



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

"АЛГОРИТМДЕНДІРУ ЖӘНЕ ПРОГРАММАЛАУ"

пәні бойынша зертханалық жұмыстарды орындауға арналған әдістемелік
нұсқаулар

Павлодар



БЕКІТЕМІН
ФМж/еАТ факультетінің деканы
_____ С.К.Тлеуқенов
200 __ ж. «__»_____

Құрастырушы: Аға оқытушы Нұрғазина Б.Қ.

Информатика және ақпараттық жүйелер кафедрасы

050716 - «Приборлар жасау» мамандығының студенттері үшін

"Алгоритмдендіру және 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.

Тақырып 2. ЭЕМ-да есептерді шығарудың кезеңдері

№1 зертханалық жұмыс. Алгоритмдерді сипаттау (ұсыну) құралдары мен тілдері.

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 Esc 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_orota;
```

```
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 memory –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.

№2 зертханалық жұмыс. Алгоритмдердің базалық құрылымын жасау әдістері.

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 $	<code>abs(x)</code>	бүтін, нақты
x^2	<code>sqr(x)</code>	бүтін, нақты
\sqrt{x}	<code>sqrt(x)</code>	нақты
$\sin x$	<code>sin(x)</code>	нақты
$\cos x$	<code>cos(x)</code>	нақты
$\arctg(x)$ $\arctan(x)$	нақты	
e^x	<code>exp(x)</code>	нақты
$\ln x$	<code>ln(x)</code>	нақты
$\lg x$	<code>ln(10)/ln(x)</code>	нақты
$\pi=3,141592..$	<code>Pi</code>	нақты

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:=(sqrt(sqrt(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.

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

өрнегінің мәнін есептеу.

4. Z:= жолына мынаны теріңіз:

```
Z:=abs(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.

$$z = \frac{3e^5 y^5 + \sqrt{\cos^2 y + \sin y^2}}{\sqrt[3]{(x+y)^2}}$$

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. e⁵ және y⁵ екеуінде де дәреже көрсеткіші 5, бірақ жазылуы әр түрлі. Неге?

7 ...sin (sqrt(y)))) жолдарында неге 4 рет))) жақша қойылған?

№3 зертханалық жұмыс. Алгоритмдік тілдердің негізгі конструкциялары.

Шығару, енгізу процедуралары

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

Теориялық бөлім

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

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

Жазылу түрлері:

writeln (өрнек); – өрнектің мәнін экранға шығарады. `writeln (sqr(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. Эcranға 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}

```


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

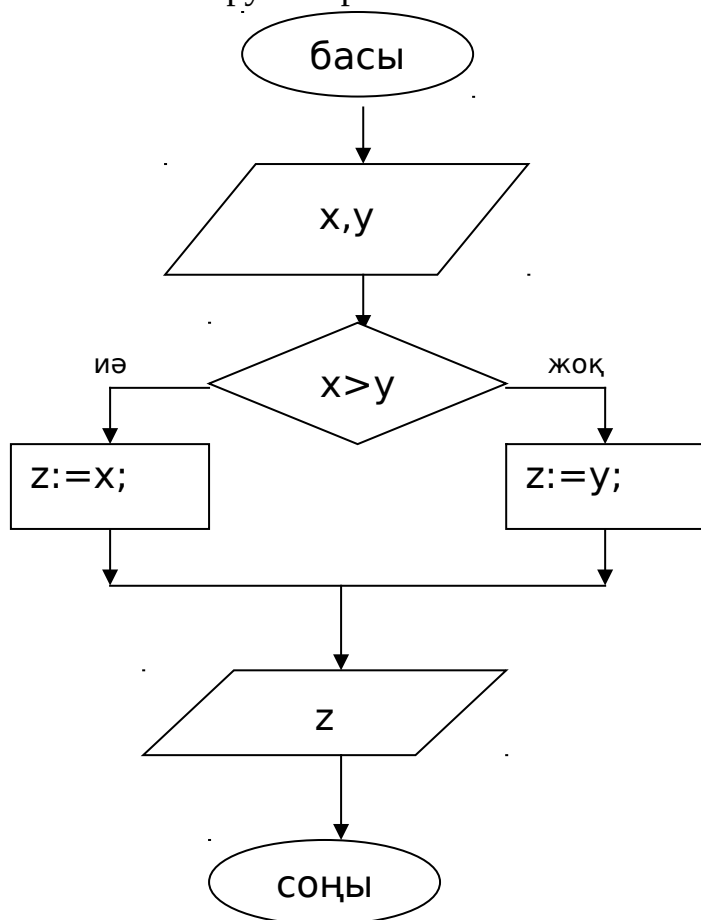
Қысқаша түрі: Шарттық операторда шарт орындалмаған жағдайда еш нәрсе орындау қажет болмаса, else тармағын жазбауға болады:

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.



```

program esep_1;
var x, y, z: real;
begin
  writeln ('2 сан енгіз');
  readln (x, y);
  if x > y then z := x
  else z := y;
  writeln (z); readln;
end.

```

□x,y-берілген айнымалылар, z-нәтиже□
□бос орынмен 2 сан енгіземіз□
□егер x > y болса, онда нәтиже x болады□
□әйтпесе нәтиже y болады □

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. $\text{If } x<0 \text{ then } x:=-x;$

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.

