де бірнеше амалдарды орындауға болады, біз саған оларды өз бетіңмен үйренуді ұсынамыз.



Сұрақтар мен тапсырмалар

1. Web-беттегі кесте қалай ұйымдастырылады?
2. Кестенің шекара сызықтарын қалай форматтауға болады?
3. Кестедегі тақырып туралы мәлімет бер.
4. Кесте тақырыбы туралы мәлімет бер.
5. Кесте мен мәтін web-бетте қалай орналастырылады?
6. Кесте ішіндегі деректерді форматтауға қатысты мысал келтір.
7. Кестенің шекара сызықтарын форматтауға қатысты мысал келтір.

Жаттығулар

1. Кестеге тиесілі тегтері мен оларды сипаттайтын бағаны бар web-бет жаса.
2. Кестені «Біздің сынып» атты web-бетке орналастыр.
3. «Менің жанұям» атты web-беттегі жанұяң мүшелері туралы мәліметтерді кестелік түрде бейнеле.

60-сабақ. Web-бетке тізім мен кесте орналастыру тақырыбын қайталау

1. Мектеппен байланысты маркерленген және реттелген тізімдерді камтитын қабаттасқан тізім жаса.

2. «Компьютер құрылғылары» атты web-беттегі құрылғыларды кестеге маркерленген тізім арқылы орналастыр.

3. «Менің сүйікті әдеби кейіпкерім» атты web-бет дайында. Онда шығарманың аты тақырып болып, шығарманың авторы мен әдеби кейіпкерінің қасиеттері кестелік түрде бейнеленсін.

4. Үздік оқитын 5 сыныптасыңның фамилиясын «Біздің сынып» атты web-бетке сынып журналындағы реттік нөмірмен орналастыр.

5. «Маған ұнаған мамандық» атты web-беттегі мамандығыңа байланысты салаларды кестенің ішіндегі тізімге ұқсатып бейнеле.

61-сабақ. Web-беттегі «өту» (гипержүгіну)

Web-беттегі ақпарат, бірнеше бөлімнен тұруы мүмкін. Бұл бөлімдерге тез «өту» мүмкіншілігі web-бетті оңай қарап шығуға жағдай жасайды. Web-беттегі осындай өтулер гипермәтіндік байланысты құрайды. Гипермәтіндік технология өз кезінде WWW қызметінің қысқа уақыт ішінде көпшілікке ортақ болып қалуына себеп болған еді.

Гипержүгіну, яғни құжаттың бір жерінен екінші жеріне, әлде бір құжаттан екінші құжатқа өту, <A> жұп тегінің жәрдемімен іске асырылады. Бұл тегтің **HREF** параметрі бар, оның мәні – өту орнының мекенжайы (адресі). Web-беттің тап осы тег жазылған орны **өту нүктесі** деп аталады. <A> тегі мен оның параметрі жалпы жағдайда былай жазылады:

** мәтін ,**

бұл жерде «мәтін» — еркін мәтін, браузер оны экранға белгілеп көрсетеді, «адрес» — өтуге керек болған орын (бөлім) мекенжайы. Әдетте өту орнын анықтайтын мәтін экранға көк түсті және асты сызылған көріністе шығады. Мекенжай да еркін мәтін болуы мүмкін.

Web-беттің өту керек болған орнында <A> тегінің **NAME** параметрінің көмегімен «мекенжай» енгізіледі. Ол өту нүктесіндегі «мекенжаймен» бірдей болуы керек. Бұл тег **NAME** параметрімен бірге жалпы күйде былайша жазылады:

 мәтін ,

Мұндағы «Мәтін» – **еркін** мәтін әдетте, мәтін ретінде web-беттің осы жерінен басталған бөлім аты жазылады; <A NAME> -дегі «адрес» <A HREF> -тегі адресстен «#» белгісімен ерекшеленеді.

```
<HTML>
<H2 ALIGN="center">Web-бетке гипержүгіну</H2>
<UL>
<LI><A HREF="#1-glava">I TAPAY</A>
<LI><A HREF="#2-glava">II TAPAY</A>
<LI><A HREF="#3-glava">III TAPAY</A>
</UL>
<P><H2><A NAME="1-glava">I TAPAY</A></H2>
<P>Бұл жерде I-TAPAY-ға тиісті ақпарат жазылады
<P><H2><A NAME="2-glava">II TAPAY</A></H2>
<P>Бұл жерде II-TAPAY-ға тиісті ақпарат жазылады
<P><H2><A NAME="3-glava">III TAPAY</A></H2>
<P>Бұл жерде III-TAPAY-ға тиісті ақпарат жазылады
</HTML>
```

Web-беттің өту нүктесі басқа мәтіндерден «ажыралып» тұруы үшін оны тізім құрамына енгізуге болады. Гипержүгіну web-беттің көшу нүктесінде берілген мәтінге жүгірткі параметрін әкеліп, жүгірткінің сол жақтағы түймесін басу арқылы орындалады. Бұл жағдайда көшу нүктесіндегі мәтіннің түсі өзгереді (әдетте қызғылт түсті болып көрінеді).

Келтірілген мысалда көшу нүктелері маркерленген тізім құрамына енгізілген. Өту нүктесімен өту орны экранда көрініп тұрған болса, өтудің орындалғаны білінбейді,

Гипержүгіну көшу нүктесіндегі мәтін орнына немесе мәтінмен бірге суретте қоюға болады. Бұл үшін өту нүктесіндегі мәтін орнына тегі істетіледі. Аталмыш тег көмегімен қойылған суретті браузер автоматты түрде көшу нұсқауымен байланыстырады. Енді web-беттің ішінде «өту» үшін мәтін де, сурет те пайдалануға болады Мысалы, HTML-құжатқа

жазбасы енгізілсе, web-бетте «mypic.jpg» сурет арқылы «1-тарауға» өтуге болады.

