ЛАБАРАТОРНАЯ РАБОТА № 1

Вывучэнне паслядоўнасці апрацоўкі Паскаль-праграм. Устроеныя матэматычныя працэдуры і функцыі

Мэта: азнаёміць з сістэмай Экспрэс Паскаль, галоўным меню, тэкставым рэдактарам, структурай праграмы; навучыць карыстацца аператарам прысвойвання, вылічваць арыфметычныя выразы, арганізоўваць увод і вывад даных.

Экспрэс Паскаль прадстаўляе сабой інтэграваную сістэму, якая ўключае рэдактар тэкстаў, кампілятар і бібліятэку працэдур часу выканання. У працэсе работы сістэма выкарыстоўвае ўсю памяць камп'ютэра. Пасля загрузкі сістэмы на экране з'яўляецца галоўнае меню, у якім выдзелены асобныя сімвалы. Кожны сімвал ці іх група адпавядае сваёй камандзе меню. Пасля націску клавіш з сімваламі выконваюцца каманды, прычым у некаторых выпадках пры гэтым на экране з'яўляюцца пытанні, ці патрабуецца ўвод параметраў.

Асноўныя этапы паслядоўнасці апрацоўкі Паскаль-праграм:

1. Стварэнне і рэдагаванне зыходнай праграмы.
2. Кампіляцыя, перавод тэксту зыходнага модуля на машынную мову.
3. Выкананне праграмы.

Галоўнае меню сістэмы Экспрэс Паскаль

Пасля загрузкі сістэмы экран прымае такі выгляд:

```
 Express Pascal v.2.0. Корвет - CP/M-80 © mkSoft
-------------------------------------------------
Initialize disk system      ^Compile to Memory (Disk)
Default drive user: A00:
Main file:                  ^Parameter string:
Work file:

Edit Pick Save Verify       ^BAK-copy: off    ^Verify:off
Compile Run Quit Help                ^Autosave EP.CFG: off
Find run-time error         ^Default/^Restory/^Save EP.CFG
```

У гэтым меню змешчаны ўсе каманды, якія можа выконваць сістэма Экспрэсс Паскаль. Яно падзелена на дзве часткі: левую і правую. На экране ў назве кожнай каманды першая літара (у левай частцы) ці сімвал ^ і першая літара (у правай частцы) адмечаны белым прамавугольнікам. Гэта азначае, што для выканання каманды неабходна націснуць клавішу з адмечанай літарай (пры

націснутай клавішы Ctrl, калі ёсць ^). Напрыклад, каб выканаць каманду Run, трэба націснуць клавішу R; каб выканаць каманду Parameter string, трэба пры націснутай клавішы Ctrl націснуць P.

Каманды асноўнага меню

Initialize disk system – ініцыялізаваць дыскавую сістэму. Для выканання гэтай каманды трэба націснуць клавішу I.

Гэта каманда адлюстроўвае некаторыя асаблівасці аперацыйнай сістэмы CP/M. Аперацыйная сістэма сочыць за тым, якая дыскета знаходзіцца ў дыскаводзе, і, калі дыскета мяняецца, аўтаматычна пераводзіць дыскавод ў становішча "толькі чытанне" (каб знарок не сапсаваць дыскету). Таму, калі вам патрэбна запісаць адрэдагаваны тэкст праграмы ці адкампіліраваную праграму на другую дыскету, то вы павінны пасля ўстаноўкі новай дыскеты выканаць каманду Initialize disk system.

Default drive user – устанавіць дыскавод і нумар карыстальніка, якія далей улічваюцца аўтаматычна. Для выканання гэтай каманды трэба націснуць клавішу D.

Пры рабоце аперацыйнай сістэмы CP/M адзначана паняцце цяперашняга дыскавода, на які будуць запісвацца файлы і будуць шукацца файлы для чытання, калі яўна не ўказаны другі дыскавод. У сістэме CP/M версіі 2.2 работа з рознымі нумарамі карыстальніка вельмі нязручная, і таму не трэба змяняць нулі пасля назвы дыскавода.