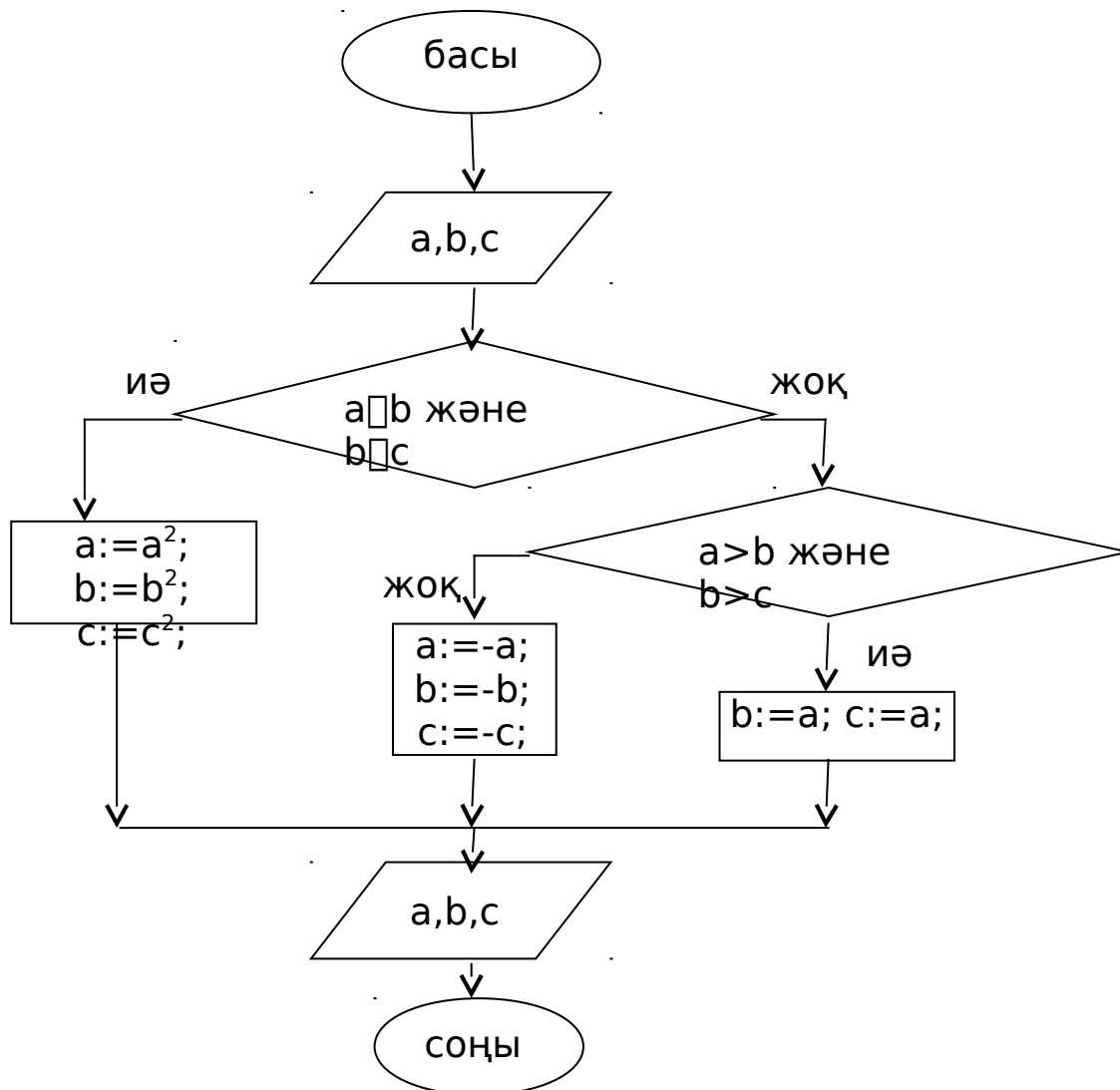
Мысал: Бүтін a, b, c сандары берілген. Егер $a \leq b \leq c$ болса, барлық сандарды өз квадратымен ауыстыр, егер $a > b > c$ болса, әрбір санды үшеуінің ішіндегі ең үлкенімен, басқа жағдайда, әрбір санның таңбасын өзгерт.

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

```
program esep_3;
var
  a , b , c : integer ;
begin
  writeln ('a, b ,c сандарын енгіз');   readln (a , b , c );
  if (a<=b) and (b<=c) then
    begin
      a:=sqr(a); b:=sqr(b); c:=sqr(c);
    end
  else
    if (a>b) and (b>c) then
      begin c:=a; b:=a; end
    else begin a:=-a; b:=-b; c:=-c; end ;
end.
```

```
writeln (a:5,b:5,c:5);  readln;
end.
```

Есепті шешудің блок-схемасы төмендегідей болады.



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.

№5 зертханалық жұмыс. Программаларды жобалау әдістері, декомпозиция. 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!

Мысал: Арифметикалық төрт амалды (+,-,*, /) орындайтын калькуляторды бейнелейтін бағдарлама.

```
program calk;
var
    x, y, r: real; amal, a : char;
begin
    repeat
        write ('x = '); readln (x);
```

```

write ('y = ' ); readln (y) ;
writeln ('операция +,-,*,/' );
readln (amal ) ;
case amal of
    '+' : rezult := x + y ;
    '-' : rezult := x-y ;
    '*' : rezult := x*y ;
    '/' : rezult := x/y ;
else writeln ('қате белгі') ;
end;
writeln (x, amal,y,'=', r );
write ('жалғастырасың ба Y/N ' );
readln (a ) ;
until (a =' N') or (a ='n') ;
end.