<A> тегі көмегімен бір web-бет ішінде ғана емес, web-беттен басқа web-бетке көшуді де жүзеге асыруға болады. Бұл үшін тап осы тегтің «адресі» ретінде Интернет жүйесіндегі бірер web-бет мекенжайды, яғни URL-мекенжайды жазса жеткілікті. Мысалы,

<A HREF="http://www.rambler.ru"//Rambler.ru ға көшу

Көшудегі «адрес» ретінде дискідегі web-бет файлының атын жазса да болады, сонда дискідегі web-бет ашылады. Бұл қасиет **құрылымдық web-беттер (web-сайттар)** жасауға мүмкіндік береді.



Құрылымдық web-бет — бір тақырыпқа арналған, бір-бірімен байланысты және бір-біріне «өту» мүмкіндігі болған web-беттер.

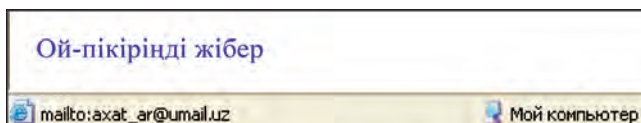
Бір тақырыпты бөліктерге бөліп, әрбір бөлім үшін арнайы web-бет дайындау, олардың әрқайсысында бір-біріне көшуді ұйымдастыру, артықша жұмыс болып көрінуі мүмкін. Бірақ бұның төмендегідей (негізгі) артықшылықтары бар:

- Web-беттегі ақпарат көлемі қанша аз болса, оны редакциялау соншалықты оңай болады;
- Web-беттегі ақпарат көлемі қаншалықты аз болса, оны оқу (қарап шығу) соншалықты қолайлы;
- Web-бет файлының көлемі қаншалықты кіші болса, оны Интернет тармағынан «алу» соншалықты жедел іске асырылады;

<A> тегі көмегімен электронды поштаға да мәлімет жіберуді ұйымдастыру мүмкін. Бұл үшін пошта адресінен алдын **mailto** (поштаға) сөзі жазылады, мысалы:

Пікір-ойларыңды жібер.

Осы гипержүгіну басқалардан ерекшеленбейтіндей болып көрінеді. Егер жүгірткі көрсеткішін гипержүгінуге бағыттаса, күй жолында электронды поштаның адресі шығады:



Бірақ, бұл адреспен қатынаста болғанда (Интернет тармағында істеп жатқан болсаң) web-браузер электронды поштамен

истеу үшін өз терезесін ашады. Бұл терезенің интерфейсі түрлі web-браузерлерде түрліше болуы мүмкін. Бірақ көп бағдарламалар адрес пен пайдаланушының аты туралы хабар шығарады да, «Субъект» жолын толтыруды ескертеді.



Сұрақтар мен тапсырмалар

1. URL-адрес дегенде нені түсінесің?
2. Web-бетте гипержүгіну қалай ұйымдастырылады?
3. Web-беттегі өту нүктесі нелерді қамтитынын түсіндір.
4. Құрылымдық web-бет дегенде нені түсінесің?
5. Электронды поштаға жүгіну қалай ұйымдастырылады?

Жаттығулар

1. «Менің жанұям» атты web-бетте жанұя мүшелерінің тізімі арқылы олар туралы мәліметтерге көшу ұйымдастырылсын.
2. «Үй жануарлары» атты web-бетте үй жануарының суреттері арқылы олар туралы мәліметтерге көшу ұйымдастырылсын.
3. «Менің web-бетім» атты web-бет дайында, одан өзің дайындаған web-беттерге және тағы негізгі бетке оралатын гипержүгіну ұйымдастыр.

62-сабақ. Пішіндер

Web-беттегі пішіндер (формалар) сұрақ-жауап өткізу, web-сервердің клиентпен диалог жүргізу яки берілген тізімнен керекті құжатты таңдау мақсатымен қолданылады. Мысалы, web-беттің рейтингін анықтау, бірер кәсіпорынның өнімдері туралы ой-пікір жинау, Интернет арқылы танысу мақсатында web-бетте формалар ұйымдастырылады. Мақсатқа қарай формадағы сұрақтар да түрліше болады. Бірақ, web-беттегі сұрақтарды бірдей тегтер көмегімен ұйымдастыруға болады. Бұл тегтер web-бетте сұрақ мәтінімен бірге жауап терезесін де жасайды. Сұрақ мәтіні web-бет дайындалып жатқанда енгізіледі. Ол құрылысына қарай негізгі екі түрге бөлінеді:

1. Еркін жауап енгізуге арналған.
2. Ұсынылған жауаптардың біреуін таңдауға арналған.

Сұрақтарды web-беттің қалаған жеріне орналастыруға болады, яғни сен пішін құрылымын өзіңе ұнаған пішінде құра аласың.

Диалогты басқа жолмен де ұйымдастыруға да болады. Мысалы, бұрын қарастырылғанындай, web-бетте керекті сұрақтар-

ды (қарапайым мәтін ретінде) теріп, бұл сұрақтарға электронды пошта арқылы жауап беруді өтініш етіп, электронды поштаның адресін көрсете аласың. Бірақ бұл жағдайда көп (дерлік) жауап ала алмайсың. Өйткені, біріншіден, бейтаныс кісіге әркім хат жаза бермейді. Екіншіден, формадағы сұрақтарға жауап беру көп түсінік талап етпейді, бірақ хатты ойлап жазу керек болады.

Форма жасау үшін **<FORM>** жұп тегі қолданылады. Оның **ACTION** және **METHOD** сияқты параметрлері бар. **ACTION** – параметр, оның мәні – **URL**-адрес.

