



Қазақстан Республикасы Білім және ғылым министрлігі
С. Торайғыров атындағы Павлодар мемлекеттік университеті
Физика, математика және ақпараттық технологиялар факультеті
Информатика және ақпараттық жүйелер кафедрасы

050604 Физика мамандығының студенттеріне арналған
" ПРОГРАММАЛАУ" пәні бойынша
зертханалық жұмыстарды орындауға арналған
ӘДІСТЕМЕЛІК НҰСҚАУЛАР



БЕКІТЕМІН
ФМжАТ факультетінің деканы
_____ С.К.Тлеуқенов
200 __ ж. «__» _____

Құрастырушы: Аға оқытушы Нұрғазина Б.Қ.

Информатика және ақпараттық жүйелер кафедрасы

050604 - «Физика» мамандығының студенттері үшін

"Программалау" пәні бойынша зертханалық жұмыстарға әдістемелік нұсқаулар

Кафедра мәжілісінде бекітілді, 200__ж. «__»_____ Хаттама №_____.

Кафедра меңгерушісі _____ Нұрбекова Ж.Қ.

Факультеттің әдістемелік кеңесінде құпталды,

200__ж. «__»_____ Хаттама №_____.

ӘК төрайымы _____ А.Т.Кишубаева

Зертханалық жұмыстардың мақсаты:

1. Студенттерге бағдарламалау тілінің негізгі құрылымдарын үйрету, командалар мен операторлардың қызметі мен пайдалану ережелерімен таныстыру.
2. Студенттердің алгоритмдік ойлау дағдыларын қалыптасыту.
3. Өз бетімен оқып-үйренуге, талдауға, бағдарлама бойынша нәтижені анықтауға үйрету.
4. Есепке алгоритмдер мен бағдарламалар құру, олардың тиімділерін таңдау және нәтижелерге талдау жасау әдістерін меңгерту.

Зертханалық жұмыстардың тәртібі:

1. Жұмыстың теориялық бөлігімен танысады.
2. Берілген есептің алгоритмімен танысады.
3. Бағдарлама тексін қарап, оның құрылымына талдау жасайды. Ондағы командалардың пайдаланылуын негіздейді.
4. Алгоритмнің орындалу тәртібін ауызша түсіндіреді.
5. Қандай да бір мәндермен, бағдарламаның орындалу протоколын толтырып, нәтижелерді компьютерсіз анықтайды. Пайдаланылатын мәндер бағдарламадағы барлық жағдайларды қамту қажет.
6. Бағдарламаны компьютерде орындап, 5-пунктте анықталған нәтижелердің дұрыстығын тексереді.

№1 зертханалық жұмыс. Алгоритм ұғымы

Алгоритм – дегеніміз алға қойылған мақсатқа жету немесе берілген есепті шешу бағытында орындаушыға (адам, ЭЕМ, робот, автомат) біртіндеп қандай әрекеттер жасау керектігін түсінікті түрде әрі дәл көрсететін нұсқаулар тізбегі.

Алгоритмдер құрылымы үш түрге бөлінеді:

- 1) сызықтық құрылым;
- 2) тармақталған құрылым;
- 3) қайталанушы құрылым.

Сызықтық алгоритмдік құрылым

Алгоритмнің командалары тізбектей, жазылу реті бойынша бірінен соң бірі орындалады. Әдетте формула бойынша есептеуде қолданылады. Формуладағы айнымалының мәнге ие болуы немесе мәнінің өзгеруі меншіктеу командасымен ($=$) жүзеге асырылады.

Меншіктеу командасының жазылу түрі:

Айнымалы := өрнек ; Айнымалы :=мән;

1 мысал. А айнымалысына 10 мәні меншіктелсін. $a := 2 * a$ – 1-қадам Меншіктеу командасы орындалған соң А айнымалысы қандай мән қабылдайды?

Шешуі: $a := 10$; $a := 2 * 10 - 1$; $a := 19$; Меншіктеу командасы орындалған соң а айнымалысы 19 мәніне ие болады.

2 мысал. $x := 3$ және $y := 7$ айнымалыларының мәндерін ауыстыру қажет.

Шешуі. Алгоритмде x және y мәндерін уақытша сақтау үшін қосымша z айнымалысы қажет.

Алгоритм меншіктеу командаларының тізбегінен тұрады:

$x := 3$; $y := 7$; $z := x$; $x := y$; $y := z$

Орындалуы: $x := 3$; $y := 7$; $z := 3$; $x := 7$; $y := 3$;

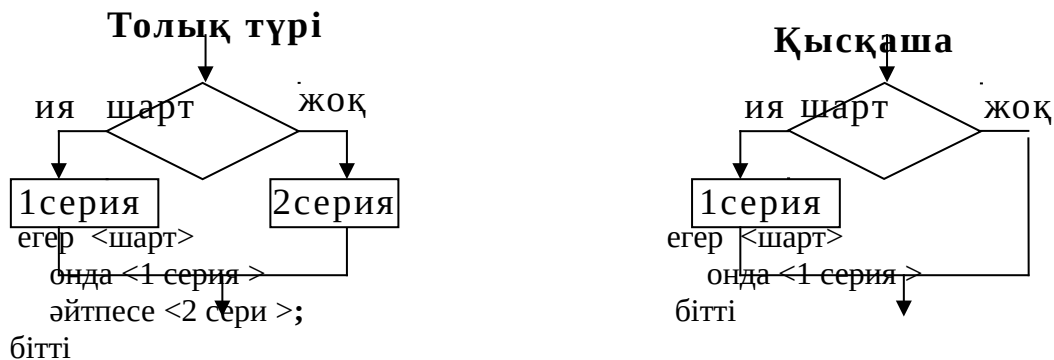
3 мысал. x және y айнымалыларының мәндерін қосымша айнымалыны қолданбай ауыстыру қажет.

Шешімі. Алгоритм меншіктеу командаларының тізбегінен тұрады: $x := 3$; $x := 7$; $x := x - y$; $y := x + y$; $x := y - x$

Орындалуы: $x:=3; y:=7; x:=3-7; x:=-4;$
 $y:=-4+7; y:=3; x:=3-(-4); x:=7;$

Тармақталған алгоритмдік құрылым

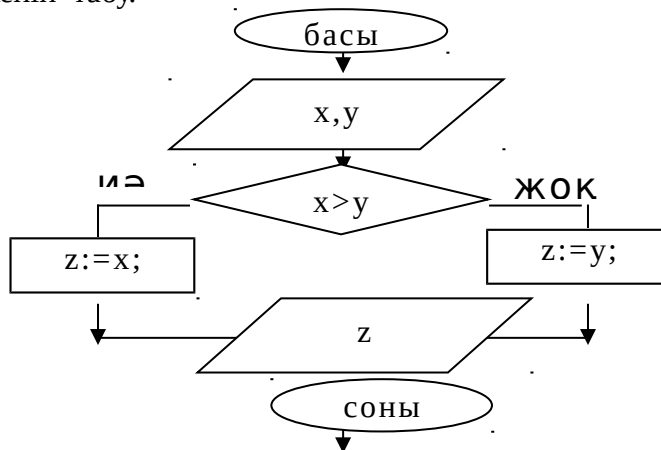
Кейбір жағдайларда орындалатын әрекеттер белгілі бір шартқа байланысты болады. Шарт логикалық блокта тексеріледі. Егер шарт орындалса, әрекеттердің бір бір тармағы, әйтпесе екінші тармағы орындалады. Шарт дұрыс болмаса ешқандай әрекеттер орындалмауы да мүмкін.



4 мысал. Екі санның үлкенін табу.

Есепті шығару алгоритмі

басы
 енгізу x, y
 егер $x > y$
 онда $z := x$
 әйтпесе $z := y$
 бітті
 шығару z
 соңы



Алгоритмнің циклдік құрылымы

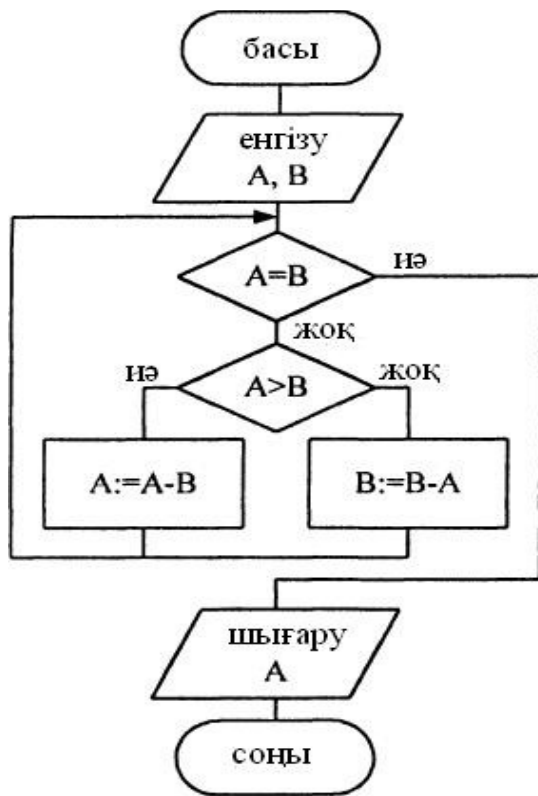
Циклдік алгоритмдік құрылымның үш түрі бар: параметрлі цикл (үшін), кейінгі шартты цикл (дейін), алғы шартты цикл (әзірше) (3 сурет).

Егер алгоритмде қайталану саны алдын ала белгілі үрдісті ұйымдастыру қажет болса, онда параметрлі цикл (үшін) қолданылады. Ондай алгоритмде цикл параметрінің бастапқы мәні, соңғы мәні және өзгеру қадамы алдын ала айқын болады.