```

Пайдаланылатын айнымалылар: **x** , **y** - берілген сандар; **r** - нәтиже;
amal айнымалысы 1 символдық типті, бұл айнымалыда арифметикалық операцияның белгісі сақталады. **a** айнымалысында бағдарламаның аяқталу белгісі сақталады, типі-символдық; **N** немесе **n** клавишасы басылғанда бағдарлама жұмысын аяқтайды.

Бағдарламаның орындалу барысында x,y мәндері клавиатурадан енгізіледі. Одан соң мына жолдармен

```

writeln ('операция +,-,*,/' );
readln (amal ) ;

```

операцияның белгісі *amal* айнымалысына енгізіледі. Case операторы *amal* айнымалысының мәні бойынша, арифметикалық операцияны орындайды. Мысалы, егер *amal* айнымалының мәні “ * “ болса, онда $r := x*y$ операторы орындалады. Егер *amal* мәні ешбір белгі болмаса, else-ден кейінгі жол орындалады да, экранға “қате белгі” деген хабарлама шығады. (Repeat, until операторлары туралы алдағы тақырыптарда түсінік беріледі).

Пайдаланушының типтері. Айнымалының қабылдайтын мәндері алдын ала белгілі жағдайда, пайдаланушы стандарттық емес өз типтерін жасауына болады.

Паскаль тілінде пайдаланушы жасаған типті сипаттау үшін **type** резервтелген сөзі қолданылады.

Жазылу форматы:

type

<типтің аты> = <тип мәні>;

Пайдаланушы жасайтын типтер екі түрлі болады. Олар: **санақтық** (перечисляемый) және **аралықтық** (интервальный) типтер.

Бұл типтегі берілгендер жадының бір байтын алады, сондықтан пайдаланушының типтері 256-дан көп элементтен тұра алмайды. Бұл типтерді қолдану енгізілетін мәндерді бақылайды, бағдарламаның көрнекілігін жақсартады,

жадыны үнемдейді және қатені іздеуді жеңілдетеді.

Санақтық тип - ие болатын мәндерінің тізімімен беріледі. Мәндер тұрақты болады және жақша ішінде үтірмен анықталып жазылады. Мәндердің тізімі реттелген жиынды құрайды. Бірінші мәннің реттік нөмері-ноль, екіншісі-бір, т.с.с.

Мысалы:

```
var
    planeta:(mars, pluton, saturn, mercury);
    kun:(1, 2, 3, 4, 5, 6, 7);
```

Бұл мысалдағы `planeta`, `kun` айнымалыларының типі санақтық, себебі, айнымалыны сипаттау бөлімінде `var` резервтелген сөзінен кейін айнымалылардың қабылдайтын тұрақты мәндері жақша ішінде тізіммен көрсетілген. `Planeta` айнымалысы `mars, pluton, saturn, mercury` мәндеріне, ал `kun` айнымалысы `1, 2, 3, 4, 5, 6, 7` мәндеріне ие бола алады.

Жоғарыдағы мысал бойынша `planeta`, `kun` типтерін жасасақ, олар былай сипатталады.

```
type planeta = ( mars, pluton, saturn, mercury);
    kun = (1, 2, 3, 4, 5, 6, 7);
```

пайдаланушы жасаған осы типтер айнымалыны сипаттау үшін былай қолданылады:

```
var    p1, p2, p3 : planeta;
        k1, k2 : kun;    tys : (ak, kara, sary, kuzyl);
```

Бұл мысалда **type** сөзінен кейін пайдаланушы жасаған `planeta`, `kun` типтері сипатталған, яғни, олардың мәндері ретінде бірнеше планета мен күндер номері анықталған.

Ал **var** сөзінен кейін `p1, p2, p3` айнымалылары `planeta` типімен, `k1, k2` айнымалылары `kun` типімен сипатталған. Яғни, `p1, p2, p3` айнымалылары `planeta` типінің бір мәніне, ал `k1, k2` айнымалылары `kun` типінің бір мәніне ие болады. Ал **tys** айнымалысы `ak, kara, sary, kuzyl` мәндерінің біріне ие болады

Аралықтық тип – берілген айнымалының қабылдайтын мәндерінің төменгі және жоғарғы шегін көрсететін екі тұрақты шаманы анықтайды.

Тұрақтыларға қойылатын талаптар:

1. Тұрақтылар бір типті болу керек.
2. Типтер `real`-дан басқа кез келген скаляр болады.
3. Бірінші тұрақты екіншіден кіші болу керек.

Мысал: `type`

```
kunder=1..31;
arip='a'..'я'
```

`var`

```
d,a: kunder;    l: arip
```

`Kunder` деген типті пайдаланушы өзі жасады. Төменгі шегі `1`, жоғарғы шегі `31`, типтері – `byte`. Осы типтегі `d, a` айнымалылары `1..31` аралығындағы мәндердің

біреуін қабылдайды. Сол сияқты, әгір типіндегі 1 айнымалысы а..я аралығындағы әріптердің біріне ие болады.