Main file – галоўны файл. Для выканання гэтай каманды трэба націснуць клавішу M.

Галоўны файл – гэта файл, які змяшчае зыходны тэкст праграммы, з якога павінна пачацца кампіляцыя. Галоўны файл трэба вакарыстоўваць у тых выпадках, калі праграма разбіваецца на некалькі файлаў, што ўключаюцца ў адзін з дапамогай дырэктывы кампілятара {$I імя файла}. Пры напісанні невялікіх праграм гэта каманда не выкарыстоўваецца.

Work file – рабочы файл. Для выканання гэтай каманды трэба націснуць клавішу W.

Рабочы файл – гэта файл, які змяшчае зыходны тэкст праграмы, якую трэба рэдагаваць. Імя рабочага файла запісваецца лацінскімі літарамі да 8 сімвалаў, расшырэнне PAS задаецца аўтаматычна.

Edit – рэдагаваць тэкст праграмы. Для выканання гэтай каманды трэба націснуць клавішу E.

Пасля яе выканання сістэма пераходзіць у рэжым тэкставага рэдактара, дзе можна рэдагаваць зыходны тэкст праграмы, які знаходзіцца ў памяці камп’ютэра альбо запісваць тэкст новай праграмы. Некаторыя каманды рэдагавання змешчаны ў дадатку 1.

Pick – выбраць чарговы зыходны тэкст праграмы з pick-табліцы. Для выканання гэтай каманды трэба націснуць клавішу P.

У pick-табліцы захоўваецца інфармацыя аб 12-і апошніх файлах, з якімі працавалі. Для кожнага файла захоўваецца: імя файла, месца знаходжання курсора і маркера ў тэксце, размяшчэнне тэксту на экране. Pick-табліца захоўваецца ў файле EP.CFG.

Save – запісаць зыходны тэкст праграмы на дыск. Для выканання гэтай каманды трэба націснуць клавішу S.

Пасля яе выканання запісваецца зыходны тэкст праграмы з аператыўнай памяці на дыск. Гэту каманду лепш выкарыстоўваць заўсёды, калі праграма створана альбо ў яе ўнесены змены.

Verify – параўнаць тэкст у памяці з тэкстам, які запісаны на дыску. Для выканання гэтай каманды трэба націснуць клавішу V.

Compile – кампіліраваць. Для выканання гэтай каманды трэба націснуць клавішу C.

Пасля яе выканання ўся праграма (альбо модуль) пераводзіцца ў машынныя каманды для выканання працэсарам. Калі ў тэксце праграмы знаходзяцца памылкі мовы праграмавання, то кампіляцыя перарываецца і з'яўляецца паведамленне аб памылцы. Адкампіліраваная праграма (альбо модуль) ці змяшчаецца ў аператыўнай памяці, ці запісваецца на дыск – у залежнасці ад стану, які ўказаны ў радку Compile to Memory/Disk асноўнага меню. Калі праграма запісваецца на дыск, то яна запісваецца ў файл, імя якога супадае з іменем рабочага файла і прымае расшырэнне COM. Гэта звычайны праграмны файл CP/M, і ён цалкам гатовы да выканання.

Run – выканаць праграму. Для выканання гэтай каманды трэба націснуць клавішу R.

Пасля гэтага выконваецца толькі адкампіліраваная ў памяць праграма. Інакш з'яўляецца паведамленне аб немагчымасці выканання. Адкампіліраваную праграму можна выконваць некалькі разоў, да той пары, пакуль у яе тэкст не будуць ўнесены змены.

Quit – выхад з сістэмы Экспрэс Паскаль. Для выканання гэтай каманды трэба націснуць клавішу Q.

Help – падказка. Для выканання гэтай каманды трэба націснуць клавішу H.