Егер алгоритмде қайталану саны алдын ала белгісіз болса, онда кейінгі шартты цикл немесе алдыңғы шартты цикл қолданылады. Бұл циклдерде қайталану шарты басқарылуы қажет. Әйтпесе қайталану тоқтамай цикл шексіз орындала береді.

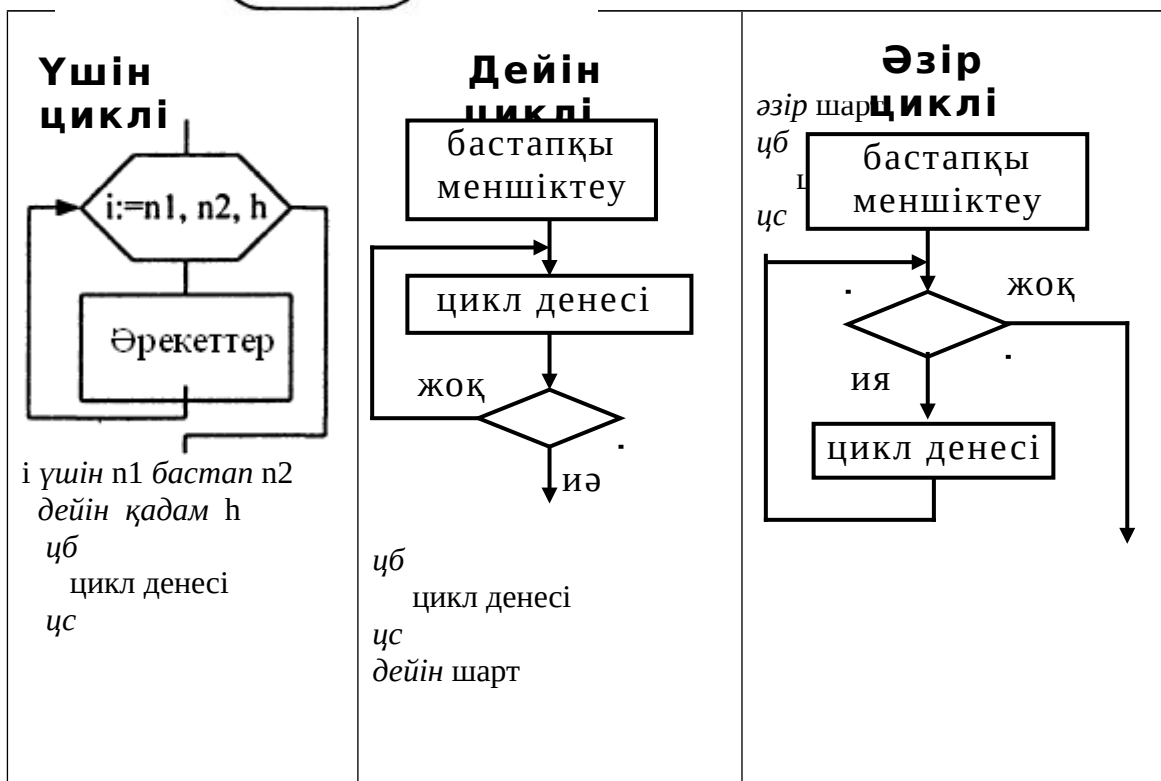
6 мысал. Екі санның ең үлкен ортақ бөлгішін табу (Евклид алгоритмі).

Есепті шығару алгоритмі 4 суретте көрсетілген.



басы
 енгізу A, B
 әзір $A \neq B$
 цб
 егер $A > B$
 онда $A := A - B$
 әйтпесе $B := B - A$
 бітті
 цс
 шығару A
 соңы

4 сурет – Евклид алгоритмінің блок-схемасы



№2 зертханалық жұмыс. Паскаль ортасы. Бағдарламаның құрылымы.

Мақсаты: Pascal-дың жұмыс ортасымен танысу, негізгі дағдыларын қалыптастыру.

Теориялық бөлім

1.Паскаль тілін жұмысқа қосу. Жұмыс столындағы Паскальға сәйкес жарлық жұмысқа қосылады.

2. Менюмен жұмыс. **F10** клавишасы менюді жұмысқа қосады. Меню жұмысқа қосылған соң, ерекше түске боялады. Команданы жұмысқа қосуды тездетуге болады. Ол үшін меню жолындағы белгіленген әріпке сәйкес клавиша басылады. **Alt** клавишасы мен белгіленген әріпті басса, оған сәйкес команда іске қосылады. Менюден шығу үшін **Ecs** клавишасы басылады.

Меню пункттеріне сипаттама

1. File

Open-кеңейтілуі .pas файлды ашады.

New-жаңа терезе ашады.

Save (F2)-файлды өз атымен сол орнына сақтайды.

Save as-файлды басқа атпен, басқа каталогқа жазуға болады.

Change dir- ағымдағы (жұмыс істеп отырған) каталогты өзгертуге мүмкіндік береді.

Print-ағымдағы терезедегі ақпаратты принтерге шығарады.

Printer Setup-принтер орнату.

DOS shell- **DOS** режиміне шығады. exit командасымен Паскаль ортасына оралады.

Exit (Alt-X)-Паскальмен жұмысты аяқтайды. Файлды сақтау қажет.

Терезелермен жұмыс істеу. Терезе дегеніміз - біздің барлық іс-әрекетімізді жүзеге асыратын экранның бөлігі. Оның размерін өзгертуге, орнын ауыстыруға, ашуға, жабуға болады. Паскаль ортасында бірнеше терезені қатар ашуға, әрбіреуімен жеке жұмыс жасауға, олардың арасында информация алмастыруға болады. Жұмыс істеп отырған терезе **белсенді (активті)** деп аталады. Белсенді терезе қос рамкалы және онда жабу кнопкасы, масштабтау кнопкасы, терезені жылжыту кнопкасы, терезенің өлшемін өзгерту бұрышы болады.

Меню- WINDOW

Tile-терезелерді ретпен орналастыру.

Cascade-терезелерді каскадпен орналастыру.

Close all-барлық терезелерді жабу.

Size/Move-терезенің өлшемін өзгерту (Shift+стрелкалар).

Zoom-терезені үлкейту немесе кішірейту.

Next - терезелерді ретпен қарау.

Close (Alt + F3)-белсенді терезені жабу.

Previous- терезелерді кері ретпен қарау.

List-барлық ашық терезелер тізімін көрсетеді.

Alt + № - № номерлі терезеге өту. Alt клавишасы мен керек терезенің номері қатар басылады.

Alt + F5 - Window/ User Screen - Бағдарламаны орындау нәтижесі бар пайдаланушы терезесі шығады

Тапсырма.

1.Паскальді жұмысқа қос.

2. Паскаль ортасында 5 терезе ашып, әрбіреуін сәйкесінше tereze1, tereze2, tereze3, tereze4, tereze5 деген атпен сақта.

3. Терезелерді ретпен орналастыр.

4. Терезелерді каскадпен орналастыр да, мына ақпаратты тер.

1-терезеге: {фамилияң мен атыңды}

2-терезеге: program p_orla;
const k=100;

3-терезеге: label m1,m2;
begin clrscr; writeln('Salem!'); writeln(' Men Pascalmyn!');
end.

4-терезеге: uses crt;

Осы ақпараттарды терген соң, әрбір терезені сақта.

5. Терезелерді ретімен, одан соң кері ретпен қарап, жазылған ақпаратты тексер.

6. 2-терезеге өтіп, мына жазуды қосып, сақта.

type kun=1..31;

7. Барлық ашық терезелердің тізімін шығар. 4-терезеге өтіп, мына жазуды қосып, сақта.

var k:kun; a:byte;

8. 1 және 2-терезелер экранға қатар сиятындай етіп, өлшемдерін өзгерт.

2. Меню-EDIT

Undo (Alt+BkSp)- алдыңғы әрекетті жояды.

Redo – Undoдағы әрекетті қайтарады.

Cut (Shift+Del) – белгіленген фрагментті жойып, буферге алады.

Copy (Ctrl+Ins) - белгіленген фрагментті буферге көшіреді.

Paste (Shift+Ins) – буфердегі фрагментті курсордың орнына қояды.

Clear (Ctrl+Del) – фрагментті жояды.

Show Clipboard- буфердегі ақпаратты көрсетеді.

Тапсырма.

9. 1,2,3,4 терезелердегі ақпараттарды ретімен 5-терезеге біріктіріп, мына тексті құрастыр.
{фамилияң мен атың}

```
program p_orla;
```

```
const k=100;
```

```
label m1,m2;
```

```
begin
```

```
clrscr;
```

```
writeln('Salem!'); writeln(' Men Pascalmyn!');
```

```
end.
```

```
uses crt;
```

Тексті сол атымен сақта.

3.Меню- SEARCH-текст фрагментін және қате табылған орынды іздейді.

4. Меню-RUN

RUN (Ctrl+F9)– бағдарламаны орындайды.

5. Меню- COMPILE

Compile – белсенді терезедегі файлды компиляциялайды.

Make – негізгі бағдарламаға кіретін модульдер қайтадан компиляцияланадыда,
орындалушы файлға біріктіріледі.

Build –алдыңғы пунктке ұқсайды.

Distination memoгу –орындалушы файл қайда орналасады, дискіде ме, әлде жадыда ма?

Primary file –компиляциялау қай файлдан басталатынын көрсетеді.

6. TOOLS – бағдарламаны жұмысқа қосу

7. OPTIONS –Паскаль ортасына қажетті каталогтардың жолы көрсетіледі.

Бағдарламаны компиляциялау. Бағдарламаны компиляциялау үшін Alt+F9 клавишалары қатар басылады. Егер текст дұрыс болса, ол жөнінде хабарлама шығады:
Compile Successful: Pres any key

Жұмысты жалғастыру үшін, кез келген клавишаны басу керек.