Сұрақтар

1. Таңдау операторы қай кезде қолданылады?
 2. Таңдау айнымалысы қандай қызмет атқарады?
 3. Таңдау операторының жазылу форматы қандай?
 4. Таңдау операторының орындалуын түсіндір ?
 5. Таңдау операторын пайдалану ережелері қандай
1. Стандарттық типтерді ата.
 2. Пайдаланушының типтері қай кезде қолданылады?
 3. Санақтық тип қалай жазылады?
 4. Аралықтық тип қалай жазылады?

**№6 зертханалық жұмыс Программалауға модульдық ыңғай.
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;>
```

```
    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.

3-мысал: $\frac{1}{2} + \frac{2}{2 \cdot 3} + \frac{3}{3 \cdot 4} + \dots + \frac{20}{20 \cdot 21}$ сан қатарының қосындысын табу.

Шешуі: Берілген есепте параметрі 1-ден 20-ға дейін 1 қадаммен өзгертін (бөлшектің алымы) цикл құрған дұрыс. Бөлімінде алымы мен оған келесі санның көбейтіндісі жазылады. Егер алымын **i** деп белгілесек, бөлімі **i · (i+1)** түрінде болады.

```
program kosyundy;
var i:byte; s : real;
begin
  s:=0;
  for i:=1 to 20 do
    s:=s+i/(i*(i+1));    { цикл денесі }
  write ('қосынды =',s);  writeln(y);  readln;
end.
```

4-мысал: 10-нан 99-ға дейінгі сандардың арасынан цифрларының қосындысы **n**-ге тең сандарды табу. ($0 < n \leq 18$)

Шешуі: Мынадай айнымалыларды белгілейміз.

n- берілген сан; **p1**- санның ондық разрядты цифрасы;

p2- санның бірлік разрядты цифрасы;

s- берілген санның цифрларының қосындысы;

k -ізелінді сан.

```
Program san_kosyundy;
var k,n,p1,p2,s : integer;
begin
  writeln ('бүтін сан енгіз');  readln (n);  { бүтін санды енгіземіз }
  for k:=10 to 99 do          { 10-нан 99-ға дейінгі сандарды қарастырамыз }
    begin                    { цикл денесінің басы }
      p1:=k div 10;
      p2:=k mod 10;          { бірінші, екінші цифраны табамыз }
      s:=p1+p2;
      if s=n then writeln (k);    { егер қосынды n-ге тең болса,
      end;                          берілген k санын шығарамыз }
      readln;
    end.
end.
```

Сұрақтар

1. Цикл дегеніміз не?
2. Циклді қолдануға мысалдар келтір.
3. Параметрлі қайталану қай кезде қолданылады?
4. Параметр деген не?
5. Параметрлі қайталанудың жазылу форматы.
6. For операторының орындалуын түсіндір.
7. Цикл денесі деген не?

№7 зертханалық жұмыс Программалауға құрылымдық ыңғай. Евклид алгоритмі

Теориялық бөлім

Евклид алгоритмі дегеніміз, ол - теріс емес бүтін екі санның ең үлкен ортақ бөлгішін (ЕҮОБ) табу алгоритмі.

x және y – бір уақытта 0-ге тең емес, бүтін, теріс емес сандар және $y \leq x$ болсын. Егер $y=0$ болса, онда $ЕҮОБ(x,y)=x$, ал егер $y \leq 0$ болса, онда x , y және r сандары үшін, (мұндағы r дегеніміз $x:y$ -тің қалдығы), $ЕҮОБ(x,y)=ЕҮОБ(y, r)$ теңдігі орындалады. Мысалы, $x=48$, ал $y=18$ болсын.

$$ЕҮОБ(48,18)=ЕҮОБ(18,12) = ЕҮОБ(12,6) = ЕҮОБ(6,0) =6$$

1-мысал: Теріс емес екі санның ең үлкен ортақ бөлгішін (ЕҮОБ) табу бағдарламасы.

1-әдіс: Қалдықтар арқылы.

```
program algor_evklid;
var x,y: integer;
begin
  writeln (' Екі санды енгіз' );  readln(x,y);
  repeat                          { циклдің басы }
    if x>y
      then x:=x mod y           { екі санның үлкенін үлкенді кішіге
      else y:=y mod x;          бөлгендегі қалдықпен ауыстыру }
  until (x=0) or (y=0) ;        { екі санның біреуі 0-ге тең
                                болғанша, цикл орындалады }
  writeln ('ЕҮОБ= ',x+y);      readln;  end.
```

2-әдіс: Айырма арқылы.

```
program algor_evklid;
var x,y: integer;
begin
  writeln (' Екі санды енгіз' );  readln(x,y);
  repeat                          { циклдің басы }
    if x>y
      then x:=x - y else y:=y - x; { екі санның үлкенін
                                айырмамен ауыстыру }
  until x=y;                      { екі санның тең болғанша қайталады }
  writeln ('ЕҮОБ= ',x);  readln;
end.
```

Евклид алгоритмін негізге ала отырып, екі санның ең кіші ортақ еселігін (ЕКӨЕ) табуға болады. Формуласы: $ЕКӨЕ(x,y) = x*y/ ЕҮОБ(x,y)$.

№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 uppercase(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’; a:=’14.2’	val(a,x,cod) val(a,x,cod) val(a,x,cod)	cod=0 cod=0 cod=5
3.st:=’1237’ st:=’klassio’	length(st) length(st)	4 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 ‘a’ әріптерін санау, ‘b’ әрпіне ауыстыру.