Пасля яе выканання на экране адлюстроўваецца падказка – кароткая інфармацыя аб камандах асноўнага меню і аб камандах рэдактара тэкстаў, пры гэтым неабходна, каб на цяперашнім дыску знаходзіўся файл EP.HLP (у ім захоўваецца тэкст падказкі).

Find run-time error – знайсці памылку часу выканання. Для выканання гэтай каманды трэба націснуць клавішу F.

Пасля яе выканання ідзе пошук пункта ў зыходным тэксце праграмы, дзе ўзнікла памылка часу выканання.

Compile to Disk/Memory – устанавіць рэжым кампіляцыі на дыск/у памяць. Для выканання гэтай каманды трэба націснуць клавішу C пры націснутай клавішы Ctrl.

Гэта каманда ўстанаўлівае месца размяшчэння вынікаў кампіляцыі: на дыск альбо ў памяць. Калі ў асноўным меню знаходзіцца радок Compile to Disk (Memory), то адкампіліраваная праграма будзе запісана на дыск у COM-файл; калі ў асноўным меню знаходзіцца радок Compile to Memory (Disk), то адкапіліраваная праграма (альбо модуль) будзе змешчана ў памяці (звычайна менавіта гэта асноўны рэжым кампіляцыі).

Parameter string – радок параметра. Для выканання гэтай каманды трэба націснуць клавішу P пры націснутай клавішы Ctrl.

Гэта каманда дазваляе адрэдагаваць радок параметраў, якія перадаюцца ў праграму пры яе запуску.

BAK-copy on/off – уключыць/выключыць стварэнне BAK-копіі. Для выканання гэтай каманды трэба націснуць клавішу B пры націснутай клавішы Ctrl.

Гэта каманда пераключае рэжым стварэння BAK-копіі, якая можа з'яўляцца пры выкананні каманды Save. Пры уключаным рэжыме перад тым, як захаваць тэкст праграмы ў рабочы файл, папярэдняя яго версія змяшчаецца ў файл з іменем рабочага файла, але прымае расшырэнне BAK.

Verify on/off – уключыць/выключыць аўтаматычную праверку пры запісу файла на дыск. Для выканання гэтай каманды трэба націснуць клавішу V пры націснутай клавішы Ctrl.

Астатнія каманды адпаведна адносяцца да работы з файлам канфігурацыі EP.CFG.

Простыя тыпы даных

Тып	***Абазначэнні***	***Межы***	***Патрабуецца памяці (байт)***
цэлы	byte	0..255	1
	word	0..65535	2
	integer	-32768..32767	2
	shortint	-128..127	1
	longint	-2147483648..2147483647	4
сапраўдны	real	2.9*E*-39..1.7*E*38	6
сімвальны	char	кодавая табліца ПЭВМ	1
лагічны	boolean	true, false	1

Аператар прысвойвання

Гэты аператар абазначаецца як *:=* і мае такі агульны выгляд:

пераменная:=выраз;

Пры гэтым трэба ўважліва сачыць за тым, каб супадалі тыпы пераменнай у левай частцы аператара і значэнне выразу ў правай частцы. Магчыма прысвойванне цэлага значэння выразу пераменнай сапраўднага тыпу.

Аператар уводу

Для ўводу інфармацыі з клавіятуры выкарыстоўваецца аператар *read*. Фармат:

read (спіс пераменных); ці *readln(спіс пераменных)*;

Напрыклад, *readln (a,b,c)*;.

Пераменныя пералічваюцца праз коску. У час выканання аператара ўводу праграма прыпыняецца і чакае ўводу інфармацыі з клавіятуры. Увод ажыццяўляецца наборам лікаў, падзеленых прабелам, і заканчваецца націскам клавішы ўводу. Неабходна адзначыць, што значэнне, якое ўводзіцца, павінна адпавядаць тыпам пераменных і сэнсу праграмы.

