Дәріс - 7.

Тақырыбы: Функциялармен жұмыс. Режимдік функциялар. Түсті орнату. Геометриялық объектілерді сызу операторлары.

Жоспар:

1. Функция және режимдік функциялар
2. Геометриялық объектілер мен сызу операторлар
3. Фонды түсті орнату

Программа құруда алдыңғы тақырыптарда тек негізгі баскарушы main() функциясы пайдаланылып келді. C++ тілінде программаға main() функциясынан *басқа функциялардың (блоктардың)* енгізілуі де мүмкін. Олар, Турбо Паскаль тіліндегі сияқты, ішкі программалар ретінде орындалады. Мұндағы ескеретін жайт: олардың әркайсысының соңына баскарушы функцияға қайту командасын орындайтын return; операторын жазып қою керек.

Turbo C++ тілінде мұндай функциялардың мәтіні main() функциясының алдында енгізіледі.

Блоктар соңына енгізілген **return;** операторы басқаруды **main()** функциясына қайтарады да, функцияны шақыру операторының орнына осы функция орындап шығатын нәтижелік мән жазылады. Программада шақырылған функциялар тақырыптары **void blok(void)** сияқты түрде жазылған. Мұндағы **void** аргументі шақыру функциясында жіберілетін аргументтер жоқ екенін білдіреді, функция алдына енгізілген **void** типі шақырылған блокта есептелетін мәндердің алдын ала типі көрсетілмегенін білдіреді. Бұл кезде қажетті тип блок ішінде жергілікті айнымалы түрінде сипатталады (жергілікті айнымалылар блоктан шығу кезінде жойылып кетеді. Яғни, мұндай блокты пайдалану жолы Турбо Паскальдағы параметрсіз процедураны пайдалану тәсілі сияқты).

Ескерту. (25_1)-программаны мынадай құрылымдық түрде жазуға да болады:

```
{ Препроцессор нұсқаулары }
blokl();
blok2();
main() (25₂)
{ функция денесі; }
blokl()
{ блок денесі; return; }
blok2()
{ блок денесі; return; }
```

Ескерту. Құрылымға енгізілген денелердің орнына (25_1) — программада жазылған денелерді кірістіру керек.

Режимдік функциялар

Turbo C++ тілінде графикалық режімге өтіп, түрлі фигуралық кескіндерді салуға арналған стандартты фунцкиялар бар. Олар graphics.h файлында сақталып қойылған. Графикалық драйверлер (басқарушы программалар) BGI (Borland Graphics Interfase) кеңейтілуі бар файлдарда сақталған. Оны қосу арнайы initgraph() функциясы арқылы орындалады.

Графикалық режім EGA, VGA, SVGA адаптерлерінде түрлі-түрлі. Мысалы, EGA адаптері 640 X 480 графикалық режімді пайдаланады.

Графикалық режімде жұмыс істеу функциялары:

1. Драйвер мен графикалық режімді тандау: detectgraph (&gd, &gm);

2. Драйверді жүктеу және графикалық жүйені инициалдау:

initgraph (&gd, &gm, "BGI файлына жол");

Мұндағы gd - қажетті драйвер нөмірі, gd=GraphDriver; gm - пайдаланатын графикалық режім нөмірі, gm=GraphMode. (gd, gm айнымалылары программада int типі

арқылы сипатталады. Программада драйверді тандау операторының орнына gd = DETECT; операторын пайдалануға да болады. Мысалы, (30)-программаны қараңыз. DETECT — автоматты түрде анықталатын графикалық драйвер нөмірі). Егер драйвер жадта дұрыс орналастырылған болса, бұл функция жадтан 4кб көлемді графикалық буфер бөліп қояды.

Егер BGI файлдары ағымдық каталогта бар болса, программада әр жолы функцияны толық жазбай, оның үшінші параметрінің орнына бос орын (" ") жолын енгізіп кету жеткілікті:

initgraph (&gd, &gm, " ");

3. Қате жағдайды тексеру функциясы: graphresult ();

Егер программа дұрыс қосылса, онда қате коды = 0 деп есептеледі (жазылу түрі: grOK=0), ал, түрлі кателер кодтары - 0-ден басқа бүтін сандар.

Сонымен, графиктік режімді іске қосу үшін программа басын алғашқы кезде мынадай түрде жазу керек:

```
#include <stdio.h>
#include <conio.h> #include <stdlib.h>
#include <graphics.li> main ()
 // мұны көп жағдайда // енгізбеуге де болады
{
int gd, gm, error;
defectgraph (&gd, &gm); initgraph (&gd, &gm, "");
era>r=graphresult();
//(error!=grOK)
{ puts ("графика катесі"); exit (1); }
```

Ескерту. Компьютерде режімнің дұрыс істейтініне сенімді болсаңыз, графиктік режімнің бар-жоғын тексеру, (*error, if*) операторларын енгізудің қажеті жоқ.