Форманы жөнелту бірнеше әдіспен жүзеге асырылады. Форманы жөнелту әдісін көрсету үшін **METHOD** параметрі қолданылады. Көп жағдайда форманы жөнелту үшін электронды пошта көбірек қолданылады. Бұл үшін **<FORM>** тегіне **METHOD=POST** және **ACTION= "mailto: электронды пошта жүйесіндегі адрес"** параметрлерінің мәнімен қосылады. Мысалы:

**<FORM METHOD=POST
ACTION="mailto:rtm@umail.uz">**

Формада сұрақ-жауап ұйымдастыру үшін **<INPUT>** тақ тегі **NAME** параметрімен қолданылады. Бұл тег көмегімен берілген сұраққа жауап енгізу үшін мәтін алаңы (жауап жолы) жасалады. Жауап жолы белгілерінің саны **SIZE** (өлшем) параметрімен анықталуы мүмкін. Ал сұрақ қарапайым мәтін ретінде енгізіледі. Мысалы:

**<P> Сенің есімің:
<INPUT NAME = "есім енгізілетін орын" SIZE=25>**

Бұл тегтер формада «Сенің есімің:» мәтінін және 25-ке дейін белгі енгізуге болатын «есім енгізілетін орын» атты мәтін алаңын шығарады.

Форманың кейбір жеріне жазылатын жауап бір жолға сыймай қалуы мүмкін. Мысалы, формада «Түсіндірме» тарауы болса, әдетте бұл тарауға бірнеше жолдан тұратын мәтін жазылатын алаң ажыратылады. Бұл үшін **<TEXTAREA>** қос тегі пайдаланылады. Бұл тег құрамында мәтін алаңының атын (**NAME**), жолдар (**ROWS**) және бағандар (**COLS**) санын белгілейтін параметрлер іске қатысады. Мысалы,

```
<P> Түсіндірме:
<TEXTAREA NAME="Түсіндірме" ROWS=4 COLS=40>
</TEXTAREA>
```

Бұл тегтер формада «Түсіндірме:» сөзі, сондай-ақ 4 жол және 40 бағанды (яғни әрқайсысы 40-тан символдық 4 жол) «Түсіндірме» атты мәтін өрісін жасайды.

Сондай сұрақтар да кездеседі, оларға анық жауаптардың біреуі таңдалады. Мысалы, білімнің туралы сұраққа сен әрине, «бастауыш», «орта», «арнаулы білім» немесе «жоғары білім» жауаптарының бірін таңдайсың. Бұндай сұрақтарға жауаптарды формаға алдын ала енгізіп қоюға болады. Әдетте, осылай ұйымдастырылған жауаптардың алдына шеңбер орналастырып, қайсы жауап таңдалса, сол жауаптың алдындағы шеңбер (жүгірткінің көмегімен) белгіленеді. Формада осындай сұрақ-жауапты ұйымдастыру үшін <INPUT> тегінде NAME параметрі мен TYPE (тип) параметрі RADIO (жөнелт) мәнімен бірге кіреді. Белгіленген шеңберге сәйкес саған (я web-серверге) келетін мәлімет VALUE (мән) параметрінің мәні болады. Мысалы:

```
<P>Мәліметің:<BR>
```

```
<INPUT TYPE = radio NAME = "Білімі" value =
«Бастауыш">Бастауыш<BR>
```

```
<INPUT TYPE = radio NAME = "Білімі" value = "Орта">
Орта <BR>
```

```
<INPUT TYPE=radio NAME="Білімі" value="арнаулы білім">
Орта арнаулы <BR>
```

```
<INPUT TYPE=radio NAME="Білімі" value="Жоғары білім">
Жоғары
```

Бұл жерде TYPE=radio параметрі экранда шеңбер жасайды; NAME= -нен кейінгі тырнақша ішінде жазылған «Білім» сөзі алаңның аты болып, экранға шығарылмайды; VALUE= ден кейінгі «Бастауыш» сөзі web-серверге жіберіледі; одан кейінгі «Бастауыш» сөзі экранға шеңбер жанынан шығарылады.
 тегі кейінгі мәтіннің мәні болады.

Кейде ұсынылған жауаптардың бірнешеуін таңдау қажет болып қалады. Сол кезде TYPE параметрінің RADIO мәні орнына **CHECKBOX** (таңдау орны) мәні істетіледі. Бұл жағдайда формада шеңбер орнына тор пайда болады. Мысалы, менгеріп алын-

ған тілдер туралы сұрақты HTML-құжатта былайша өрнектеу мүмкін:

```
<P>Сөйлесе алатын тілдерің<BR>
```

```
<INPUT TYPE = checkbox NAME = "Тіл" value=
"Uzbekish">Өзбек<BR>
```

```
<INPUT TYPE=checkbox NAME="Тіл" value="Russian"> Орыс
<BR>
```

```
<INPUT TYPE=checkbox NAME="Тіл" value="English"> АҒЫЛ-
ШЫН <BR>
```

```
<INPUT TYPE=checkbox NAME="Тіл" value="German"> Неміс
```

Қарастырылған тегтерді пайдаланып, күрделі емес формалар ұйымдастыруға болады. Жасалған форманды Интернет тармағына орналастырсаң, оны миллиондаған адамдар көреді. Бірақ ондағы сұрақтарға берілген жауаптар саған жетіп келмейді. Жасалған форма жауаптарымен бірге қайтып келуі үшін, <INPUT> тегі **SUBMIT** (қосылу) мәні TYPE параметрі пайдаланылады. Формадағы мәтін өрісін тазалау үшін <INPUT> тегі **RESET** (қайта таңдау) мәнді TYPE параметрі пайдаланылады. Аталмыш тегте VALUE параметрі істетілсе, браузер экранға түйме жасайды.

Мысалы:

```
<INPUT TYPE="submit" VALUE="Форманы жөнелту">
```

тегі экран ішінде «Форманы жөнелту» деген сөйлем жазылған түйме шығарады, бұл түйме таңдалса, формадағы мәліметтер қажетті мекенжайға жөнетіледі.

```
<INPUT TYPE="reset" VALUE="Форманы тазалау">
```

тегі экранға ішінде «Форманы тазалау» деген сөйлем жазылған түйме шығарады, бұл түйме таңдалса, барлық мәліметтер өшіп, мәліметтерді жаңарту мүмкіндігі туады.