1- әдіс: Алгоритм:

- 1.Сөзді енгізу.
- 2.Сөздің бірінші әрпін белгілеу.
- 3.Белгіленген символды ‘a’ символымен салыстыру.
- 4.Егер сәйкес болса, санауышты бірге арттыру, сол символдың орнына ‘b’ символын қою;

5.Осылайша барлық символдарды қарастыру.

Бағдарлама.

```
program sanau;
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 sanau;
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;
  writeln ('сөзде, 'n, ' a әрпі бар');
end.
```

№6 тақырып. Ішкі бағдарламалар: процедуралар және функциялар

№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 қирсетілген *формальді параметрлер тізімінен* тұрады. Процедура денесі бағдарламалық блоктан тұрады.

Процедураның жалпы т%орі:

```
procedure <аты>[(формалдық параметрлер тізімі)];
```



```
сипаттама бұлімі
begin
    операторлар бұлімі
end;
```

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

Фактілік және формальдық параметрлердің арасында мынадай сәйкестік болу керек:

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

Формальдық параметрлердің мынандай т%орлері болады:

- параметр – мөндер;
- параметр – айнымалылар;

Параметр – мөндер. Параметр – мөндер негізгі бағдарламадан ішкі бағдарламаға мөндерді беру %шін ғана қолданылады, процедура нәтижесі негізгі бағдарламаға қайтарылмайды. Параметр – мөндер фактілік мөндерге ешқандай ықпал жасамайды.

1-мысал: түрт санды екі-екіден ж±птап, квадраттарының қосындысын табу.

Б±л мысалда параметр – мөндер 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 формальдық 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 т%сініктеме

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 мөндеріне 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 мынадай 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 ескеру (қайтару) үшін 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. Дерезенің нетижесі d негізгі бағдарламаға қайтарылатындықтан `var d:real` сипаттамасы жазылады. i -формальдық параметр емес, сондықтан ол процедураның сипаттама бөлімінде жазылған.

Процедура денесінде a санының n рет қубейтіндісі есептеледі, a мен n -нің мәндері негізгі бағдарламадан беріледі.

Негізгі бағдарламаға сипаттама. Негізгі бағдарламада қолданылатын фактілік параметрлер: x, a_4, a_3, a_2, y, s ; y -нетиже, ал s -санның дерезесінің нетижесі.

x, a_4, a_3, a_2 -мәндері клавиатурадан енгізіледі.

x_4, s -параметрлерімен `dareje` процедурасы шақырылады.

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

Негізгі бағдарламада $y:=s*a_4$ мәні есептеледі.

$x, 3, s$ параметрлерімен процедура тағы да шақырылады, яғни $a=x, n=3$ мәндерінде x -тің 3 дерезесі есептеледі. Нетижесі, яғни d -ның мәні s -ке

меншіктеледі, s жаңа мәнге ие болады да, негізгі бағдарламада $y:=y+s*a_3$ есептеледі. $x, 2, s$ параметрлерімен процедура шақырылып, іс-ерекет қайталанады.

№10 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 бөлімі

`begin`

функция денесі

`end;`

Функция негізгі бағдарламадан аты бойынша шақырылады. Функцияға берілетін мәндер, ондағы формальдық 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:=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+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

№11 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:text; 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: text; 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

№12 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}

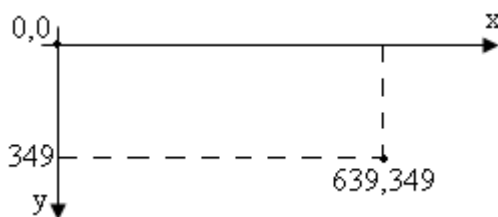
InitGraph(dv,mv,'c:\tP7\BGI'); {graph модулінің орналасқан жолы көрсетіледі}

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
Көпбұ-рыш 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); </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; </pre>
--	---

<pre> setfillstyle(1,6); bar(100,250,200,300); readln; closegraph; end.</pre>	<pre> pp[4].x:=150; pp[4].y:=200; pp[5]:=pp[1]; DrawPoly(5,pp);FillPoly(3,pp); readln; closegraph; end.</pre>
---	---

```

program kozgalys;
uses Graph,CRT;
var
x,y,i,dv, mv:integer; x1,x2,y1,y2:integer;
begin
dv:=detect; initgraph(dv,mv, 'c:\bp\bgi');
if graphresult <>grok then halt (1);
x:=460;y:=400;           суреттің бастапқы координаталары
for i:=1 to 200 do
begin
setcolor(6);
circle(460,y,40);
ellipse(460,y-40,0,180,20,20);
ellipse(440,y-60,0,90,20,20);
ellipse(480,y-60,90,180,20,20);
ellipse(440,y-40,270,360,20,65);
ellipse(480,y-40,180,270,20,65);
circle(440,y-20,5); circle(480,y-20,5);
circle(440,y+10,5); circle(480,y+10,5);
delay(500);   пауза
SetColor(0);
circle(x,y,40);
ellipse(460,y-40,0,180,20,20);
ellipse(440,y-60,0,90,20,20);
ellipse(480,y-60,90,180,20,20);
бұл
ellipse(440,y-40,270,360,20,65);
cleardevice
ellipse(480,y-40,180,270,20,65);
circle(440,y-20,5); circle(480,y-20,5);
circle(440,y+10,5); circle(480,y+10,5);
delay(500);
y:=y-1;   жаңа координата
end;
SetColor(6);
circle(x,y,40);
ellipse(460,y-40,0,180,20,20); ellipse(440,y-60,0,90,20,20);

```

сурет салынды

сурет өшірілді,
фрагментті

процедурасымен
алмастыруға
болады.

```

ellipse(480,y-60,90,180,20,20); ellipse(440,y-40,270,360,20,65);
ellipse(480,y-40,180,270,20,65);
circle(440,y-20,5); circle(480,y-20,5);
circle(440,y+10,5); circle(480,y+10,5);
readln;
closegraph;
end.