Напрыклад,

```
var b:real; a:char;
BEGIN
    read(a,b);
```

Калі ў нас ёсць некалькі аператараў *read*, падзеленых другімі камандамі, то ў час выканання першага аператара *read* можна ўвесці ўсе пераменныя. Для астатніх камандаў праграма не прыпыніцца. У адрозненне ад аператара *read* аператар *readln* пасля ўводу ўсіх указаных у аператары даных здзяйсняе пераход да наступнага радка.

Аператар вываду

Аператар вываду *write* служыць для вываду інфармацыі на экран дысплея ці прынтэра. Выгляд аператара:

write (спіс выразаў);.

Выразы ў спісе падзяляюцца коскамі. У выніку выканання аператара *write* могуць вылічвацца значэнні выразаў:

*write (' y= ', 2*y+sqr(y));*

Радкі выводзяцца без усякіх змен, пры гэтым самі апострафы не выводзяцца. Разам з аператарам *write* выкарыстоўваецца аператар *writeln*. Адрозненне іх у наступным: пры выкананні аператара *write* інфармацыя выводзіцца на экран і курсор застаецца ў тым жа радку. Пераход на наступны радок выконваецца толькі тады, калі радок, які выводзіцца, будзе поўнасцю запоўнены. У выніку

выканання аператара *writeln* пасля вываду інфармацыі курсор пераводзіцца на наступны радок. Вынікам выканання камандаў

write (1); writeln; write(-2,3);

будзе:

1

-23.

Калі не прыняць спецыяльных мер, то значэнні сапраўдных пераменных і выразаў выводзяцца ў форме з плаваючай коскай. Для вываду сапраўдных лікаў у форме з фіксаванай кропкай указваецца фармат пасля ліку:

write(лік:w:d);.

Тут: *w* – агульная шырыня поля ліку пасля вываду; *d* – колькасць лічбаў пасля кропкі, якая аддзяляе дробную частку ад цэлай (апошнія лічбы акругляюцца).

writeln(- 73.12:6:1);.

Вынік (на экране клетак няма, будзем лічыць, што адна клетка табліцы – адно знакамесца на экране):

	-	7	3	.	1

Калі колькасць сімвалаў у прадстаўленні вывадзімага значэння аказваецца меншай, чым *w*, то да яго злева дабаўляюцца прабелы. Калі *w* не ўказываецца, то назначаецца стандартная даўжыня поля, часцей за ўсё 13 знакамесцаў.

Прыклады аператараў вываду:

1. Арганізацыя вываду значэнняў пераменных *a, b, c* ***цэлага тыпу***:

```
    var a,b,c:integer;
BEGIN
    readln(a,b,c); writeln(a,b,c);
```

12_ 34_ 98 – увод (_ – гэта знак прабела ва ўсіх астатніх прыкладах)

123498 – вывад;

```
write(a:4,b:5); writeln(c);
```

12_ -34_ 6 – увод;

_ _ 12_ _-346 – вывад;

```
write('a=',a:4,'_ b=',b:5);writeln(c);
```

555_ -899_ -8 – увод;

a=_ 555___b=_-899-8 – вывад.

Калі колькасці ўказаных пазіцый недастаткова, то аўтаматычна павялічваецца месца вываду да неабходных памераў.

Напрыклад, у аператары *writeln(30521:2)* для вываду ўказана толькі 2 знакамесцы, але на экране выведзецца лік 30521.

2. Арганізацыя вываду ***сапраўдных лікаў***:

var a,b,c:real;

BEGIN

readln(a,b,c); writeln(a,b,c); – даўжыня поля не ўказана;

222.677777 _ 888.4444444_ -6.7 – увод ;

_2.2267777700e+02_8.884444440e+02-6.7000000000e+00 – вывад ;

writeln(a:6:3,b:7:2,c:4:1);

222.677777_888.4444444_-6.7 – увод ;

_222.678_888.44-6.7 – вывад .