Бағдарламаны орындау. Бағдарламаны орындау үшін, Ctrl+F9 клавишасы басылады, немесе менюден /Run/ Run енгізіледі.

Тапсырма. 5-терезедегі бағдарламаны орында. Қате туралы хабарлама шықса, төмендегі құрылым бойынша бағдарламаны түзет. Uses бөлімін орнына келтір.

Паскаль тіліндегі программаның құрылымы

program < идентификатор >; - программаның аты
uses < идентификатор >; - библиотикалық модульдер тізімі
label ... белгілерді сипаттау.
const ... тұрақтыларды сипаттау.
type ... типтерді сипаттау.
var ... айнымалыларды сипаттау.
begin
операторлар программа денесі
end.

Бағдарламаны орындап, пайдаланушы терезесінен нәтижені қара.

Орындалушы (кеңейтілуі .exe) файл жасау. Паскаль тіліндегі бағдарлама тексін сақтағанда стандартты **.pas** деген кеңейтілу (расширение) беріледі. Бағдарламаны орындалушы файл ретінде сақтау үшін, оған **.exe** кеңейтілуі берілу керек. Ол үшін:

а) жадыға сақтау үшін: *Меню / Compile / Destination / Memory*

б) дискіге сақтау үшін: *Меню/Compile/Destination/Disk*

Memory немесе Disk Enter арқылы таңдалады.

Memory мен Disk-нің біреуі таңдап алынған соң, *Меню /Compile/ Make* енгізіледі немесе F9 басылады. Орындалушы файл жасалады. Каталогтағы файлдар тізімінде кеңейтілуі .exe файлы пайда болады.

Тапсырма. Орындалушы (кеңейтілуі .exe) файл жаса. Каталогты ашып, файлдың жасалғандығын тексер.

6. Жұмысты аяқта.

№3 зертханалық жұмыс. Математикалық функциялардың Паскаль тілінде жазылуы
Мақсаты: Қарапайым және күрделі математикалық өрнектердің Паскаль тілінде жазылуын меңгеру .

Теориялық бөлім

Математикалық өрнекті Паскаль тілінде жазу ережелері:

- 1.Өрнек бір жолға тізбектей жазылады, сондықтан амалдардың орындалу тәртібінің бұзылмауын ескеру керек.
2. Жоғары және төменгі индекстер ереже бойынша жазылады.
3. Функцияның аргументі жақша ішіне алынады.
4. Тригонометриялық функциялардың аргументі градуспен емес, радиан түрінде ($x \cdot \pi / 180$) жазылады.
5. Ашылған және жабылған жақшалар саны тең болады.
6. Математикалық өрнекте дөңгелек жақшалар ғана жазылады.

Дәрежені есептеудің стандарттық функциясы жоқ. Бірақ, оны стандарттық функциялар арқылы былайша өрнектеп жазуға болады:

$a^x = \exp(x \cdot \ln(a))$, (егер $a > 0$ болса ғана).

Сол сияқты, $\operatorname{tg}(x) = \sin(x) / \cos(x)$;

Математикада	Паскальда	Нәтиженің типі
$ x $ $\operatorname{abs}(x)$	бүтін, нақты	
x^2	$\operatorname{sqr}(x)$	бүтін, нақты
\sqrt{x}	$\operatorname{sqrt}(x)$	нақты
$\sin x$	$\sin(x)$	нақты
$\cos x$	$\cos(x)$	нақты
$\operatorname{arctg}(x)$	$\operatorname{arctan}(x)$	нақты
e^x	$\operatorname{exp}(x)$	нақты
$\ln x$	$\ln(x)$	нақты
нақты $\lg x$	$\ln(10)/\ln(x)$	нақты
π $\pi=3,141592..$		

Жұмыс технологиясы.

1.Паскаль терезесін ашып, мына жолдарды теріңіз.

```
Program ornek;
uses crt;
var x,y,z,a:real;
begin
clrscr;
x:=
y:=
z:=
writeln('z=',z);
readln;
end.
```

1 жаттығу

$x=5; y=3$ мәндерінде

$$z = \frac{\sqrt{x^2 + 2y} + \sin 15^\circ}{2x}$$

өрнегінің мәнін есептеу.

2.Бағдарламадағы мына жолдарды өзгертіңіз.

$x:=5;$

$y:=3;$

$z:=(\operatorname{sqrt}(\operatorname{sqr}(x)+2*y)+\sin(15*\pi/180))/(2*x)$

3.Бағдарламаны жұмысқа қосып, нәтижесін қараңыз.

Сұрақтарға жауап беріңіз.

1. $(\sqrt{\sqrt{x}+2*y}+\sin(15*\pi/180))$ өрнегі неге жақшаға алынған?
2. $15*\pi/180$ формуласы нені білдіреді ?

2 жаттығу.

$$z = \left| \frac{|x - 100| \cdot 3}{2x + \ln y} + 2\operatorname{tg} y \right| + \frac{e^{x+y}}{(1+x)^y}$$

4. Z:= жолына мынаны теріңіз:

$$Z:=\operatorname{abs}(\operatorname{abs}(x-100)*3/(2*x+\ln (y))+2*\sin(y)/\cos(y))+\exp(x+y)/\exp(y*\ln(x));$$

5. Бағдарламаны жұмысқа қосып, нәтижесін қараңыз.

Сұрақтар:

3. $\sin(y)/\cos(y)$ нені білдіреді?

4. $\exp(x+y)$ неге жақшаға алынған? Осы жазуды $\exp(x)+y$ деп жазуға бола ма?

3 жаттығу.

$$z = \frac{3e^5 y^5 + \sqrt{\cos^2 y + \sin y^2}}{\sqrt[3]{(x+y)^2}} \quad \text{өрнегінің мәнін есептеу.}$$

Z:= жолына мынаны теріңіз:

$$Z:=(3*\exp(5)*\exp(5*\ln(y))+\sqrt{\sqrt{\cos(y)+\sin(\sqrt{y})}})/\exp(2/3*\ln(x+y));$$

Сұрақтар:

5. e^5 және y^5 екеуінде де дәреже көрсеткіші 5, бірақ жазылуы әр түрлі. Неге?

7. $\dots \sin(\sqrt{y}))$ жолдарында неге 4 рет \sqrt{y} жақша қойылған?

№4 тақырып Негізгі процедураларға бағытталған алгоритмдік тілдерде программалау.

№3 зертханалық жұмыс. Шығару, енгізу процедуралары

Мақсаты: Экранға әр түрлі хабарлама, нәтиже шығаруды және мәндерді енгізуді үйрену.

Теориялық бөлім

WRITE – айнымалының мәнін, өрнектің мәнін, текстті экранға шығарады.

Жазылу форматы: **write(ln) (<параметр>);**

Жазылу түрлері:

writeln (өрнек); – өрнектің мәнін экранға шығарады. `writeln (sqrt(25));`

writeln(x); – x айнымалының мәні шығады.

writeln ('текст'); – апострофтың ішіндегі текст шығады.

writeln (өрнек, 'текст', айнымалылар);

writeln; – бос жол шығарады.

read(ln) – сандық мәндерді, символдарды, жолдарды клавиатурадан енгізеді.

Жазылу форматы: **read(ln)(айнымалылар тізімі);**

Жазылу түрлері:

readln(x) – x-тің мәні енгізіледі.

readln(a,b) – a,b мәндері бос орынмен ажыратылып енгізіледі.

readln- енгізу клавишасы басылғанын күтеді.

Read процедурасы орындалғанда, айнымалының мәнін енгізу үшін бағдарлама жұмысында кідіріс болады. Айнымалының мәнін оқушы клавиатурадан енгізеді. Егер read процедурасында айнымалылар тізімі көрсетілсе, онда тізімдегі айнымалылардың мәндері клавиатурадан бос орынмен ажыратылып енгізіледі. Әрбір read процедурасынан соң Enter клавишасы басылады.

1 жаттығу. Қадыр Мырзалиевтің «Ана тілі» тақпағының бір шумағын экранға шығару.

1. Бағдарлама тексін теріңіз

```
program text;
uses crt;
begin
clrscr;
Writeln;   Writeln('Ана тілі');   Writeln;
Writeln('Ана тілің арың бұл,');
Writeln('Ұятың боп тұр бетте.);
Writeln('Өзге тілдің бәрін біл,');
Writeln('Өз тіліңді құрметте');   Writeln;
Writeln('Қ.Мырзалиев');   Writeln;
readln;
end.
```

Бағдарламаны орындап, нәтижесін көріңіз. Экранда орналасуына назар аударыңыз

2. Мына жолдарға өзгерістер енгізіңіздер

```
Writeln ('Ана тілі':20);
Writeln ('К. Мырзалиев':30);
```

Бағдарламаны орындап, нәтижесін көріңіз. Қандай айырмашылық бар? 20 және 30 цифрларын өзгертіңіз. Нәтижесіне түсініктеме беріңіз.

2 жаттығу. Жаңа терезеде мына жолдарды теріңіз

```
uses crt;
var x,y,a,b: byte;
begin
clrscr;           {1}
readln(x);       {2}
y:=10;           {3}
writeln(x);      {4}
writeln(x+y);    {5}
readln;
end.
```

Орындап нәтижені қараңыз да, дәптерге жазып алыңыз

3. Мына {4} және {5} жолдарды өзгертіңіз:

```
Writeln ('x=', x);
Writeln ('y+x=', x+y);
```

Орындап нәтижені қараңыз. Қандай өзгеріс бар? Қорытынды жасаңыз.

4. Мына өзгерістерді енгізіңіз. {3} жолдан соң мына жолды теріңіз:

```
a:= x*y;           {3a}
{4}, {5}жолдарды өзгертіңіз:  writeln (x,'*', y, '=', a);   {4}
Writeln ('x=', x:10, 'y=';10, 'a=';:10, a:10); {5}
```

Орындап нәтижені қараңыз. Қандай өзгеріс бар? Қорытынды жасаңыз. {5}жолдағы : белгісі қандай қызмет атқарады?