```

```

program grafik;
uses Graph,CRT;
var
x,y,i,dv, mv:integer; x1,x2,y1,y2:integer;
begin
dv:=detect; initgraph(dv, mv, 'c:\tp7\bgi');
if graphresult <>grok then halt (1);
line(300,50,300,400); line(150,400,500,400);
for i:=-100 to 100 do
begin
x:=i; y:=trunc(sqrt(x));
putPixel(trunc(x/2+300),trunc(400-y/50),14);
end;
readln;
closegraph;
end.

```

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

Тақырып 7. Программалауға объектілік-бағытталған амал
№13 зертханалық жұмыс. Деректердің құрылымдандырылған типтерін өңдеу.

57 мысал. Сызықтық алгебраның жиі қолданылатын алгоритмдерін модульге біріктіреміз.

```

unit algebra;
interface
const n=10;
type vector=array[1.. n] of real;
type mr=array[1.. n] of vector;
function scalar(n:byte; a,b:vector):real;
procedure summatr(n,m:byte; a,b:mr; var c:mr);
implementation
uses crt;
var i,j,k:byte;
procedure sumvector(n:byte; a,b:vector; var c:vector);

```

```

begin
  for i:=1 to n do
    c[i]:=a[i]+b[i]; end;
  procedure peek; begin sound(400); delay(100); nosound; end;
  function scalar;
    var s:real;
    begin
      s:=0;
      for i:=1 to n do
        s:=s+a[i]*b[i]; scalar:=s; end;
    procedure summatr;
    begin
      for i:=1 to n do
        sumvector(m,a[i],b[i],c[i]); end;
    begin readln; end.

```

Егер бұл модульді компиляцияласа, algebra.tpu файлы пайда болады. Бұл модуль біз құрған программаларды пайдаланады.

Мысалы:

```

uses algebra;
var w,q,z:matr;
begin summatr(3,4, w, q, z);
end.

```

Программа мен онда пайдаланылатын модульдер F9 клавишасымен компиляцияланады. Нәтижесінде .tpu .exe файлдары пайда болады. algebra модулінің тексі жазылған файлдың аты algebra.pas болу керек!

№14 зертханалық жұмыс. Деректердің динамикалық құрылымдарын өңдеу.

Delphi ортасымен танысу

Delphi ортасы – программист жұмысының өте жоғары тиімділігін қамтамасыз ететін орта.

Delphi ортасын шақырғаннан кейін ортаның жұмысын басқаратын алты негізгі терезе шығады (1-сурет).

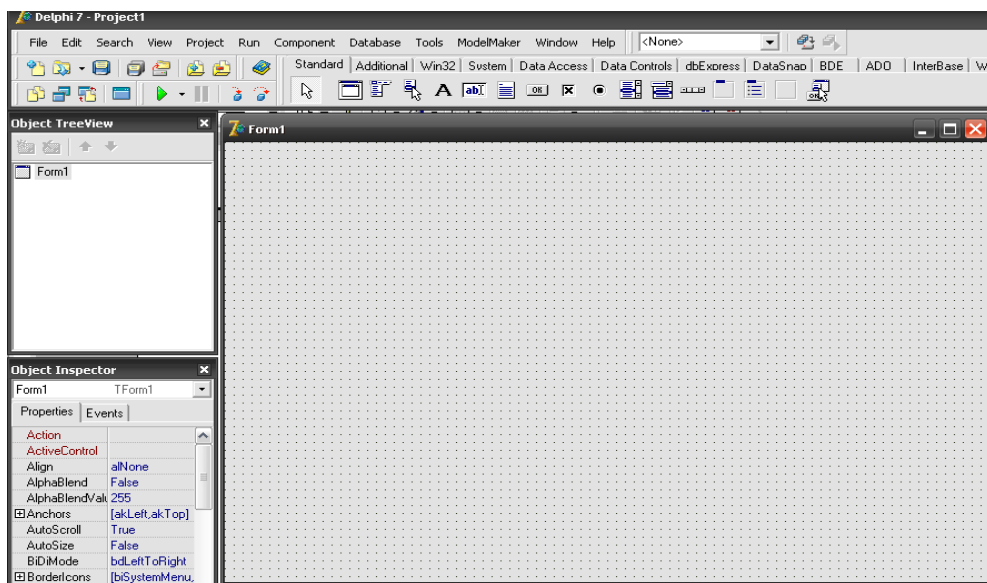
1) негізгі терезе – негізгі терезе құрылатын программаның жобаларын басқаратын негізгі қызметті атқарады. Бұл терезе экранда барлық уақытта болады және ең жоғары бөлігінде орналасады. Негізгі терезеде Delphi ортасының бас менюі, пиктографиялық командалы батырмалар жиынтығы, компоненттер палитрасы орналасқан.