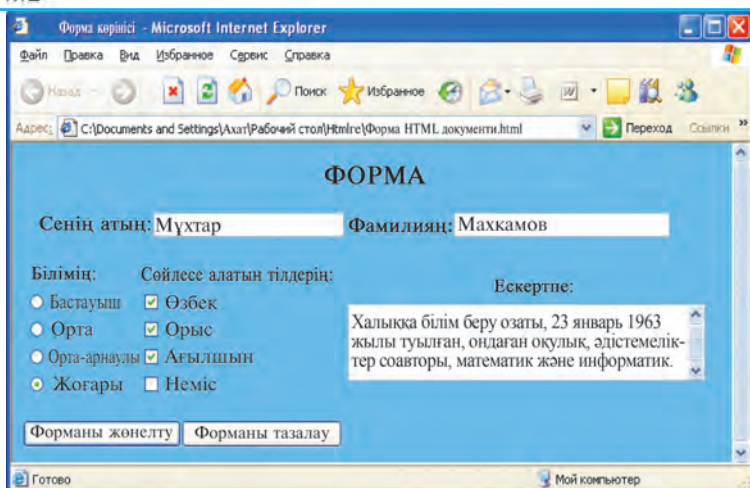
Төменде форма үшін HTML-құжат және оған сәйкес web-бет келтірілген:

Жоғарыдағы форманы жасауда web-бетке кесте орналастыру мүмкіншілігін пайдаландық. Сондықтан қосарлама таңдау түймелері іргелес бейнеленген.

Форма жасауда таңдау мүмкіншілігін <SELECT> жұп тегі көмегімен де жүзеге асыруға болады. Мұнда таңдалуы керек болған мәліметтер жылжымалы тізім сияқты кескінделеді. Тізім элементтері <SELECT> тегі ішінде қолданылған <OPTION> тақ тегі көмегімен енгізіледі. SELECT тегінің NAME, SIZE,

```

<HTML>
<title>Форма түрі</title>
<BODY BGCOLOR="#55Aaff"><H2 ALIGN="center">ФОРМА</H2>
<FORM METHOD=POST ACTION="mailto:rtm@umail.uz">
<FONT SIZE=4>Сенің атың: <INPUT NAME="Мұхтар" SIZE=26>
Фамилияң: <INPUT NAME="Махкамов" SIZE=30><P>
<table>
<TR><TH>Білімің:<BR></TH><TH>Сөйлесе алатын тілдерің:<B
R></TH><TH>Ескертпе:</TH></TR> <TR><TD><INPUT TYPE=radio
NAME="Білім" value="Бастауыш">
Бастауыш</TD>
<TD><INPUT TYPE=checkbox NAME="Тіл" value="Uzbekish">
Өзбек</TD>
<TD ROWSPAN=4><TEXTAREA ROWS=4
COLS=40></TEXTAREA></TD> </TR>
<TR><TD><INPUT TYPE=radio NAME="Білім" value="Орта">Орта
</TD>
<TD><INPUT TYPE=checkbox NAME="Тіл" value="Russian">Орыс
</TD></TR>
<TR><TD><INPUT TYPE=radio NAME="Білім"
value="Орта-арнаулы">Орта-арнаулы</TD>
<TD><INPUT TYPE=checkbox NAME="Тіл" value="English">Ағылшын
</TD></TR>
<TR><TD><INPUT TYPE=radio NAME="Білім" value="Жоғары">
Жоғары</TD>
<TD><INPUT TYPE=checkbox NAME="Тіл" value="German">Неміс
</TD></TR>
</table>
<P><INPUT TYPE="submit" value="Форманы жөнелту "><INPUT
TYPE="reset" value="Форманы тазалау "></FONT></FORM>
</BODY>
</HTML>
    
```



MULTIPLE сияқты параметрлері бар. Параметрлердің жұмысын мына төменде келтірілген мысалға қарап түсініп алуға болады.

```
<html>
<title>Select</title>Нешінші сыныпта оқисың?<br>
<SELECT name="" size=3>
<OPTION value="9">9-сыныпта <OPTION value="8">8-сыныпта
<OPTION value="7">7-сыныпта <OPTION value="6">6-сыныпта
<OPTION value="5">5-сыныпта </select>
</html>
```

Нешінші сыныпта оқисың?

Бұл мысалдан форманың таңдау бөлімі өте аз орын алатынын көруге болады. Бұл мысалдағы таңдау нәтижесінде керекті мекенжайға тек болғаны бір мән – «8» жіберіледі. Егер бірнеше мән таңдау мүмкіндігі қажет болса, мәнді талап етпейтін MULTIPLE параметрін қосудың өзі жеткілікті.



Сұрақтар мен тапсырмалар

1. *Өміріңде кездескен формалар туралы әңгімелеп бер.*
2. *Формалар не үшін қолданылады?*
3. *Web-бетте форма қалай ұйымдастырылады?*
4. *Форманы жөнелту туралы айтып өт.*
5. *Формадағы мәтін өрісі қалай белгіленеді?*
6. *Формада бірнеше жолдары бар мәтін өрісі қалай ұйымдастырылады?*
7. *Формада алдын ала берілген жауаптар таңдаудың ұйымдастыру әдістерін түсіндір.*

Жаттығулар

1. «Мен Өзбекстанды не үшін сүйемін?» атаулы тақырыптағы web-беттер конкурсына қатысу үшін сұрақ формасын дайында.
2. «Достасуға ұсыныс жасау» атты форма дайында. Онда сен білуің керек болған мәліметтер бейнеленсін.
3. «Кітапхана анкетасы» атты форма дайында.

63-сабақ. Web-беттегі «өту» және пішіндер тақырыбын қайталау

1. «Мен дайындаған web-беттер» атаулы web-бет дайында. Онда негізгі беттен өзің дайындаған web-беттерге көшу және артқа қайту сурет пен мәтін арқылы орындалсын.

2. «Информатика оқулықтары туралы ой-пікірлер» атты форма жаса. Онда оқулық туралы ой-пікірлерді жинау мақсат етіп алынсын.