3 жаттығу. Жаңа терезеде жазыңыз:

```
uses crt;
var x,y,a,b:real; s:string;
begin
clrscr;           {1}
x:=100; y:=3;     {2}
a:=x/y;
writeln ('a=';a); {3}
writeln ('a=';10);
```

```
writeln ('a':10:5);    {4}
writeln('a':10:2);
writeln('a':10:0);
writeln('a':0:5);    {5}
writeln('a':0:2);    {6}
writeln('a':0:0);
readln;
end.
```

Бағдарламаны орындап, нәтижесін қараңыз. Экранға шығудың қандай айырмашылықтары бар? Санның бөлшек бөлігіндегі цифрлар санының өзгеруі неге байланысты?

4 жаттығу. Жаңа терезеде жазыңыз:

```
uses crt;
var x,y,a,b:real; s:string;
begin
clrscr;                {1}
readln(x,y);  a:=x/y; {2}
writeln ('x=';x);     {3}
writeln ('y=',y);
writeln ('a=',a);    {4}
readln;  end.
```

Бағдарламаны орындаңыз. Экранға курсор шығады. Бос орынмен ажыратып екі сан енгізіңіз. Экранға шыққан x пен y-тің мәндері енгізілген сандарға ие болды ма?

Бағдарламаны бірнеше рет орындап, нәтижесін жазып алыңыз.

Сұрақ: Readln қандай қызмет атқарады? Ол орындалғанда экран қандай түрде болады? Курсордың орналасуы қалай?

5 жаттығу. Диалогтық программа құру

```
uses crt;
var s, a, b:string; x,y,z:byte;
begin
clrscr;                {1}
Writeln('Сәлем!');
Write('Атың кім?'); Readln(s); Writeln;
Write('Екі сан енгіз '); readln(x,y);
Write('Бір сан енгіз '); readln(z);  Writeln;
Writeln('Үш санның қосындысы = ', x+y+z);  Writeln;
Writeln('Сау бол! ',s); Writeln;
readln;  end.
```

Сұрақ: Write және Writeln айырмашылығы неде?

Өрнектерді бағдарламалау тілінде жазып есептеңіз.

$$1. p = \frac{\sqrt{\lg 15 + 4,5 + ctg^3 3,1}}{\cos 1,04 - 35} \quad \text{Жауабы: } 402.1339$$

$$2. k = \sqrt{\frac{e^{25-12,3} + 121}{50 \cdot 1,7}} - \frac{(458,78 - 48,97) \cdot 1,5}{25,78 + \sqrt[3]{3} \cdot 354} \quad \text{Жауабы: } 62.10672$$

$$3. c = \left| 125 - 6 \cdot \frac{(0,75 + 3,25)^5 + \sin^3 45^\circ}{tg 1,2 - \cos^2 3,5} \right| + 100 - 2^7 \quad \text{Жауабы: } 3472.599$$

$$4. d = \frac{|\cos 45^\circ - 412|}{e^{1,8+7} + 12} \cdot \sqrt{\frac{1525 - 418}{|1254 - 3564| + 255,08}} \quad \text{Жауабы: } 0.0406535$$

Пайдаланылатын әдебиет: [4], 5-24 беттер;

№5 зертханалық жұмыс Тармақты алгоритмдер. Шарттық оператор

Теориялық бөлім

Белгілі бір шартты тексеру нәтижесіне байланысты екі түрлі іс-әрекеттің біреуі ғана орындалатын жағдайда шарттық оператор қолданылады. Шарттық оператордың жазылу форматы:

Толық түрі: **If** <шарт> **then** <1 оператор >
 else <2 оператор >;

Шарттық оператор былай орындалады: Ең алдымен шарттағы логикалық өрнектің нәтижесі анықталады. Егер нәтиже ақиқат болса <оператор1> орындалады, ал нәтиже жалған болса <оператор2> орындалады.

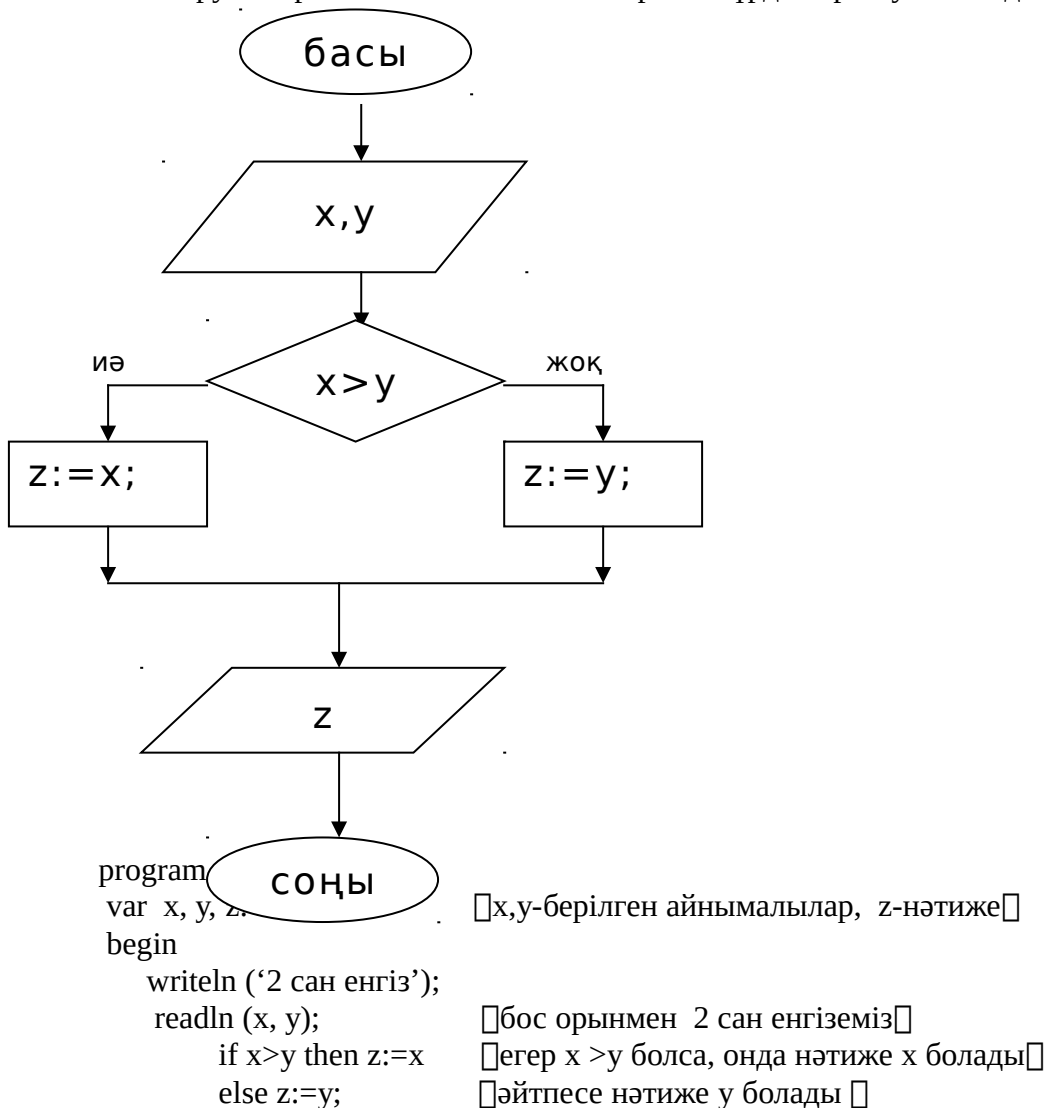
Қысқаша түрі: Шарттық операторда шарт орындалмаған жағдайда еш нәрсе орындау қажет болмаса, else тармағын жазбауға болады:

If <шарт> **then** <оператор>;

Тармақты алгоритмдердің орындалу реті блок-схемада анық көрінеді. Блок-схема дегеніміз алгоритмді геометриялық фигуралар арқылы бейнелеу.

Мысал: экранға екі санның үлкенін шығару керек.

Есепті шығару алгоритмін блок-схемамен көрнекі түрде көрсетуге болады.



```
writeln(z); readln;
end.
```

Бағдарламаның орындалу барысында 5 және 7 сандарын енгіземіз. Айнымалы x-ке 5, у-ке 7 меншіктеледі (x:=5,y:=7). 5>7 шарты орындалмайды, нәтижесі жалған, сондықтан else –ден кейінгі оператор орындалады, ол оператор нәтижеге у-ті меншіктейді. Одан соң z-тің мәні экранға шығарылады. Экранға 7 жазылады.

Мысал: Енгізілген санның [-5;5] аралығында жататындығын анықтау.