2) объектілер инспекторының терезесі – формаға орналасатын кез келген компонент бірқатар параметрлер жиынымен сипатталады: орналасуы, өлшемі, түсі және т.б. Мысалы, компоненттің орналасуы мен өлшемін программист

формалар терезесінде компонентті басқара отырып, өзгерте алады. Ал, объектілер инспекторы терезесі қалған өзгертулерді орындауға арналған. Бұл терезеде екі бет бар: Properties (Қасиет) және Events (Оқиға). Properties беті – компоненттің керекті қасиеттерін тағайындау қызметін атқарады. Events беті – қандай да бір оқиғаға компоненттің жауабын анықтайды.

3) формалар терезесі – windows жобасы болашақта программа жазатын терезе болып табылады. Басында бұл терезе бос болады Дәлірек айтқанда, ол windows-дің интерфейстік элементтері үшін стандартты жүйелік менюді шақыру батырмаларынан, терезені үлкейту, кішірейту, жабу батырмаларынан және тақырып жолынан тұрады. Әдетте, терезенің жұмыс алаңы нүктелік координаталық тормен толтырылған, ол формадағы компоненттердің орналасуын реттеу қызметін атқарады (бұл нүктелерді алып тастау үшін Tools/Environment Options меню командаларының көмегімен, Preferences қойымтасындағы Display Grid ауысытырып қосқышындағы жалаушаны алып тастау қажет).

4) программа кодының терезесі – программа мәтінін құрып, жөндеуге арналған. Бұл мәтін арнайы ереже бойынша құрылады және программаның жұмыс алгоритмін сипаттайды. Мәтінді жазу ережелерінің жиынтығы программалау тілі деп аталады.



Сурет 1 - Delphi ортасының интерфейсі

2 Delphi-ді баптау

Программалар жобасымен жұмыс істеу процесінде формалар мен модульдер жиынтығын құру қажет. Бұл мәліметтерді жеке қапшықта файлдар түрінде сақтаған дұрыс. Delphi ортасының стандартты келтірілген күйіне өзгеріс енгізу қажет. Ол құрылған программаның соңғы нұсқасының автоматты түрде сақталуы үшін керек. Ол үшін Tools/Environment Options сұхбат терезесінде Preferences бетін таңдау керек. Осы беттің сол жақ жоғарғы бұрышында Autosave Options тобында «Editor Files»

және «Desktop» деген ауыстырып қосқыштары бар. Ауыстырып қосқыштарды екпінді жасау – программа коды терезесінің мәтінін және Delphi терезесінің жалпы орналасуын автоматты түрде сақтайды. Бұл программаны жоғалудан қорғайды. Мәтін шығару үшін шрифтті тағайындау қажет:

Tools/Editor Options – Display бетінде Editor Font-қа Times New Roman шрифт түрін тағайындау қажет.

Программаны жүргізуге барлығы дайын болған соң негізгі терезеден батырмасын немесе F9 басқару пернесін басу керек. Сонымен, Delphi ортасындағы программа үш кезеңнен өтеді: компиляция – компоновка – орындау.

Компиляция кезіңінде код терезесінде дайындалған программа мәтіні Object Pascal тіліне ауыстырылады; компоновка кезеңінде қажетті қосалқы, ал орындалу кезеңінде дайын программа жедел жадыға шақырылып, орындалуға жіберіледі.

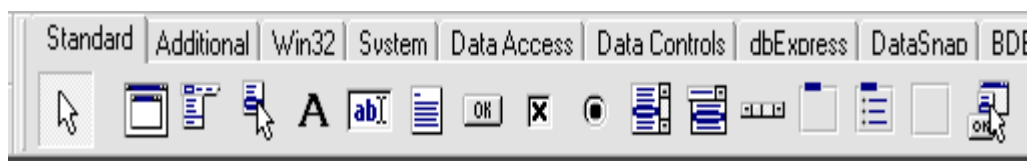
F9 пернесін басқаннан кейін Save Unit1 As сұхбат терезесі пайда болады, яғни Delphi Unit1.pas модулі үшін файлдың аты мен ол орналасатын папканың атын сұрайды, сондай-ақ, модульдің де атын өзгерту қажет.

2.1 Компоненттермен танысу

Компонеттер дегеніміз жұмыс істейтін программада құрылатын, көрінетін бейнелер тұрғызылатын элементтер болып табылады. Компонеттер палитрасының ішінде бейне ретінде көрінбейтін де компоненттер жеткілікті.



2.2 STANDARD беті








STANDARD бетінде программа құруға ең қажет Windows-ге арналаған стандартты интерфейстік элементтерден тұратын компоненттер палитрасы орналасқан (2-сурет).



Сурет 2 - STANDARD беті

Кесте 1- STANDARD бетіндегі компоненттер тізімі

Компонеттер	Сипаттамасы
 Frame	Басқа компоненттерді орналастыруға арналған контейнер қызметін атқарады.
 Main Menu	Программаның бас менюі. Бұл компонент күрделі иерархиялық меню құр меню құрып жұмыс істеуге мүмкіндік береді.