3. 5-сыныптан 9-сыныпқа дейін компьютерде орындаған практикалық істерінді ашып беретін, «Мектеп, информатика және мен» атаулы web-бет дайында.

64-сабақ. Интерактивті web-сайттар

Web-сайттар – гипержүгінулер арқылы (hyperlink) байланысқан бірнеше web-беттер кешені. Оларды шартқа байланысты мынадай екі түрге бөлуге болады:

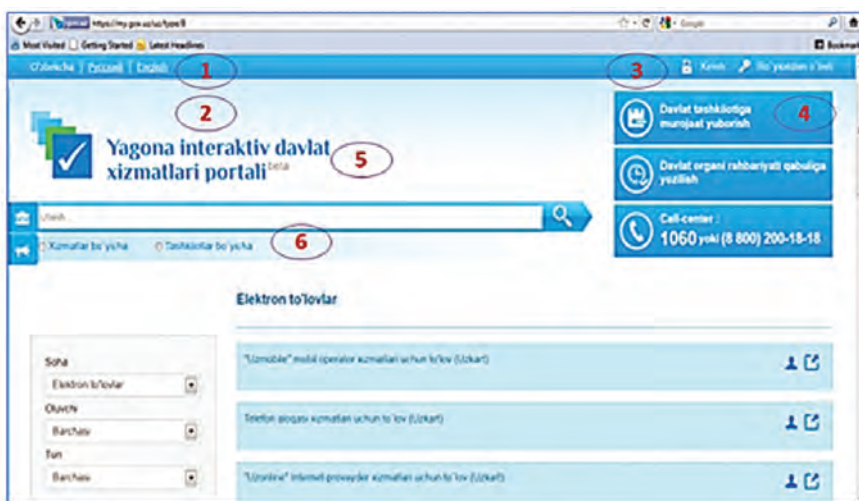
статистикалық	динамикалық
----------------------	--------------------

Статистикалық web-сайттар – бір-бірімен байланысты кодтандырылған, мәтіндік, суретті және басқа да түрдегі ақпараттан тұратын web-беттер кешені. Олардың құрамын пайдаланушының назарын тартатын стандартты күйдегі құжаттар мен деректер түзеді. Егер оларды жаңалату немесе қосымша мәлімет қосу керек болса, онда бағдарламаның кодын әр кез өзгертіп отыруға тура келеді. Ал бұл өз кезінде көп уақыт пен еңбекті талап етеді де, web-беттердің сан жағынан артқаны web-сайтты басқаруды күрделендіріп жібереді. Ескерте кету керек, Интернеттің іргесін қалаған алғашқы кездегі web-сайттардың бәрі статистикалық түрде болған.

Қазіргі уақытта web-сайттар динамикалық түрде істеп шығарылады. Динамикалық web-сайттар – пайдаланушының сұрағын орындаған кезде ақпараттары бір-бірімен байланысты болмаған түрде өзгеріп тұратын web-беттер кешені. Динамикалық web-сайттағы ақпаратпен жұмыс істеу, пайдаланушы сұрағы бойынша серверде сақталып отырған деректер базасына жүгіну арқылы ұйымдастырылады.

Статистикалық пен динамикалық web-сайттардың айырмашылығын төмендегі мысалдан көруге болады (my.gov.uz – бірыңғай мемлекеттік интерактивті қызмет көрсету порталы). Суретте бейнеленген web-беттер бір web-сайтқа тиесілі, сондай-

ақ пайдаланушының сұрағына сүйене отырып, бірінші web-беттің орнына екінші web-бет ашылған.



Егер тап осы web-сайт статистикалық түрде құрылғанда, онда екі сайттың әрқайсысын бейнелейтін бірдей ақпарат, әрбір сайтқа арналып қайта кодталатын еді (қайталанған ақпарат цифрлар арқылы бейнеленген). Бұл web-сайт – динамикалық web-сайт, сондықтан арнайы сценарий негізінде web-сайттың тиісті бөлімін өзгерту жолымен бірінші web-бетке екіншісі өтеді. Web-беттің өзгеріссіз қалған бөлігіне арналып, бір рет жазылған кодтар екі беттің әрқайсысына да ортақ болады. Демек, web-сайт бір-біріне

жақын, өте көп web-беттерден тұратын болса, динамикалық түрдегі web-сайттың маңызы өте зор болмақ. Динамикалық web-сайттардың негізгі артықшылықтарының бірі – ақпарат ресурстарын администратор терезесінен басқару оңай болады.

Динамикалық web-сайттар интерактивті (ағылш. Interaction – өзара әсер ете алу) технологияларды пайдалану үшін өте қолайлы, сондықтан осындай технологияларды қолданған web-сайттар **интерактивті web-сайттар** деп аталады. Қазіргі уақытта web-сайттардың интерактивті мүмкіншіліктеріне ерекше назар аударылуда. Дегенмен, көптеген жағдайда флеш-анимация яки мультимедиа ресурстары бар web-сайттарды интерактивті web-сайттар деп қате түсіндіреді.

Динамикалық web-сайттар тек қана деректерді қарау немесе танысу мүмкіншіліктерін ұсынумен қанағаттанбай, сондай-ақ онда тіркелу, хабар жіберу әрі қабылдау, онлайн (ағылш. online – байланыста, желіде) сұрақ-жауап жүргізу, тапсырыс бойынша мәлімет алу, түрлі есептегіштер мен басқа да элементтер арқылы пайдаланушыға «қатынас» жасау мүмкіншілігін де ұсынады. Сонымен бірге, көптеген web-сайттар пайдаланушының сайт әкімшілігімен онлайн-әңгіме өткізу, онлайн-чаттар (ағылш. Chatter – сұхбаттасу) нақты уақыт режимінде тікелей байланысу сияқты мүмкіншіліктер береді.

Интерактивтік қабілетін web-сайтқа ендіру үшін арнайы бағдарламалық кодтар – сервер скриптері пайдаланылады. Пайдаланушыдан алынған деректер серверде қайта өңделген сон, бұл скриптер олардың web-бетте бейнелеуін қамтамасыз етеді.