Калі ў аператары вываду ўказана агульная колькасць пазіцый *w* і не запісана колькасць пазіцый пасля коскі, то лікі выводзяцца ў экспаненцыяльнай форме з шырынёй поля *w*.

writeln(a:6,b:7,c:4);

222.677777_ 888.4444444_ -6.7 – увод;

_2.2e+02_8.9e+02-6.7e+00 – вывад.

3. Арганізацыя вываду значэнняў ***сімвальнага тыпу***:

var a,b,c:char;

BEGIN

readln(a,b,c); writeln(a:6,b:8,c:4);

pro – увод без прабелаў (прабел – гэта сімвал);

_ _ _ _ _p_ _ _ _ _ _ _ _ r_ _ _ o – вывад;

writeln(a,b,c);

pro – увод;

pro – вывад.

4. Пры вывадзе значеннняў ***булеўскага тыпу*** на экран выводзіцца *True* ці *False*.

var a,b,c:integer;

BEGIN

readln(a,b,c); writeln(a<b:7,b>c:4);

1 2 3 – увод;

_ _ _TrueFalse – вывад.

Арыфметычныя аперацыі

Абазначэнне	***Прызначэнне***	***Тып аргументаў***	***Тып выніку***
a+в	складанне *а* і *в*	толькі цэлы	цэлы
		толькі сапраўдны	сапраўдны
		адзін цэлы, другі сапраўдны	сапраўдны

а–в	адыманне *в* ад *а*	толькі цэлы	цэлы
		толькі сапраўдны	сапраўдны
		адзін цэлы, другі сапраўдны	сапраўдны
*а*в*	множанне *а* на *в*	толькі цэлы	цэлы
		толькі сапраўдны	сапраўдны
		адзін цэлы, другі сапраўдны	сапраўдны
а/в	дзяленне *а* на *в*	толькі цэлы	сапраўдны
		толькі сапраўдны	сапраўдны
		адзін цэлы, другі сапраўдны	сапраўдны
а mod в	астатак ад дзялення *а* на *в*	толькі цэлы	цэлы
а div в	цэлалікавае дзяленне *а* на *в*	толькі цэлы	цэлы

Матэматычныя функцыі

Абазначэнне	***Прызначэнне***	***Тып аргумента x***	***Тып выніку y***
abs(x)	вызначае модуль велічыні *x*	цэлы	цэлы
		сапраўдны	сапраўдны
sin(x)	вызначае сінус *x*	цэлы	толькі сапраўдны
		сапраўдны	
cos(x)	вызначае косінус *x*	цэлы	толькі сапраўдны
		сапраўдны	
arctan(x)	вызначае арктангенс *x*	цэлы	толькі сапраўдны
		сапраўдны	
exp(x)	*e* узводзіцца ў ступень *x*	цэлы	толькі сапраўдны
		сапраўдны	
ln(x)	вызначае натуральны лагарыфм *x (x > 0)*	цэлы	толькі сапраўдны
		сапраўдны	
sqr(x)	узводзіць *x* у квадрат	цэлы	цэлы
		сапраўдны	сапраўдны
sqrt(x)	вызначае корань з *x (x ≥ 0)*	цэлы	толькі сапраўдны
		сапраўдны	
frac(x)	вызначае дробную частку *x*	цэлы	толькі сапраўдны
		сапраўдны	
int(x)	вызначае цэлую частку *x*	цэлы	толькі сапраўдны
		сапраўдны	

trunc(x)	вызначае цэлую частку *x*	цэлы сапраўдны	толькі цэлы
round(x)	знаходзіць цэлае, бліжэйшае да *x*	цэлы сапраўдны	толькі цэлы
odd(x)	правярае на няцотнасць *x*	цэлы	лагічны
random(x)	вызначае выпадковае цэлае значэнне	цэлы	цэлы ў межах $0 \le y < x$
random	вызначае выпадковае сапраўднае значэнне	без аргумента	сапраўдны ў межах $0 \le y < 1$
pi	вызначае лік π	без аргумента	сапраўдны