 Label	Мәтінді шығару өрісі.
 Edit	Мәтінді енгізу/редакциялау өрісі
 Button	Командалық батырма
 CheckBox	Таңдаудың тәуелсіз батырмасы
 RadioButton	Таңдаудың тәуелді батырмасы
 ListBox	Таңдаудың тізімі
 ComboBox	Таңдаудың комбинаторлық тізімі











2.3 ADDITIONAL беті



ADDITIONAL бетінде сұхбат терезелерінің түрін түрлендіруге мүмкін-дік беретін 18 қосымша компоненттер орналасқан (3-сурет)













Сурет 3 - ADDITIONAL беті

Кесте 2 - ADDITIONAL бетіндегі компоненттер тізімі

Компоненттер 1	Сипаттамасы 2
 BitBtn	жазуы бар, пиктограммалы командалық батырма
 SpeedButton	пиктографиялық батырма. Әдетте бас менюдің опцияларына жылдам кіру үшін пайдаланылады.
 MaskEdit	арнайы мәтіндік редактор. Енгізілген мәтінді сүзгілеуге мүмкіндік береді, мысалы, датаны дұрыс енгізу үшін қажет.
 StringGrid	жолдар кестесі. Бұл компонент мәтіндік ақпаратты кесте түрінде шығаруға мүмкіндік береді.
 DrawGrid	еркін кесте. StringGrid компонентінен айырмашылығы бұл компоненттің ұяшықтарында кез келген ақпарат, оның ішінде сурет те сақталады.
 Image	сурет. Бұл компонент суреттерді, оның ішінде пиктограммалар мен метафайлдарды бейнелуге арналған.
 Shape	фигура. Бұл компонент көмегімен терезеге дұрыс геометриялық фигураларды – тіктөртұрыш, шеңбер, эллипс және т.б. қоюға болады.
 Bevel	жиек. Терезенің жеке бөлектерін үш өлшемді қоршаумен немесе жолақпен ерекшелеуге арналған.
 ScrollBar	орағыш жолақтар панелі. Panel компонентімен айырмашылығы – егер орналасқан компоненттер оның шекараларымен қиылысып қалса, онда орағыш жолақтарды автоматты түрде қояды.
 CheckListBox	жиындық таңдаулар тізімі. Стандартты ListBox компонентінен айырмашылығы әрбір опциясының қасында бірнеше опцияны бірден таңдауды жеңілдететін CheckBox тәрізді тәуелсіз ауыстырып

	қосқышы бар
 Splitter	шекара.Бұл компонент формада көрінетін екі компоненттің арасында орласады да, пайдаланушыға программаны жүргізу кезінде компоненттерді бір-бірінен бөліп тұрған шекараларын алмастыруға мүмкіндік береді.
 StaticText	Статикалық мәтін. Стандартты Label компонентінен айырмашылығы өзінің жеке Windows терезесінің болуымен сипатталады.

2 кестенің жалғасы

1	2
 ControlBar	басқару жолағы. Drag&Drop технологиясында “тақалып” тұратын компоненттерді орналастыратын контейнер қызметін атқарады.
 ApplicationEvents	оқиғаны қабылдаушы. Егер бұл компонент формаға орналасқан болса, онда программаға арналған барлық Windows хабарламаларын алып отырады.
 ValueListEditor	атау = мән жұбынан тұратын жолдар редакторы. Мұндай жұптар Windows-де жиі пайдаланады.
 LabeledEdit	бір жолдық редактор мен белгінің комбинациясы
 ColorBox	жүйелі түстердің бірін таңдауға арналған арнайы ComboBox варианты
 Chart	диаграмма. Бұл компонент мәліметтерді графикалық жолмен беруге арналған арнайы панельдер құруды жеңілдетеді.
 ActionManager	әрекеттер менеджері. Төмендегі үш компонентпен бірге қосымшалар, интерфейс құруды қамтамасыз етеді.
 ActionMainMenuBar	меню жолағы, ол оның опциялары ActionManager көмегімен құрылады.
 ActionToolBar	ActionManager компонентінің көмегімен құрылатын пиктографиялық батырмаларды орналастыруға арналған жолақ.
 CustomizeDLG	баптау сұхбаты. Бұл компоненттің көмегімен пайдаланушы жұмыс программасының интерфейсін қалауынша баптай алады.

№15 зертханалық жұмыс. Деректер қорын жасаудың құралдары мен технологиясы.

Тапсырманың берілуі. X және Y бүтін сандар берілген. Олардың қосындысын табу.

Формаға келесі компоненттерді орналастыру

Компонеттер	Объектілер инспекторы терезесіне енгізілетін өзгертулер
Edit1:Tedit;	Text:бос қалдыру
Edit2:Tedit;	Text:бос қалдыру

Label1:Tlabel;	Caption: «X мағынасын енгізіңіз»
Label2:Tlabel;	Caption: «Y мағынасын енгізіңіз»
Label3:Tlabel;	Caption: «Есептеу нәтижесі»
Memo1:Tmemo;	Lines: бос қалдыру
Button1:Tbutton	Lines: бос қалдыру

```

unit U1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;
Type
  TForm2 = class(TForm)
  Edit1: Tedit;
  Edit2: Tedit;
  Label1: Tlabel;
  Label2: Tlabel;
  Label3: Tlabel;
  Memo1: Tmemo;
  procedure Button1Click(Sender: TObject);
  var
    x, y: integer;
  begin
    edit1.setfocus; // edit1 компонент атына енгізу фокусын орналастырады.
    x := strtoint(edit1.text); // аргумент мағынасын жолға түрлендіреді
    y := strtoint(edit2.text);
    memo1.clear; // тазарту
    memo1.Lines.Add(inttostr(x+y));
  end;
end.

```

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.