Енгізетін санды x деп белгілейміз, типі – real. Егер x саны үшін $x > -5$ және $x < 5$ шарттары бір уақытта орындалатын болса, онда x саны [-5;5] аралығына тиісті болады

```
program aralyk;
var x: integer;
begin
  writeln('x санын енгіз'); readln(x);
  if (x > -5) and (x < 5)
  then writeln('аралықта жатады')
  else writeln('аралықта жатпайды');
  readln;
end.
```

Тапсырма: Енгізілген сан теріс болса, оның таңбасын қарама-қарсыға ауыстыр. Есепті шешу үшін мына шарттық операторды қолдан. If $x < 0$ then $x := -x$;

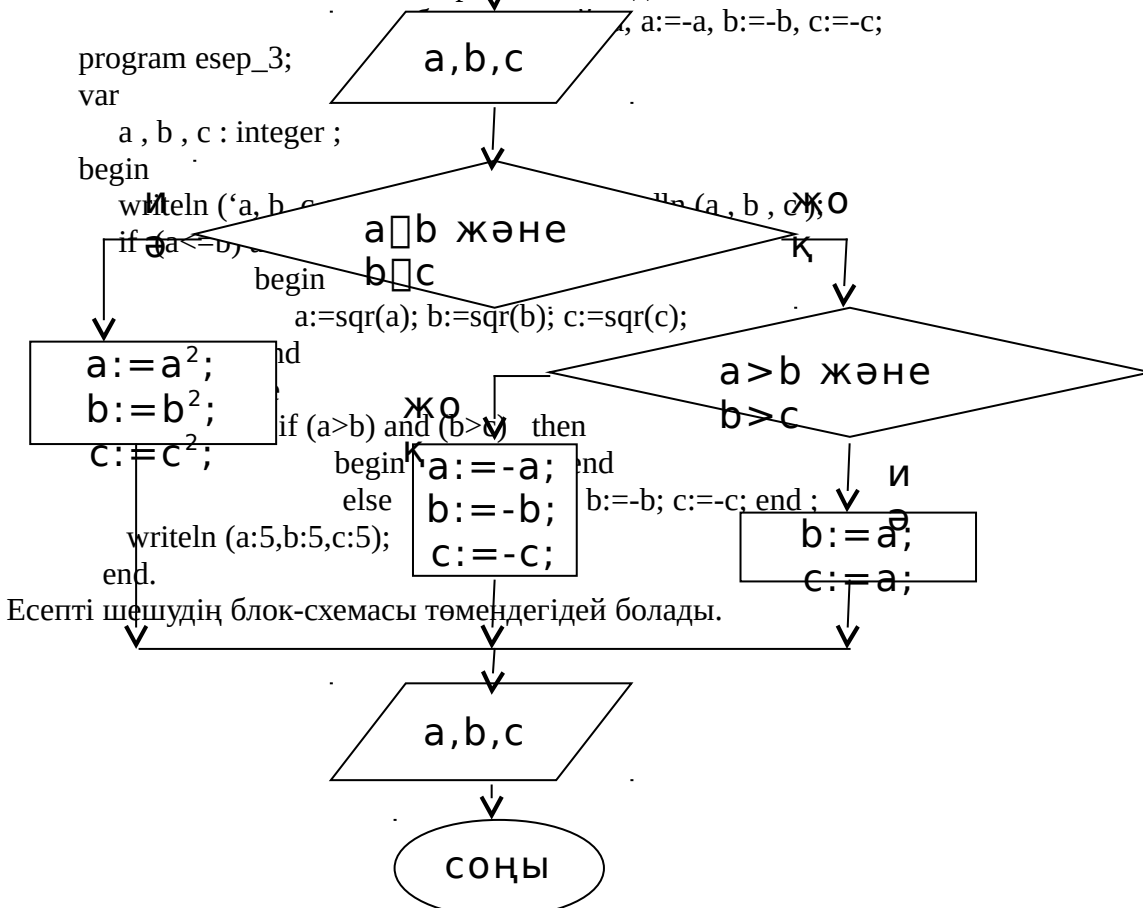
Пайдаланылатын әдебиет: [2], 14-25 беттер; [1], 91-98 беттер;

Қабаттасқан шарттық оператор. Кейбір есептерді шешу кезінде, бірнеше варианттарды қарастыруға тура келеді. Бұл жағдайда, бірнеше шарттық операторлар қолданылады, яғни then, else қызметші сөздерінен кейін, жаңа шарттық оператор жазылады.

Мысал: Бүтін a, b, c сандары берілсін. Егер $a \leq b \leq c$ болса, барлық сандарды өз квадратымен ауыстыр, егер $a > b > c$ болса, ең кіші санды өз квадратымен ауыстыр, егер $a > b > c$ болса, ең үлкенімен, басқа жағдайда, әрбір санның таңбасын өзгертіп, $a := -a, b := -b, c := -c$.

Есептің берілгені бойынша: егер $a \leq b \leq c$ болса, онда $a := a^2, b := b^2, c := c^2$;

басы



Тапсырма: Жоғарыдағы мысалдағы шарттық операторды былай өзгертсек:

```
if (a<=b) and (b<=c) then
begin
  a:=sqr(a); b:=sqr(b); c:=sqr(c);
  if (a>b) and (b>c) then
    begin a:=c; b:=c; end;
  else begin a:=-a; b:=-b; c:=-c; end ;
end.
```

бағдарламаның орындалуы қалай өзгереді? Есептің берілгені қалай болады. Блок-схемасын сыз.

№6 зертханалық жұмыс Таңдау командасы

Мақсаты: Таңдау операторының жазылуы мен пайдаланылуын меңгеру.

Теориялық бөлім

Егер алгоритмде бірнеше жағдайдың біреуін пайдалану қажет болса, онда таңдау операторы қолданылады. Таңдау **case** операторымен программаланады. If операторы екі жағдайдың біреуін орындайды, ал case операторы бірнеше жағдайдың біреуін таңдайды. Жағдайлар таңдау айнымалысының мәніне байланысты таңдалады.

Жазылу форматы:

```
case <таңдау айнымалысы> of
  <1-тізім (мән)>:<оператор 1;>
  <2-тізім (мән)>:<оператор 2;>
  <n-тізім(мән)>:<оператор n;>
  [else <оператор>]
```

end;

Case операторының орындалу тәртібі:

Алдымен <таңдау айнымалысының> мәні анықталады. Анықталған мән қай тізімге жатса, сол тізімге сәйкес келетін оператор орындалады да, басқа операторлар орындалмай, case операторы жұмысын аяқтайды.

Егер <таңдау айнымалысының> мәні тізімнің ешбіреуіне жатпаса, онда else-де көрсетілген <оператор> орындалады. Else болмаса, case операторы жұмысын аяқтайды.

Case операторын пайдалану ережелері:

1. Таңдау айнымалысы мен тізімдегі мәндер бір типті болу керек.
2. Else тармағының болуы міндетті емес, кей жағдайда жазбауға да болады, егер else тармағы жазылса, алдыңғы оператордан соң (;) қойылады.
3. Мәндер тізімінде үтірмен ажыратылған жеке мәндерді немесе бастапқы және соңғы мәндері “..” ажыратылған диапазонды көрсетуге болады.
4. <Операторда> “;”-мен ажыратылған бірнеше операторды жазуға болады.
5. Таңдау айнымалысы литерлік, бульдік және пайдаланушының типтерінде болады.

Ескерту: String, real типтегі айнымалылар таңдау айнымалысы бола алмайды!

№7 зертханалық жұмыс Циклдер.

Теориялық бөлім

Параметрлі қайталану. Қайталанатын әрекеттердің саны алдын-ала белгілі болғанда For операторы пайдаланылады. Бұл оператор **параметрлі қайталану** деп аталады, себебі, қайталану саны **параметр** немесе **басқарушы айнымалы** деп аталатын айнымалының мәніне байланысты болады. Бұл айнымалыда қайталану саны көрсетіледі. Қайталанатын әрекеттер **цикл денесі** деп аталады.

Жазылу форматы:

For <циклдің параметрі>:= <s1> **to** <s2> **do**
<оператор>; { өсу ретімен }

For <циклдің параметрі> := < s2> **downto** < s1> **do** <оператор>;
{ кему ретімен }

Мұндағы: s1, s2 - параметрдің бастапқы және соңғы мәндері;

For ... do - *циклдің тақырыбы*;

<оператор> –*цикл денесі*.

Цикл денесі жай немесе құрама оператор болуы мүмкін.

For операторы мынаны анықтайды:

– параметрдің өзгеру аралығын және цикл денесінің қайталану санын;

– параметр мәнінің өзгеруін (**to-өсу, downto-кему**);

FOR операторы параметр барлық мәндеріне ие болып біткенше, цикл денесінің орындалуын қайталауды тоқтатпайды.

Егер $s2 > s1$ болса, цикл денесі $(s2 - s1 + 1)$ рет орындалады.

Егер $s1 > s2$ болса, цикл денесі орындалмайды.

Алғы шартты цикл. Егер іс-әрекеттің қайталану саны белгісіз, бірақ қайталану шарты белгілі болса, онда *while* немесе *repeat* операторлары қолданылады.

While (әзір) операторында қайталану шарты цикл денесінен бұрын тексеріледі. Сондықтан *while* операторы **алғы шартты цикл** деп аталады.

Жазылу форматы:

While <қайталану шарты> **do**
<цикл денесі>

Қайталану шарты – бульдік (логикалық) өрнек, цикл денесі - жай немесе құрама

оператор. Цикл денесі орындалудан бұрын шарттағы өрнектің мәні анықталады. Егер ол мән true болса, цикл денесі орындалады. Шарттық өрнектің мәні тағы да анықталады, егер нәтижесі false болса циклдың жұмысы аяқталып, while-ден кейінгі бірінші оператор орындалады.

Кейінгі шартты цикл. Циклдің бұл түрінде:

- қайталану шарты цикл денесінен кейін тексеріледі, сондықтан **кейінгі шартты цикл** деп аталады.
- қайталанатын іс-әрекеттер кем дегенде 1 рет орындалады.
- шарттың нәтижесі true болғанда циклдің орындалуы тоқтайды.

Жазылу форматы: **repeat**

```
<оператор;>
  :
< оператор;>      {цикл денесі}
until <циклді аяқтау шарты>;
```

Repeat және **until** сөздерінің арасындағы операторлар *цикл денесі* болады.

Кейінгі шартты циклдің орындалу тәртібі: алдымен цикл денесі орындалады, одан соң циклден шығу шарты тексеріледі. Егер нәтиже false болса, цикл денесі тағы да орындалады, егер true болса, циклден шығады.

