

Дәріс - 4.

Тақырыбы: C++Builder бағдарламасындағы жоба. Жобаның құрамы. Пішін құрастырушысы.

Жоспар:

1. SWITCH операторы. Ентаңба. BREAK операторы
2. Циклдық операторлар типтері.
3. While, for, While типті циклдық операторлары.

Тармақталу (if) операторы, процедуралық программалау тіліндегі сияқты, екі үлгіде жазылады:

1) if (шарт) оператор; /* қысқа құрылым */

2) if (шарт) операторі;

else оператор2; /* толық құрылым */

Соңғы құрылымды мынадай түрде алуға да болады:

if (шарт 1) операторі; else if (шарт 2) оператор 2;

Егер if операторының құрамында бірнеше оператор болса, олар *фигуралық жақшаларға алынып* жазылады. Мысалы

$$y = \begin{cases} 2x + 1, & x < 0, x > 2, \\ x^2 + 3, & 0 \leq x \leq 2 \end{cases}$$

функциясының мәнін есептеу керек.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main()
```

```
{
```

```
float x,y; clrscr();
```

```
printf("x=?"); scanf("%f", &x);
```

```
if (x<0 || x>2 ) y=2*x+1;
```

```
else if (x>=0 && x<=2)
```

```
{ y=x*x+3; printf(" Нәтиже:"); }
```

```
printf("y=%f", y); getch();
```

```
}
```

Процуралық программалау тілдерінде **GOTO** (сөзсіз өту) операторы пайдалана берілмейтіні, бірақ ол ентаңбаға (метка, label) сілтеме ретінде қолданылатыны белгілі. Ентаңбаға сілтеме тек бір блок (функция) ішінде пайдаланылады; ентаңба жөнінде кеңірек түсінік келесі тақырыпта берілген.

Программаға енгізілген **puts** операторы **printf** операторы сияқты орындалады, бірақ ерекшелігі: ол тек мәтін шығаруда пайдаланылады және енгізілген мәтінді шығарып болған соң *курсорды жаңа жолдың басына* орналастырып қояды. Мысалы төмендегі функция мәнін есептеу керек:

$$y = \begin{cases} x + 4, & x \leq 2 \\ 3x + 8, & 2 \leq x \leq 5 \\ 2x - 1, & x > 5 \end{cases}$$

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main()
```

(7)

```
int x, y; clrscr();
```

```
printf("x=?"); scanf("%d", &x);
```

```
if(x<2) {y=x+4; goto j;}
```

```
if (x >=2 && x<=5) {y=3*x+8; gotoj; }
```

```
if(x>5) y=3*x-1;
```

```
j: printf("y=%d", y);
```

```
getch(); return 0;
}
```

SWITCH операторы. Ентаңба. BREAK операторы

C++ тілінде тармақталу тобынан тұратын және ентаңбамен белгіленген деректердің (берілгендердің) бірін тандау үшін **switch** операторы, сұрыптаушы үшін **case** операторы пайдаланылады (*switch* - ауыстырып қосқыш, *case* жағдай). **Switch** операторының жазылу үлгісі:

```
switch (өрнек)
{
case <тұрақты1> : операторлар;
break;
case <тұрақты2 >: операторлар;
break;
.....
case < тұрақтыN > : операторлар;
break;
default: операторлар;
}
```

Мұндағы **тұрақтылар** — ентаңбалар ($k=1, 2, \dots, N$). **Ентаңба** — оператор алдында жазылатын (оператор жолын белгілейтін) бүтін сан, символ не символдар тізбегінен тұратын атау. Ентаңба соңына қос нүкте (:) қойылып жазылады. Мысалдар (мұндағы ентаңбалар: 10, 'C'):

```
case 10: m=3;
case 'C': printf("Ура!");
```

Ентаңбаға өту үшін **switch (өрнек)** операторы пайдаланылады; мұндағы **өрнек** - мәні айқындалатын айнымалы не математикалық өрнек.

Switch операторының орындалуы әдеттегідей алгоритмдік тілдердегі сияқты: алдымен **switch** кілттік сөзінен соң жақшалар ішінде жазылған өрнек мәні есептеледі не айнымалы мәні оқылады. Ол ентаңбалар енгізілген сұрыптаушылармен салыстырылады да, жүйе басқаруды ентаңба мәніне сәйкес келетін операторға (операторлар блогына) өткізеді. Егер оқылған мән бірде-бір ентаңбаға тең болмаса, басқару **default** (жоқ болғанда) кілттік сөзінен соң жазылған операторға беріледі.

Үлгіде әр **case** операторынан кейінгі жолда **break** (тоқтату) операторы енгізілген. Ол сәйкес **case** операторы орындалған соң, оны тоқтатады да, жүйе басқаруды **switch** операторынан соң жазылған программаның келесі операторына өткізеді.