Түсті орнату

Graphics файлында анықталған түстік палитралар

Түс аты	Код	Казақша аты
BLACK	0	қара
BLUE	1	көк
GREEN	2	жасыл-көк
GRAY	3	сұр
RED	4	қызыл
MAGENTA	5	қызыл күрең
BROWN	6	қоңыр
LIGHT GRAY	7	ашық қоныр
LIGHT GREEN	10	ашық көк
LIGHT RED	12	ашык қызыл
YELLOW	14	сары
WHITE	15	ақ

Фон түсін орнату функциясы: setbkcolor (түс):

Түс үшін оның латынша атын не кодын енгізуге болады.

Сызылатын кескіндер мен жазылатын мәтін түсін орнату:

setcolor (түс);

Геометриялық объектінің өзгешілігі мен қалыңдығын орнату:

setlinestyle (түр, үлгі, қалындық);

Мұндағы түр кодтары:

0 SOLID_LINE (тұтас)

1 DOTTED_LINE (нүктелерден туратын)

2 CENTER_LINE (нүктелер мен сызықшалардан)

3 PASHED_LINE (пунктирлермен)

4 USERBIT_LINE (пайдаланушы анықтайтын);

Қалыңдық:

1 NORM_WIDTH (бір пиксельдік сызық)

3 THICK_WIDTH (үш пиксельдік сызық)

‡лгі параметрі түр коды 4-ке етіп ашылғанда ғана беріледі, әшейінде ол үшін 0 коды енгізіледі.

Тұйық облысты бояу үшін арналған функциялар:

а) setfillstyle (бояу типі, түс);

Мұндағы бояу типінің негізгі кодтары мен атаулары:

0 EMPTY_FILL	фон түсімен бояу
1 SOLID_FILL	көрсетілген түспен тұтас толтыру
3 LTSLASH_FILL	көлбеу сызықтармен штрихтау

б) **floodfill(x,** у, түс);

(x, у) — тұйық облыс ішінде жататын нүкте, **түс** -тұйық облыс жиегінің түсімен бірдей түс (соңғы функция көбірек пайдаланылады).

Экранды тазалау (орнатылған параметрлерді алып тастап, алғашқы графикалық режімге өту):

cleardevice ();

Берілген графикалық режімде нүктелер координаттарының **ең үлкен мәндерін** анықтайтын функциялар:

getmaxx (); - көлденеңі бойынша, **getmaxy** (); - тігі бойынша.

Графикалық **экранды тазалау: clearviewport ():**

Ағымдық координаттарды анықтау: **getx(); gety();**

Графикалық режімде **жадты тазалап, алғашқы мәтіндік режімге өту:**

closegraph ();

Графикалық экран ішінде **терезе ашу:**

setviewport (xl, yl, x2, y2, clip);

-(xl, yl), (x2, y2) - терезенің сәйкес сол жақ жоғарғы төбесі мен оң жақ төменгі төбесінің координаттары;

clip - кесіп тастау коды. Егер clip=l, онда терезеге сыймайтын кескін элементтері анынып тасталады, егер clip=0 болса, терезе шекаралары ескерілмейді.

Геометриялық объектілерді сызу операторлары

1. (xl, yl), (x2, y2) нүктелерін қосатын түзу кесіндісі:

line (xl, yl, x2, y2)

ал, lineto (x, y) - функциясы ағымдық нүктеден (x, y) нүктесіне дейін түзу кесіндісін сызады.

2. Сәйкес сол жақ жоғарғы және оң жақ төменгі координаттары (xl, yl), (x2, y2) болатын тік төртбұрыш:

rectangle (xl, yl, x2, y2);

3. Орталық нүктесі (x, y) шеңбер:

circle (x, y, радиус);

4. Шеңбер доғасын сызу:

arc (x, y, баст_бїрыш, соңғ._бұрыш, радиус);

- бұрыштар градустық өлшеммен беріледі және саәат тіліне қарсы есептеледі.

Мұндағы ескеретін жайт: кескіндерді салу пиксельдік нүктелер бойынша орындалады.

5. Жартылай өстері rx, ry болатын боялған эллипсті орнату,

(x, y) — орталық нүктесі:

ғillellipse (x, y, rx, ry);

6. Эллипс доғасын сызу:

ellipse (x, y, баст_бұрыш, соңғ._бұрыш, rx, ry);

7. Боялған тік төртбұрышты салуға арналған *функция:*

bar (xl, yl, x2, y2);

- (xl, yl,), (x2, y2) - сәйкес сол жақ жоғарғы және оң жақ төменгі төбелерінің координаттары. Төртбұрыш алдын ала салынған *тұйық, облысты бояу түсімен* боялып *көрсетіледі* және контуры сызылмайды.