1-мысал: 999 саны енгізілгенше бүтін сандарды енгізе отырып, олардың қосындысын табу.

```
...
x: integer; sum: real;
begin
  sum:=0;
  repeat
    write ('x-ті енгіз'); readln(x);
    if x<> 999
      then sum:=sum+x;
  until x=999;
  writeln ('сандардың қосындысы=',sum); readln;
end.
```

Айнымалыларға түсініктеме:

x-енгізілетін бүтін сандар; sum-олардың қосындысы (real);

Бағдарламаның басында қосынды 0-ге теңестіріледі. Одан соң repeat сөзімен цикл ашылады. Цикл денесінде “x-ті енгіз” сұрауымен x айнымалының мәні енгізіледі. If операторы ол санның 999-ге тең емес екендігін тексереді. Егер тең болмаса, Sum қосындысының мәніне x саны қосылады. Циклдің соңындағы until x=999 циклдің аяқталу шартын тексереді. Егер x=999 болса, цикл аяқталады, until-ден кейінгі оператор орындалады. Ол оператор writeln ('сандардың қосындысы=',Sum); нәтижені шығарады.

№8 зертханалық жұмыс Жолдар

Жол дегеніміз – ұзындығы 255-тен артпайтын символдар тізбегі. Тізбектің мағынасы болуы міндет емес. Мысал: 'df56', '*d-шар56', '*-4 лд'

Жолдар *string* жолдық типпен сипатталады. Жолдық типті анықтағанда ондағы символдар санын көрсетуге болады.

Жазылу форматы:

type

<типтің аты>=string [символдар саны];

var <идентификатор>: <типтің аты>;

String типтегі айнымалыны типті алдын-ала сипаттамай-ақ көрсетуге болады:

var

<идентификатор>:string[жолдың ұзындығы];

N символдан тұратын жолға жадыдан N+1 байт бөлінеді. N байт-символдарды сақтау үшін, ал бір байт – жолдың ұзындығын сақтау үшін.

Жолдық өрнектер. Олар жолдық тұрақтылардан, айнымалылардан, функциялардан және операция таңбаларынан тұрады. Мысал: 'ма'+ 'ма'

Жолдық процедуралар мен функциялар

Аты, жазылуы	Қызметі	Ескерту
1. жою delete(a,p,n)	a жолындағы p позициядан бастап, ұзындығы n символды жояды.	p<=255; нәтиже басқа айнымалыға меншіктелмейді.
2. кіргізу insert (a,s,p)	a жолын s жолына p позициядан бастап кіргізеді.	нәтиже басқа айнымалыға меншіктелмейді.
3. типті ауыстыру str(x,a)	x сандық шаманы жолға өңдеп, a-ға меншіктейді	x-ті шығару форматымен жазуға болады. X сандық типте, a – жолдық типте.
4. типті ауыстыру val(a,x,c)	a жолын сандық шамаға өңдеп, x айнымалыға орналастырады. A жолында бос символ болмау керек.	c-өңдеу нәтижесі, бүтін сан. Егер өңдеуде қате болмаса c=0 болады. A:string, x сандық типте, c:integer.
5.ұзындық length(a)	a жолының ұзындығын табады.	Нәтижені айнымалыға меншіктеуге болады.
6. ретімен тіркестіру concat(a,b,..s)	a,b,..s жолдарын сол ретімен тіркестіреді	-
7. белгілеу copy (a,p,n)	a жолынан p позициядан бастап, ұзындығы n символды белгілейді.	егер p>length(a) болса, нәтижесі бос символ; p>255 болса, қате.
8. позиция pos (a,s)	a жолы s жолында нешінші позицияда тұрғанын табады.	егер a жолы s жолында болмаса, нәтиже=0
9.регистрді ауыстыру upcase(ch)	кіші әріпті бас әріпке өзгертеді.	тек латын алфавитін ғана.

Мысалдар

Берілгені:	Қолданылған процедура, функция	Нәтиже
1.a:='абвгде'	delete(a,4,2)	'абве'
2.s1:='интика' s2:='форма'	insert(s2,s1,3)	'информатика'
1.var x:integer; a:string; ----- x:=72584	str(x,a) str(-x:7,a)	'72587' '-72584'
2.var a:string; cod:integer; ----- a:='25'; a:='14.2E+2';	val(a,x,cod) val(a,x,cod)	cod=0 cod=0

a:='14.2'	val(a,x,cod)	cod=5
3.st:='1237'	length(st)	4
st:='klassio'	length(st)	7
4.a1:=' ab'; a2:='cd'; a3:='ej'	concat(a1,a2,a3) concat(a1,'nm',a2)	'abcdej' 'abnmcd'
5.st:='abcdefjk'	copy(st,2,4) copy(st,5,7)	'bcde' 'efjk'
8.a1:='abcdef'; a2:='def';	pos(a2,a1) pos('e',a2) pos('k',a1)	4 2 0
9.Ch:='d' A:='x'	UpCase(Ch) UpCase(A) UpCase('a')	'D' 'X' 'A'

1-мысал. Енгізілген сөздегі 'а' әріптерін санау, 'b' әрпіне ауыстыру.

1- әдіс: Алгоритм:

- 1.Сөзді енгізу.
- 2.Сөздің бірінші әрпін белгілеу.
- 3.Белгіленген символды 'а' символымен салыстыру.
- 4.Егер сәйкес болса, санауышты бірге арттыру, сол символдың орнына 'b' символын қою;
- 5.Осылайша барлық символдарды қарастыру.

Бағдарлама.

```

program sanaу;
var s:string; n,l,i:byte;      {n-'a' әрпін санауыш }
begin
  n:=0; readln(s); l:=length(s);  { l сөздің ұзындығы}
  for i:= 1 to l do
    if copy(s,i,1)='a'          { кезектегі әріпті белгілеп,
      then                      'a' әрпімен салыстыру}
      begin
        delete (s,i,1); insert('b',s,i);
        n:=n+1;  { a әрпін санау}
      end;
  writeln ('сөзде, 'n, ' a әрпі бар');
end.

```

2-әдіс. Алгоритм:

- 1.Сөзді енгізу.
2. 'a' әрпін санайтын санауыш енгізу.
- 3.Сөздегі 'a' әрпі тұрған позицияны тауып, сол орынға 'b' әрпін жазу.
- 4.Сөздегі барлық позициялар үшін 3 пунктті қайталау.
- 5.Нәтижені шығару.

Бағдарлама.

```

program sanaу;
var
  s:string; n:byte;
begin
  write ('сөзді енгіз'); readln(s);  n:=0;
  while pos ('a',s)>0 do
    begin
      n:=n+1; s[pos('a',s)]:='b';
    end;
end;

```

```
writeln ('сөзде',n,' а әрпі бар');  
end.
```

№9 зертханалық жұмыс Процедуралар

Процедураның сипаттамасы **процедураның тақырыбынан** және **денесінен** тұрады.

Процедураның тақырыбы **procedure** деген резервтелген сөзден, процедураның атын білдіретін **идентификатордан** және жақшаға алынып, типтері қарастырылған **формальді параметрлер тізімінен** тұрады. Процедура денесі бағдарламалық блоктан тұрады.

Процедураның жалпы тәрізі:

```
procedure <аты>[(формальді параметрлер тізімі)];  
    сипаттама бөлімі  
begin  
    операторлар бөлімі  
end;
```

Процедура мүддігінен орындалмайды. Ол аты бойынша негізгі бағдарламадан шақырылады. Шақыру жолында оның фактілік (нақты) параметрлері қарастырылады.

Фактілік және формальді параметрлердің арасында мынадай сәйкестік болуы керек:

- саны бірдей;
- типтері бірдей;
- жазылу реті бірдей;

Формальді параметрлердің мынандай тәріздері болады:

- параметр – мендер;
- параметр – айнымалылар;

Параметр – мендер. Параметр – мендер негізгі бағдарламадан ішкі бағдарламаға мендерді беру үшін ғана қолданылады, процедура нәтижесі негізгі бағдарламаға қайтарылмайды. Параметр – мендер фактілік мендерге ешқандай ықпал жасамайды.

1-мысал: түрт санды екі-екіден жинап, квадраттарының қосындысын табу.

Бұл мысалда параметр – мендер қолданылады. Берілген сандар 2,5 пен 3,1; -7,2 және 5,3 болсын.

```
program mander;  
var k, z, x, y: real;  
procedure sum_kv(a, b:real);      {a, b формальді параметрлер}  
begin                            {процедура денесі}  
    a:=a*a;  b:=b*b;  
    writeln('квадраттар қосындысы=', a+b);  
end;  
begin                            {негізгі бағдарлама денесі}  
    x:=2.5; y:=3.1;  
    sum_kv(x,y);  
    z:= -7.2; k:=5.3;  
    sum_kv(z,k);  
    readln;  
end.
```

Бағдарламаға тәсілігі

X, y, z, k - негізгі бағдарламадағы фактілік параметрлер. Олар негізгі бағдарламада сипатталады.

Sum_Kv процедурасында екі санның квадраттарының қосындысы экранға шығарылады; a мен b - процедурадағы формальді параметрлер.

Процедура x, y параметрлерімен шақырылғанда $a=2.5$ және $b=3.1$ мәндеріне ие болады да, олардың квадраттарының қосындысы есептеледі.

Экранға мынандай хабарлама шығады:

2.5 және 3.1 квадраттарының қосындысы

Бағдарлама жұмысы процедурадан шығып, негізгі бағдарламаның орындалуы жалғасады. z пен k -ның мәндері меншіктеледі. Процедура енді z, k параметрлерімен шақырылады.

$A=-7.2, b=5.3$ мәндеріне ие болады. Енді осы екі санның квадраттарының қосындысы есептеледі.