Әдетте, браузер html-файлды оқи алады; егер тап осы файлда сервер скрипті бар болса, ондағы амалдар әуелі солардың ішіндегі сценарий бойынша орындалады, кейін шығарылған нәтижелер браузерде бейнеленеді. Скрипттер серверде орындалады, ал нәтиже браузерге жіберіледі, сондықтан сервер скриптінің бастапқы коды браузерде көрінбейді.

Сервер скриптері мыналарды орындай алады:

- кез келген ақпаратты жедел қосу әрі өзгерту;
- кез келген деректерге яки деректер базасына кіруге мүмкіндік беру;
- пайдаланушы сұрағына жауап беру яки оларға мәлімет жіберу;

- пайдаланушының қалауынша web-бетті өзгерту әрі баптау және т.б.

Интерактивті web-сайттардың бірнеше тәртіп ережелері бар, солардың ішінде негізгі ережелер мыналар:

– web-сайттағы web-беттердің барша бөлімі дерлік сервер бағдарламаларының жәрдемімен туындайды (генерация) және қайта өңделеді;

– web-беттер тудыру үшін барша ақпарат тиісті деректер базасынан алынады. Деректер базасы арнаулы сервер компьютерлерінде түрлі көріністе сақталады;

– көптеген web-сайттарда қатынас жасауға шек қоятын элементтер пайдаланылады. Бұл шектеуіштер түрлі клиенттер үшін түрліше болуы мүмкін. Қарапайым клиенттер web-сайтты тек қарап шыға алады, ал басқа клиенттерге өзгертулер енгізуге мүмкіндік беріледі.

Бұл жерде идентификациялау (лат. identifico – теңестіру, бірдей деп есептеу) амалы, яғни логин (пайдаланушы идентификаторының аты) және пароль (фран. parole – сөз) элементтері пайдаланылады (төмендегі суретке қара);

- web-сайтта деректерді издестіру жүйесі бар болады.

The image shows a web browser window displaying the registration page of uMail.uz. The page has a header with the uMail.uz logo and navigation links: BOSH SAHIFA, LOYIHA HAQIDA, MALUMOT, YO'RIQNOMALAR, and QAYTA ALOQA. The main content area is titled 'RO'YXATDAN O'TISH' (Registration). Below the title, there is a paragraph of text in Uzbek explaining the registration process. The registration form consists of several input fields: 'Login' (with a dropdown menu showing '.id.uz'), 'E-mail' (with a dropdown menu showing '@umail.uz'), 'Familyasi', 'Ismi', 'Maqbul parol', and 'Parolni qayta kiriting'. There are also asterisks indicating required fields.

Динамикалық web-сайттар Java Script, PHP, Perl сияқты кең таралған және жалпы мойындалған бағдарламалау жүйелері жәрдемімен істеп шығарылып, олардың жәрдемімен сервер скриптері жазылады. Осы бадарламалау тілдері қалай да болса күрделі web-сайттарды жетілдіруге мүмкіндік береді. Дегенмен,

бұл міндетті атқаруда терең білім керек болғандықтан, оны бағдарламашылар жүзеге асырады.









Сұрақтар мен тапсырмалар

1. *Web-беттер ұйымдастыру технологиясына қарай қандай түрлерге бөлінеді?*
2. *Интерактивті web-беттер туралы мәлімет беріп, мысалдар келтір.*
3. *Логин мен парольдің пайдаланылуы керек болатын web-беттер туралы айт.*
4. *Электрондық поштаны өзіңе ашқанда, қандай web-бетпен істегенсің?*
5. *Компьютеріндегі электрондық оқулықтар мен әдістемеліктердің қайсысында интерактивті web-бет бар?*

65–66-сабақ. Өз бетімен істеуге арналған тапсырмалар

Төменде түсіндірілген тегтер мен символдарды пайдаланып, өзіндік web-беттерге өзгерістер енгіз.

1. Мәтіннің шрифт форматын өзгерту:

MS Word бағдарламасында	HTML тегі	HTML-да мысал	Web-бетте
 яғни 	<SUP> жұп тегі	ме^{кт}еп	ме _{кт} еп
 яғни 	<SUB> жұп тегі	ме_{ки}еп	ме _{кт} еп
 яғни 	<S>,<STRIKE>, жұп тегтерінің біреуі	мектеп <s>мектеп</s> <strike>мектеп</strike>	мектеп
Түсінік беру	<!> тақ тегі	<! мектеп>	көрінбейді

2. Арнайы белгілер кіргіз:

MS Word бағдарламасында	HTML жазбасы	HTML-да мысал	Web-бетте
<	<	<	<
>	>	>	>

&	&	&	&
"	"	"9-sinf"	"9-сынып"
Қатаң пробел	&NBSP	5 баға	5 баға
©	©	©	©

3. Эффектілер:

HTML теги	HTML да мысал	Web-бетте
<ACRONYM>	<acronym title= «274-мектеп»> 9-сынып< /acronym>	Жүгірткі көрсеткіші "9-сынып" жазуына бағытталғанда 9-сынып 274- мектеп
<HR>	<hr align=center size= color= red width= 60% noshade>	Беттің 60 пайызын алатын, ортасы бойынша тураланған қалыңдығы 4 пиксель қызыл түсті сызық сызылады, NOSHADE параметрі сызық дөңестігін алып тастайды.
<MARQUEE>	<marquee behavior= "alternate" width=60% height=30% bgcolor= blue> ӘРЕКЕТ </marquee>	Беттің горизонталь бағытында 60 пайызын, вертикаль бағытында 30 пайызын алатын, көк түсті төртбұрыш ішінде ақ түсті шрифт өлшемі 7-ке тең «ӘРЕКЕТ» сөзі қозғалып тұрады



67–68-сабақ. Қайталауға арналған тапсырмалар

Мына мәселелер арқылы білімінді сынап көріп, одан әрі нығайт.