8. Контур іші боялған параллелепипед сызу функциясы:

bar3d (xl, yl, x2, y2, тереңдігі, төбесі);

мұнда параллелепипедтің алдыңғы жағы боялып көрсетіледі. Тереңдік үшін 20 .. 30 аралығындағы бүтін санды алу жеткілікті, төбе үшін 1 саны алынады. Ол параллелепипед төбесін сызып көрсетеді (0 саны жазылса, төбе сызылмайды).

9. Орталық нүктесі (x, y), боялған дөңгелек секторы:

pieslice(x, y, баст_бұрыш, соңғ_бұрыш, радиус);

контуры көрсетілген және боялған сектор сызылады. Боялу түсі мен сызық қалындығы алдын ала енгізілген *setlinesyle()* функциясы бойынша орнатылады.

10. Боялған эллипстік сектор:

sector (x, y, баст_бұрыш, соңғ_бұрыш, rx, ry);

11. Салынған үшбұрыш, тік төртбұрыш, шеңбер сияқты фигураларды ағымдық кезде өшіріп, фигуралардың "қозғалуын" көрсететін (анимация ұйымдастыратын) функция:

setwritemode (режім);

режім үшін алынатын кодтар:

0- көшіру;

1- экранда сызулы кескінді "өшіру";

2- стандартты режім.

Мундағы 1 (бір) - түрлі сызықтармен құрастыру кезінде шығарып тастау операциясын орындайды. Оның ерекшелігі: бір сызық тұрған орнына әкелінген кезде ол көрінбей ("өшіп") калады, одан шығу кезінде ол қайта көрінеді. Яғни, функцияны анимация ұйымдастыру үшін пайдалануға болады.

Қозғалуды ұйымдастыратын басқа функциялар да бар:

getimage(), imagesize() және putimage(). getimage() функциясы арқылы бейне экранның берілген бөлігіне жазылады да, бейнені сақтайтын жад көлемін анықтау үшін imagesize() функциясы шақырылады. Одан әрі putimage() функциясы арқылы оны экранның қалаған жеріне шақыруға болады.

12. Нүкте орнату:

putpixel (x, y, түс);

Мұндағы x, y — терезенің координаттар жүйесінде пиксель өлшемімен орнатылатын нүкте.

13. Терезеде мәтін орнату:

outtextxy (x, y, mn);

Мұндағы x, y - мәтін шығарылатын бастапқы нүкте, mn — шығарылатын жолға жазылатын мәтін мазмұны. Ол тырнақшалар ішінде жазылады. Режімде жолды ағымдық позициядан бастап шығаратын

outtext() функциясы да бар.

Төменгі функцияны пайдаланып, мәтінді түрлі түрде шығаруға болады.

settextstyle (шрифт, бағыт, өлшем);

шрифт (қаріп) параметрлерінің негізгілері:

0 DEFAULT _FONT стандартты

3 SANS_SERIF_FONT тікесінен жазу

бағыт параметрлері:

0 HORlZ_DIR солдан оңға қарай

1 VERT DIR төменнен жоғарыға

өлшем - коды 1 .. 10 аралығында алынатын бүтін сан. Стандартты шрифт үшін өлшем коды 4-ке тең етіп алынады, ол - 32 х 32 пиксельдік өлшем.

Бақылау сұрақтары

1 С++ тіліндегі функциялардың Паскалдан айырмашылығы?
2 С++ тілінде графикалық режімге өту үшін қандай функцияны қолданады?
3 Қозғалуды ұйымдастыратын қандай функциялар бар?
4 Геометриялық объектілерді сызу операторлары?

Әдебиеттер тізімі

1. Берн Страуструп. Язык программирования С++. Москва: 1999г.
2. Бабэ Б. Просто и ясно о Borland С++: пер. с англ. – СПб.: Питер, 1997.
3. Бад Т. Объектно-ориентированное программирование в действии: пер. с англ. СПБ.: Питер, 1997. – 464с.
4. Ирэ П. Объектно-ориентированное программирование с использованием С++: пер. с англ. К.: НИПФ ДиаСофтЛтд, 1995. – 480с.
5. Подбельский В.В. Язык С++. – М.: Финансы и статистика, 1966. – 558с.
6. Шамис В.А. Borland C++ Builder. Программирование на С++: пер. с англ. –М.: БИНОМ.-480с.
7. Шилдт Г. Теория и практика С++. Пер. с англ. – СПБ.: БХВ-Петербург, 1996.-416с.
8. Архангельский А.Я. Программирование в С++ Builder 6. –М.: ЗАО «Издательство БИНОМ», 2003.-1152с.
9. Б.Пахомов. С/С++ и Borland C++ 2006. Санкт-Петербург.: БХВ-Петербург., 2006.
10. Т.Павловская. С/С++. Структурное программирование. Питер. 2005.
11. Программирование на С++. Учебное пособие. Под ред. А.Д.Хомоненко. Москва.: Альтекс-А, 2003.