Экранға мынандай хабарлама шығады :

-7.2 және 5.3 квадраттарының қосындысы

процедура жұмысы аяқталып, негізгі бағдарламаға оралады, негізгі бағдарлама жұмысын аяқтайды.

Бағдарламаның орындалу тәртібі

1. Негізгі бағдарламадағы айнымалылар сипатталады.
2. X, Y параметрлерімен Sum_Kv процедурасы шақырылып, процедура орындалады, болған соң негізгі бағдарлама жалғасады.
3. Z, K параметрлерімен Sum_Kv процедурасы шақырылып, процедура орындалады, болған соң негізгі бағдарламаға оралады.
4. Негізгі бағдарлама жұмысын аяқтайды.

Параметр–айнымалы. Параметр – айнымалылар процедураның нәтижесін негізгі бағдарламаға екелу (қайтару) %шiн қолданылады.

Параметр–айнымалылар негізгі бағдарламадағы фактілік параметрлерге ықпал етіп, оларды өзгерте алады.

2-мысал. Санның дәрежесін табуы процедура етіп алып, $y=a_4*x^4+a_3*x^3+a_2*x^2$ мәнін есептеу бағдарламасын құру. Мұндағы, a_4, a_3, a_2, x – клавиатурадан енгізіледі.

```
program kosindi;
var x, a4, a3, a2, y, s,:real
procedure dareje (a: real; n: byte; var d: real);
  var i:byte;
  begin {процедура денесі}
    d:=1;
    for i:=1 to n do
      d:= d * a
    end;
begin {негізгі бағдарламаның денесі}
  readln(x, a4, a3, a2);
  dareje(x,4,s); y:=s*a4;
  dareje(x,3,s); y:= y +s*a3; dareje(x,2,s); y:=y+s*a2;
  writeln ('y=', y); readln;
end.
```

Бағдарламадағы процедураға сипаттама. Процедура Dareje деп аталады. Онда a санының n дәрежесі есептеліп, нәтижесі d -ға меншіктеледі. A, n, d - формальдық параметрлер. Дәреженің нәтижесі d негізгі бағдарламаға қайтарылатындықтан var d:real сипаттамасы жазылады. i -формальдық параметр емес, сондықтан ол процедураның сипаттама бiлiмiнде жазылған.

Процедура денесінде a санының n рет қубейтiндiсi есептеледi, a мен n -нiң мәндерi негiзгi бағдарламадан берiледi.

Негiзгi бағдарламаға сипаттама. Негiзгi бағдарламада қолданылатын фактiлiк

параметрлер: x, a4, a3, a2, y, s; y-нетиже, ал s-санның дәрежесінің нетижесі.

x, a4, a3, a2-мендері клавиатурадан енгізіледі.

x4, s-параметрлерімен дәреже процедурасы шақырылады.

Процедурадағы a=x, n=4 мендеріне ие болып, x-тің 4 дәрежесі есептеледі. Нетижесі S параметрімен негізгі бағдарламаға қайтарылады, яғни s:=d болады.

Негізгі бағдарламада y:=s*a4 мені есептеледі.

x,3,s параметрлерімен процедура тағы да шақырылады, яғни a=x, n=3 мендерінде x-тің 3 дәрежесі есептеледі. Нетижесі, яғни d-ның мені s-ке меншіктеледі, s жаңа менге ие болады да, негізгі бағдарламада y:=y+s*a3 есептеледі. x, 2, s параметрлерімен процедура шақырылып, іс-ерекет қайталаңады.

№10-11 зертханалық жұмыс. Функциялар.

Теориялық бөлім

Паскаль тілінде пайдаланушы стандарттық функциялардан басқа, өз функциясын анықтай алады. Ондай функциялардың сипаттамасы функцияның тақырыбынан және функция денесінен тұрады.

Функцияның жазылуының жалпы тәрізі:

Function аты(формальді параметрлер тізімі): нетиженің типі;

сипаттама бұлмі

begin

функция денесі

end;

Функция негізгі бағдарламадан аты бойынша шақырылады. Функцияға берілетін мендер, ондағы формальды параметрге сәйкес болу керек. Функцияның тақырыбында функцияның аты және типтері қирсетілген формальды параметрлер тізімі жазылады. Тізім жақшаға алынады. Жақшаның сыртында функция нетижесінің типі жазылады.

Функция денесінде кем дегенде бір меншіктеу операторы функцияның атына мен меншіктейтін болу керек!

3-мысал. Санның дәрежесін табуды функция етіп алып, $z=(a^5+a^{-3}) / (2 \cdot a^m)$ – өрнегінің мәнін есептеу.

```
program z_funk;
```

```
var m: integer; a, z, r: real;
```

```
function dareje (n: integer; x : real): real;
```

```
var i: integer; y: real;
```

```
begin
```

```
  y:=1;
```

```
  for i:= 1 to n do
```

```
    y:= y*x; dareje:= y;
```

```
end;
```

```
begin
```

```
  readln (a, m);
```

```
  z:= dareje (5,a);
```

```
z:= z+ dareje(3,1/a);
```

```
if m=0 then r:=1
```

```
  else if m>0 then r:= dareje (m,a)
```

```
    else r:=dareje (m,1/a);
```

```
z:= z/(2*r);
```

```
writeln ('a=',a, 'm=':10, m, 'z=':10, z);
```

```
readln; end.
```

Бағдарламадағы функцияға сипаттама. Функцияның аты dareje деп аталады. Бұл функция x^n дәрежесін есептейді. N мен x - функцияның формальды параметрлері: n - дәреже көрсеткіші, типі integer; ал x n дәрежеге шығарылатын сан, типі real. Негізгі пограммаға қайтарылатын функцияның нәтижесі де нақты типті.

Функцияда формальды параметрлерден басқа, i және y айнымалылары сипатталған. i - қайталану санын білдіреді.

Y айнымалысы санды өз-өзіне i рет көбейтудің нәтижесін сақтайды. N -дәреже есептеліп болған соң, dareje функциясына y -тің мәні меншіктеледі.

Негізгі бағдарламаға сипаттама. Негізгі бағдарламада m, a, z, r айнымалылары сипатталады. m - дәреже көрсеткіші; a - берілген сан; r - a^n дәрежесінің нәтижесі; z - нәтиже.

Бағдарлама орындалғанда a және m мәндері клавиатурадан енгізіледі. $z := \text{dareje}(5, a)$ жолында 5 және a фактілік параметрлерімен dareje функциясы шақырылады.

Функциядағы n және x параметрлері сәйкес мәндерге ие болады, яғни $n := 5$, $x := a$. a^5 есептеліп, нәтижесі dareje айнымалысына меншіктеледі. Негізгі бағдарламаға оралған соң, функцияның мәні z айнымалысына меншіктеледі.

$Z := z + \text{dareje}(3, 1/a)$ жолы орындалғанда функцияның формальды параметрлері мына мәндерге ие болады. $n := 3$, $x := 1/a$; себебі, $(a^{-n}) = (1/a^n)$.

Функцияның орындалуы алдыңғыдай. Негізгі бағдарлама орындалған соң, z мәніне dareje мәні қосылады. Шарттық операторда n -нің мәні тексеріледі. Соның нәтижесіне байланысты r -ді анықтайды. Атап айтқанда, егер $m = 0$ болса, онда $r := 1$;

егер $m > 0$ болса, онда $r := a^m$;

егер $m < 0$ болса, онда $r := (1/a^m)$;

m, a параметрлерімен dareje функциясы шақырылады, нәтижесі r -ге меншіктеледі. $z := z / (2 * r)$ жолында $z / (2 * r)$ нәтижесі z -ке меншіктеледі.

Нәтиже экранға шығарылады.

Функциясы бар бағдарламаның орындалу тәртібі

1. Негізгі бағдарламадан фактілік параметрмен функция шақырылады.
2. Функциядағы формальді параметрлер фактілік мәндерге ие болады.
3. Функцияның нәтижесі анықталады, функцияның атына мен меншіктеледі.
4. Негізгі бағдарламада функцияның мәні пайдаланылады.

Пайдаланылатын әдебиет: [2], 57-64 беттер; [1], 130-157 беттер;

№7 тақырып. Ақпаратты компьютердің сыртқы құрылғыларында көрсету

№12-13 зертханалық жұмыс. Файлдар

Теориялық бөлім

Өте үлкен көлемді ақпараттарды сыртқы жадыда сақтау ыңғайлы. Мысалы, оқу орнындағы студенттер туралы, кітапханадағы кітаптар туралы, т.с.с. мәліметтер. Бұл ақпараттар бағдарламада файлдар арқылы пайдаланылады.

Файл дегеніміз сыртқы жадыда белгілі бір атпен сақталған деректердің жиыны.

Мәліметтерді файлдармен пайдаланудың себептері:

1. Бағдарламаның жұмыс барысында өте үлкен ақпаратты енгізу көп уақытты алады және адамды жалықтырады. Клавиатурадан енгізілген деректер мен экранға шығарылған нәтижелер сақталмайды, бағдарламаның жұмысы аяқталған соң жоғалып кетеді. Сондықтан, бұл ақпараттар алдын-ала дайындалып, дискіде сақталады да, қажетінше

пайдаланыла беріледі.

2. Берілгендер файлын басқа бағдарламамен дайындап бірнеше бағдарламаны бір-бірімен байланыстыруға болады.

3. Бағдарламаның орындалу кезінде пайдаланушының қатысуы міндет емес.