Прыклад 1. Разгледзім праграму для вылічэння значэнняў выразаў: $b = a^3$; $c = \sin(a) + \sqrt{|e|}$.

```
program umn;
    var a,b,c,e:real;
BEGIN
    writeln('ввод a e');
    readln(a,e);
    b:=a*a*a;
    c:=sin(a)+sqrt(abs(e));
    writeln('otwet b= ',b:6:2,' c= ',c:8:1);
END.
```

Тэкставы і графічны рэжымы работы экрана

Ёсць група працэдур і функцый, якія забяспечваюць аўтаматызацыю праграмавання работы з экранам і клавіятурай камп'ютэра. Трэба падкрэсліць, што экран КУВТ "Карвет" можа працаваць паралельна ў двух рэжымах – тэкставым і графічным. У тэкставым рэжыме кожны сімвал займае адно знакамесца (як у звычайнай друкавальнай машынцы). Усяго такіх знакамесцаў у радку – 64, а радкоў на экране – 15. Такім чынам, кожнае знакамесца мае дзве каардынаты: пазіцыю ў радку і нумар радка.

Ніжэй прыведзены асноўныя працэдуры і функцыі кіравання экранам у **тэкставым** рэжыме.

Працэдуры кіравання экранам

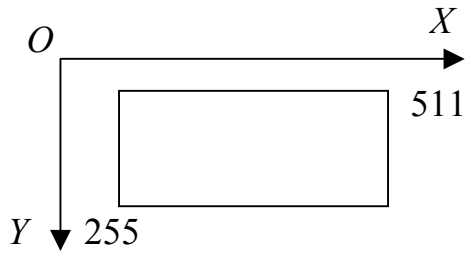
Працэдура	***Прызначэнне***
clrscr	чысціць тэкставы экран
gotoxy(x,y)	перамяшчае тэкставы курсор па ўказаным у працэдуры каардынатам

Функцыі кіравання клавіятурай

Функцыя	***Прызначэнне***
keypressed	вяртае вынік *true*, калі на клавіятуры была націснута якая-небудзь клавіша
readkey	зчытвае сімвал з клавіятуры

Графічны рэжым работы экрана

Пры пабудове розных малюнкаў, графікаў трэба ўлічваць, што кожны пункт задаецца дзвюма цэлалікавымі каардынатамі. Каардынаты задаюцца зыходзячы з наступных дамоўленасцей: пункт у левым верхнім кутку экрана мае каардынаты (0,0), вось *OX* накіравана ўправа, а вось *OY* уніз. Пункт у правым ніжнім кутку мае каардынаты (511,255).



Працэдуры кіравання графічным экранам

Працэдура	***Прызначэнне***
clrgscr	адкрыццё і чыстка графічнага экрана
setcolor(c)	устаноўка колеру для малявання з нумарам c
putpixel(x,y)	адлюстраванее пункта
line(x,y,x1,y1)	адлюстраванне адрэзка
rectangle(x,y,x1,y1,false) *rectangle(x,y,x1,y1,true)*	маляванне па зададзенай дыяганалі: контура прамавугольніка зафарбаванага прамавугольніка
circle(x,y,r,false) *circle(x,y,r,true)*	маляванне: акружнасці з зададзеным цэнтрам і радыусам круга з зададзеным цэнтрам і радыусам

Пабудаванне графічных відарысаў можа выконвацца разам з вывадам тэкставай інфармацыі.

Прывядзём табліцу кодаў колераў:

Код	0	1	2	3	4	5	6	7
Колер	чорны	сіні	зялёны	блакітны	чырвоны	ліловы	жоўты	белы

Прыклад 2. Саставім праграму малявання трохвугольніка:

```
program ris1;
BEGIN
    clrgscr;
    line(70,30,200,100);
    line(200,100,150,170);
    line(150,170,70,30);
    gotoxy(10,2);
    writeln('трохвугольнік');
END.
```