1. Сенің бағдарламаға енгізген A мәтініндегі ең көп кездесетін символдар санын бос орындар санымен салыстырып көрсететін бағдарлама жаса. Мысалы: «a – 21» > «probel – 7».

2. Мынадай мазмұны бар бағдарлама жаса: енгізілген N -нің мәніне ($1 < N < 15$) қарай, ортасы экранның ортасында орналасқан N шеңбердің суретін экранға шығар.

3. Берілген N ($N > 21$) **разрядты** санның 2-ге, 3-ке бөлгенде шығатын қалдығын анықтайтын бағдарлама жаса

4. N санның барлық бөлінділерін табатын бағдарлама жаса.

5. «Қош бол, мектебім!» атаулы web-бет дайында.

ПАЙДАЛАНЬЛҒАН ӘДБИЕТТЕР

1. *Б. Болтаев, А. Абдуқодиров, Н. Тайлақов, М. Махкамов, А. Азаматов, С. Хафизов.* Информатика және есептеу техникасы негіздері. 9-сынып. — Т.: Чулпон, 2006 ж.

2. *Б. Болтаев, М. Махкамов, А. Азаматов.* Паскаль тілінде бағдарламалау. Әдістемелік қолданба. — Т.: 2007.

3. *Б. Болтаев, А. Азаматов, Ш. Хидиров, Б. Хуррамов, Г. Ишанходжаева.* Алгоритмдеу және Паскаль бағдарламалау тілі бойынша берілген амалдар мен есептерді шешу әдістері. Оқытушыларға арналған әдістемелік қолданба. — Т.: «NIHOL» баспасы, 2012.

4. *Л.Л. Босова, Ф.Ю. Босова.* Информатика, 7–9. Издательство «БИНОМ», — М.: 2013.

5. *Шауджова Л.З.* Информатика, 10–11. Издательство «Просвещение», — М.: 2000.

6. *А.Г. Кулаков, С.К. Ландо, А.Л. Семенов, А.Х. Шень.* Алгоритмика, V–VII классы. — М.: Дрофа, 1997.

7. *А.Н. Степанов.* Информатика, Учебник для вузов. Санкт-Петербург: «Питер», 2006.

Ескертпе: оқулықтағы күндер мен атаулар бойынша пайдаланылған әдебиеттердің толық тізімі Республикалық Білім орталығы жанындағы Информатика пәні бойынша Ғылыми — әдістемелік кеңестің 2015 жылғы 12 мартындағы қаулысымен бекітіліп, баспаға ұсынылған.

Бұл тізім Республикалық Білім орталығының web-сайтында (rtm.uz) орналастырылған.

МАЗМУНЫ

I ТАРАУ. АЛГОРИТМДЕУ НЕГІЗДЕРІ

1-сабақ. Есептерді компьютерде шығару кезендері.....	3
2-сабақ. Модель және оның түрлері.....	6
3-сабақ. Есептерді компьютерде шығару кезендері және модельдер түрі бойынша тақырыптарды қайталау.....	12
4-сабақ. Алгоритм ұғымы.....	13
5-сабақ. Алгоритмнің негізгі қасиеттері	17
6-сабақ. Алгоритм ұғымы және алгоритмнің негізгі қасиеттері бойынша тақырыптарды қайталау сабағы.....	19
7-сабақ. Алгоритмнің бейнелену әдістері.....	20
8-сабақ. Алгоритмнің бейнелену әдістері бойынша тақырыпқа арналған практикалық жұмыс	23
9-сабақ. Алгоритмнің негізгі түрлері.....	24
10-сабақ. Алгоритмнің негізгі құрылымдарына қатысты практикалық тапсырма.....	28
11-сабақ. Қайталауға арналған тапсырмалар.....	30

II ТАРАУ. БАҒДАРЛАМАЛАУ НЕГІЗДЕРІ

12-сабақ. Бағдарлама және бағдарламалау тілдері.....	31
13-сабақ. Турбо Паскаль 7.0 интегралданған ортасы.....	34
14-сабақ. Паскаль бағдарламалау тілінің әліпбиі мен құрылысы.....	37
15-сабақ. Тұрақты және айнымалы шамалар.....	41
16-сабақ. Тұрақты және айнымалы шамалар тақырыбын қайталау.....	45
17-сабақ. Кестелік шамалар.....	46
18-сабақ. Кестелік шамалар тақырыбын қайталау.....	50
19-сабақ. Стандартты функциялар мен процедуралар, алгебралық өрнектер.....	50
20-сабақ. Стандартты функциялар мен процедуралар, алгебралық өрнектер тақырыбын қайталау.....	55
21-сабақ. Меншіктеу және деректерді экранға шығару операторы.....	55
22-сабақ. Меншіктеу және деректерді экранға шығару операторлар тақырыбын қайталау.....	59
23-сабақ. Деректерді жадына диалог әдісімен енгізу операторы.....	60
24-сабақ. Деректерді жадына диалог әдісімен енгізу операторы тақырыбын қайталау.....	63
25-сабақ. Мәтін режиміндегі экранмен жұмыс істеу.....	64
26-сабақ. Мәтін режиміндегі экранмен жұмыс істеу тақырыбын қайталау.....	68
27-сабақ. Сызықтық бағдарламалар жасау.....	68
28-сабақ. Сызықтық бағдарламалар жасау тақырыбын қайталау.....	71
29-сабақ. Өту және тармақталу операторлары.....	72