Файлдық тип. Бір бағдарламада бірнеше файлмен жұмыс істеуге болады. Әрбір файл өз атымен аталады. Файлдағы компоненттер бір типте болады. Файлдың ұзындығы алдын-ала анықталмайды, ол құрылғының сыйымдылығына байланысты болады. Файлды бір типтегі мәндердің шексіз тізімі деп қарастыруға болады. Файлдың элементтері нольден бастап нөмірленеді. Файлдың элементтері **ағымдағы көрсеткіш** арқылы көрсетіліп тұрады. **Ағымдағы көрсеткіш** бағдарлама жұмысына байланысты бір элементтен екіншіге ауысып тұрады. Кез келген уақытта файлдың бір элементіне ғана қол жеткізуге болады.

1 элемент	2 элемент	3 элемент	4 элемент	...
-----------	-----------	-----------	-----------	-----



ағымдағы
көрсеткіш

Файлдардың элементтерін *тізбекті* (последовательный) немесе *тікелей* (прямой) қарастыруға болады. Тізбекті файлдың элементтеріне *жазылу реті* бойынша қол жеткізіледі. Тікелей қол жетімді файлдың элементтеріне олардың *адресі* бойынша қол жеткізіледі. Сондықтан, *тікелей* файлдың кез келген элементін кез келген уақытта пайдалануға болады.

Дискідегі деректер файлы Паскаль бағдарламасымен файлдық айнымалы арқылы байланысады. Бағдарламада көпшілік жағдайда файлдық айнымалыны **f** арқылы белгілейді.

Бағдарламаның айнымалыны сипаттау бөлімінде файлдық айнымалы былай сипатталады.

var

файлдық айнымалы: **file of** элементтердің типі;

Мысал:

var

f1, f2:file of integer; f1, f2 - элементтері бүтін типтегі файл.

s1, s2:file of string; s1, s2- элементтері жолдық типтегі файл.

Файлдарды пайдалану әдістері.

1. Бағдарламаның басында файл мен файлдық айнымалыны байланыстыру қажет. Ол үшін мына процедура қолданылады:

assign(файлдық айнымалы, деректік файлдың аты);

2. Файлмен жұмыс істеу үшін алдымен оны ашу қажет. Файлды пайдалану мақсатына қарай ашудың екі түрі бар:

2.1. Файлдан деректерді оқу үшін –

Reset(файлдық айнымалы);

2.2. Жаңа файлды жасау үшін және оған деректерді жазу үшін

Rewrite(файлдық айнымалы);

3. Ашылған файлдың элементін оқу:

Read(файлдық айнымалы, айнымалы);

4. Ашылған файлға элемент жазу:

Write(файлдық айнымалы, айнымалы);

5. Файлмен жұмыс аяқталған соң файл жабылады:

Close(файлдық айнымалы);

1-мысал: Санды клавиатурадан енгізіп, san.txt файлына жаз.

program file_tip;

var f:file of integer; s:integer;

begin

assign(f1, 'san.txt'); { san.txt файлы f1 айнымалысымен байланыстырылады. }


```

rewrite(f1);          { f1 файлы деректерді жазу үшін ашылады}
readln(s); write(f1, s); { s айнымалысының мәні f1 файлына жазылады}
close(f1); { пайдаланылып болған соң, f1 файлы жабылады}
readln; end.

```

2-мысал: Деректерді san.txt файлынан оқып, экранға шығар.

Бағдарламаны жазудан бұрын san.txt файлы алдын-ала дайындалады.

```

program file_tip;
var f1:file of integer ; s, n:integer;
begin
  assign(f1, 'san.txt'); { san.txt файлы f1 айнымалысымен байланысты}
  reset(f1);           { f1 файлы деректерді оқу үшін ашылады}
  read(f1, s);         { f1 файлынан кезектегі элемент оқылып, s айнымалысына беріледі. }
  writeln(s);          {s айнымалысының мәні экранға шығарылады }
  close(f1);           {пайдаланылып болған соң, f1 файлы жабылады}
  readln;
end.

```

№8 тақырып. Графиктік операторлармен жұмыс

№14-15 зертханалық жұмыс. Графиканы бағдарламалау

Теориялық бөлім

Паскаль тілінде графикалық кескіндер жасау үшін, GRAPH модулі қолданылады. Бұл модульде 79 графикалық процедуралар, функциялар, тұрақты шамалар мен типтер орналасқан.

Графиканы жұмысқа қосу мына әрекеттерден тұрады:

uses Graph; {Graph модулін, яғни графикалық процедуралар, орналасқан кітапхананы іске қосу}

var dv, mv:integer; {dv, mv – екі айнымалысы графикалық режимді іске қосады}

dv:=Detect; {detect мәнімен қажет графикалық драйвер мен режим автоматты түрде іске қосылады}

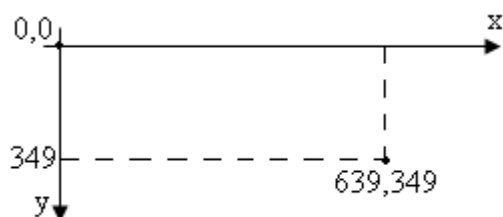
InitGraph(dv,mv,'c:\TP7\BGI'); {graph модулінің орналасқан жолы көрсетіледі}

If GraphResult<>grOk then Halt(1); {Графикалық режимді іске қосудағы қатесі тексеріледі}

Осы әрекеттерден соң графикалық операторлар жазылады. Графикалық әрекеттер орындалып болған соң, графикалық режимді жабу керек.

6.CloseGraph; {Графикалық режимді жабу}.

Монитор экраны нүктелер жиынынан тұрады. Графикалық экранда координаталар жазықтығының орналасуы төмендегідей:



Түстер таблицасы

Түстің номері	Түстің аты	Түстің номері	Түстің аты

0	Қара	8	Қою
1	Көк	9	Көгілдір
2	Жасыл	10	Ашық жасыл
3	Бирюза	11	Ашық бирюза
4	Қызыл	12	Қызғылт
5	Малина	13	Ашық малина
6	Қоңыр	14	Сары
7	Ашық сұр	15	Ақ

Графика элементі	Паскальда жазылуы	Параметрлеріне сипаттама
1	2	3
Нүкте	PutPixel(x,y,t)	x,y-нүктенің координатасы; t-түсі, санмен беріледі.
Кесінді	Line(x1,y1,x2,y2)	(x1,y1),(x2,y2) ұштарының координаталары
Сызық	LineTo(x,y)	x,y нүктелеріне дейін сызады
Тік төртбұрыш	Rectangle(x1,y1,x2,y2)	(x1,y1),(x2,y2)диагональдың координаталары.
Боялған төртбұрыш	Bar(x1,y1,x2,y2)	(x1,y1),(x2,y2)диагональдың координаталары.
Шеңбер	Circle(x,y,r)	x,y-центрдің координатасы; r-радиустың ұзындығы.
Эллипс	Ellipse(x,y,b,s,rx,ry)	b,s-эллипстік доғаның басы және соңы. b =0, s=360 эллипс салады. rx,ry – x және y бойынша радиустар.
Доға	Arc(x,y,b,s,r)	x,y,b,s,r- жоғарыда
Сектор	sector(x,y,b,s,rx,ry)	x,y, b,s, rx,ry - жоғарыда
Сызықтың түсі	SetColor(t)	t - түс номері, кестеде келтірілген.
Бояудың түсі	SetFillStyle(t1,t2)	t1 - бояу стилінің номері, t2-бояудың түсі.
Текст шығару	Outtext('текст')	Тексті экранның сол жақ шетіне шығарады.
Текст шығару	OutTextXY(x,y,'текст')	Тексті көрсетілген координатаға шығарады
Көпбұрыш салу	DrawPoly(n,pp)	n -нүктелер саны;
Тұйық ай-мақты бояу	FillPoly(t, pp)	t-бояудың түсі

Суретті бағдарламалау мысалы:

<pre> program suret; uses graph; var dv, mv:integer; begin dv:=detect; initgraph(dv, mv, 'c:\tp7\bgi'); if graphresult<>grok then halt(1); setcolor(5); rectangle(10,15,610,430); circle(350,100,50); setfillstyle(1,6); bar(100,250,200,300); readln; closegraph; end. </pre>	<pre> program kopburish; uses graph; var pp:array[1..5] of PointType; dv,mv,l,x1,y1:integer; begin dv:=detect; initgraph(dv,mv, 'c:\ tp7\bgi'); pp[1].x:=300; pp[1].y:=50; pp[2].x:=400; pp[2].y:=50; pp[3].x:=350; pp[3].y:=150; pp[4].x:=150; pp[4].y:=200; pp[5]:=pp[1]; DrawPoly(5,pp);FillPoly(3,pp); readln; closegraph; </pre>
--	---

Пайдаланылатын әдебиет: [2], 101-104 беттер; [1], 336-408 беттер;

3 Әдебиеттер тізімі

Негізгі

1. В.Б.Попов TurboPascal 7.0. Учебное пособие. М. изд. «Финансы и статистика». 1996
2. Нұрғазина Б.Қ. Бағдарламалау негіздері. Турбо Паскаль, Павлодар, ТОО НПФ «ЭКО», 2007ж
3. Айтов Ж.А., Мақамбаев М.Б. Паскаль тілінде программалау. 1998
4. Есжанов Н.Т. Алгоритмдер. Алматы, 1998ж
5. Абрамов В.Г. Введение в язык Паскаль. М. Наука. 1988
6. Климова Л. М. Практическое программирование. Решение типовых задач в Pascal 7.0. М. Кудиц-образ. 2000.

Қосымша

7. Грогно П. Программное обеспечение персональных ЭВМ. М. Мир. 1982
8. Йенсен К., Вирт Н. Паскаль: Руководство для использования и описание языка. М. Финансы и статистика. 1982.
9. Марченко А.И., Марченко Л.А. Программирование в среде TurboPascal. Киев. «Век» 1999
10. Цейл Дейл. Программирование на СИ. ДМК. 2000
11. Ян Борецкий. Турбо-Паскаль с графикой для персональных компьютеров М. Машиностроение. 1991.