30-сабақ. Өту және тармақталу операторлар тақырыбын қайталау...	76
31-сабақ. Тармақталатын құрылымдық бағдарламалар жасау.....	76
32-сабақ. Тармақталатын құрылымдық бағдарламалар жасау тақырыбын қайталау.....	80
33-сабақ. Параметрлік қайталау операторы.....	80
34-сабақ. Параметрлік қайталау операторы тақырыбын қайталау....	84
35-сабақ. Шартты қайталау операторлары.....	85
36-сабақ. Шартты қайталау операторлар тақырыбын қайталау.....	88
37-сабақ. Қайталауға арналған тапсырмалар.....	88
38-сабақ. Символдық әрі жолдық шамалармен жұмыс істеу.....	89
39-сабақ. Символдық және жолдық шамалармен жұмыс істеу тақырыбын қайталау.....	94
40-сабақ. Экранды Паскаль тіліндегі графикалық күйге өткізу.....	94
41-сабақ. Экранды Паскаль тіліндегі графикалық күйге өткізу тақырыбын қайталау.....	99
42-сабақ. Паскаль тілінде пішіндер сызу мүмкіншіліктері.....	99
43-сабақ. Паскальдің пішіндер сызу мүмкіншіліктері тақырыбын қайталау.....	103
44-сабақ. Файлдармен жұмыс істеу.....	104
45-сабақ. Файлдармен жұмыс істеу тақырыбын қайталау.....	109
46-сабақ. Процедуралар мен функциялар.....	109
47-сабақ. Процедуралар мен функциялар тақырыбын қайталау.....	113
48–49-сабақ. Қайталауға арналған тапсырмалар.....	113

III ТАРАУ. WEB-БЕТ ДАЙЫНДАУ

50-сабақ. HTML туралы түсінік.....	114
51-сабақ. Мәтінді Web-бетке енгізу.....	118
52-сабақ. Мәтінді Web-бетке енгізу тақырыбын қайталау.....	123
53-сабақ. Шрифт өлшемі мен түсі, web-беттің фоны.....	123
54-сабақ. Шрифт өлшемі мен түсі, web-беттің фоны тақырыбын қайталау.....	127
55-сабақ. Web-беттегі графика.....	127
56–57-сабақ. Web-беттегі графика тақырыбын қайталау.....	132
58-сабақ. Web-бетке тізім орналастыру.....	132
59-сабақ. Web-бетке кесте орналастыру.....	136
60-сабақ. Web-бетке тізім мен кесте орналастыру тақырыбын қайталау.....	141
61-сабақ. Web-беттегі «өту» (гипержүгіну).....	141
62-сабақ. Пішіндер.....	144
63-сабақ. Web-беттегі «өту» және пішіндер тақырыбын қайталау....	150
64-сабақ. Интерактивті web-сайттар.....	150
65–66-сабақ. Өз бетімен істеуге арналған тапсырмалар	154
67–68-сабақ. Қайталауға арналған тапсырмалар	155
Пайдаланылған әдебиеттер.....	156

Bahodir Jalolovich Boltayev
Axat Raxmatovich Azamatov
Abror Davlatmirzayevich Asqarov
Muxtor Qurbonovich Sodiqov
Gulnoza Axatovna Azamatova

INFORMATIKA

VA HISOBLASH TEXNIKASI ASOSLARI

(Qozoq tilida)

2-nashri

Umumiy o'rta ta'lim maktablarining
9-sinfi uchun darslik

Аударған Райхан Бабатаева

Редакторы Р. Өктем

Көркемдеуші редакторы Сардор Курбанов

Техн. редакторы Елена Толочко

Корректор Әсіл Джавғашев

Компьютерде терген Гүлбайра Ералиева

Лицензия нөмірі АІ № 163. Басуға 2015 жыл 25-июньде кол қойылды. Пішімі 60×90^{1/16}. Times KAZ гарнитурасы. Кеглі 11. Офсеттік басылыс. Шартты баспа табағы 10,0. Есептік баспа табағы 8,9. Таралымы 4836 нұсқа. Келісім № 31–2015. Тапсырыс № 237.

Өзбекстан баспасөз және ақпарат агенттігінің Чулпан атындағы баспа-полиграфия шығармашылық үйі. 100129, Ташкент, Науаи көшесі, 30.
Телефон: (371) 244-10-45. Факс (371) 244-58-55.

Өзбекстан баспасөз және ақпарат агенттігінің Фафур Ғұлам атындағы баспа-полиграфия шығармашылық үйі мен «Ўзбекистон» баспа-полиграфия шығармашылық үйінің баспаханаларында бірлесіп шығарылды. 100128, Ташкент, Лабзак көшесі, 86./100129, Ташкент, Науаи көшесі, 30.

Б 73 Информатика және есептеу техникасы негіздері. Жалпы орта білім беретін мектептердің 9-сыныбына арналған оқулық /Б. Болтаев [және басқ.]; жауапты редактор Н. Тайлақов.— Т.: Чулпан атындағы БПШҮ, 2015. — 160 б.

1. Болтаев, Б.

ISBN 978-9943-05-747-0

УДК: 372.8:004=512.122(075)

ББК 32.81(5Каз)я721

Пайдалануға берілген оқулықтың жағдайын көрсететін кесте

№	Оқушының аты, фамилиясы	Оқу жылы	Оқулықтың пайдалануға берілгендегі жағдайы	Сынып жетекшісінің қолы	Оқулықты тапсырғандағы жағдайы	Сынып жетекшісінің қолы
1						
2						
3						
4						
5						
6						

Пайдалануға берілген оқулықты оқу жылы аяқталғанда қайтарып тапсырады. Жоғарыдағы кестені сынып жетекшісі төмендегі бағалау мөлшері негізінде толтырады:

Жаңа	Оқулықты алғаш рет пайдалануға берілгендегі жағдай
Жақсы	Мұқаба бүтін, оқулықтың негізгі бөлігінен ажыралмаған. Барлық парақтары бар, жыртылмаған, көшпеген, беттеріне жазбаған және сызбаған.
Орташа	Мұқаба мыжылған, сәл сызылған, шеттері жейілген, оқулықтың негізгі бөлігінен ажыраған жерлері бар. Пайдаланушы жағынан қанағаттанарлық жөнделген. Жұлынған, кейбір беттері сызылған.
Нашар	Мұқаба былғанған, сызылған, жыртылған, негізгі бөлігінен ажыраған немесе мүлдем жоқ, нашар жөнделген. Беттері жыртылған, парақтары жетіспейді, сызып, бояп тасталған. Оқулық қалпына келтіруге жарамайды.