Мысалы адам аттарынан тұратын латын әріптерімен жазылған тізім берілген: Erjan, Jamila, Marjan, Sultan, Marat. Алғашқы бас әріптері бірдей болатын аттарды анықтау керек.

```
#include <stdio.h>
#include <conio.h>
main()
{
char ch; int ml; clrscr();
ml : printf("Латынша бас әріп енгізіңіз-?");
ch=getchar(); /* getch() - ағымдық келесі символды оқу операторы */
if(ch='A' && ch='Z');
switch (ch)
{
case 'E' : printf("Erjan\n"); break;
case 'J' : printf("Jamila\n"); break;
case 'M' : printf("Marat, Marjan\n"); break;
case 'S' : printf ("Sultan\n"); break;
default : printf ("Енгізілген 1-бас әріптік атау жоқ"); " : goto ml;
} getch(); return 0;
```

Циклдық операторлар типтері. While, for, While типті циклдық операторлары.

C++ тілінде циклдік операторлардың үш түрі бар:

for, while, do - while.

Құрама цикл денесі фигуралық жақшалар ішінде жазылады, ал дене бір оператордан тұрса, оны жақшаларға алу міндетті емес.

for циклінің жазылу үлгісі:

for (өрнек1; өрнек2; өрнек3) { операторлар }

өрнек1 — циклді инициалдау. Карапайым түрде **ол** — параметрге бастапқы мән меншіктеу комавдасы.

өрнек2 — циклдің аяқталу шарты. Ол ақиқат кезде циклдің денесі орындалып, басқару ретімен **3-өрнекке** өтеді.

өрнек3 — параметрдің өзгеру шамасы (цикл қадамы).

Turbo C++ тілінде параметр бүтін сан болса, ол 1-ге өседі не 1-ге кемиді. Мысалы, параметр **k** арқылы белгіленсе, оның өзгеруінің жазылу түрі: **k++** не **k--**

++ - **инкремент (өсімше) операторы** деп аталады. Ол өз операндын 1-ге өсіреді (**k++** және **k=k+1** операторлары бірдей).

-- - **декремент (төмендету) операторы** делінеді. Ол өз операндын 1-ге кемітеді (**k--** және **k=k-1** операторлары бірдей).

FOR цикліне енгізілген әр өрнек нүктелі үтір (;) таңбасымен ажыратылып жазылады. Мысалы :

```
for (K=0; K<10; k++) printf (" %d\n", k);
```

программасының үзіндісі баған бойынша 0, 1, 2, 3, ..., 9 сан-дарын, ал,

```
for (K=9; k>=0; k--) printf (" %d\n", k);
```

үзіндісі баған бойынша 9, 8, ..., 2, 1, 0 сандарын басып шығарады. Цикл параметрін символдық етіп алуға да болады. Мысалы,

```
for (ch='A'; ch='R'; ch++) printf (" %c ", ch);
```

циклі кезегімен A = дан R-ге дейін латын алфавитінің бас әріптерін басып шығарады.

C++ тілінде функция бөлімінде пайдаланылатын бүтін тұрақтының орнына айнымалы арқылы белгіленген атауын пайдалануға да болады. Ол үшін тұрақтыны main() функциясының алдында **CONST** стандартты операторы (спецификаторы) арқылы сипаттап қою керек. Сипаттау үлгісі:

```
const <атау>= <мән>
```

мысалы, **const ms =12;** Тұрақтының типі оның мәнінің типімен анықталады. Мәнді айнымалы атауы арқылы сипаттаудың да қатесі жоқ, мысалы, **const ms;**

1-мысал. For циклін пайдаланып, функциясының мәнін есептеу керек.

```
y=x5
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
const a=5;
```

```
main()
```

```
{
```

```
float x, y; int k; clrscr();
```

```
puts("x мәні : x=?");
```

```
scanf("%f", &x); y=1;
```

```
for(k=1; k<=a; k++)
```

```
y=x*y;
```

```
printf ("дәреже= %f\n", y); getch (); return 0;
```

```
}
```

Цикл шексіз болуы да мүмкін. Мысалы, **for (k=10; k>8; k++)**

Шексіз не жай циклден **break** операторын пайдаланып шығуға болады.

While, do-while циклдердің жазылу үлгісі (*while* — әзірше, *do* -орындау):

1) while (шарт) { операторлар }

2) do

{ операторлар } while (шарт)

while циклінің денесінде бір ғана оператор бар болса, оны фигуралық жақшаларға алмай жазуға да болады.

Мысалы жалпы мүшесі $a_k = k * k$ ($k=1,2,\dots$) болатын тізбектің алғашқы алты мүшесінің қосындысын табу керек.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main() (131)
```

```
int k, a, s; clrscr();
```

```
while (k<=6) { a=k*k; s=s+a; k++; }
```

```
printf("s=%d\n", s);
```

```
getch(); return 0;
```

```
}
```

```
// Нәтиже : s = 91
```

Программа денесінің while блогын do — while құрылымымен дайындауға да болады:

do

```
{ a=k*k; s=s+a; k++; } (132)
```

```
while (k <= 6 );
```

While циклінен ерте шығу үшін break не goto операторын пайдалануға болады.

Бір цикл екінші циклдің ішінде орналасса, олар бір-біріне салынған циклдер не цикл ішінде цикл делінетіні белгілі. C++ тілінде сыртқы цикл денесі фигуралық жақшаларға алынып жазылады. Егер ішкі цикл де бірнеше операторлардан тұрса, ол да осындай түрде жазылады.

Мысалы (9 X 9) көбейту кестесін дайындау керек.

```
#include <stdio.h>
```

```
#include <conio.h> main()
```

```
{
```

```
int k, j; clrscr();
```

```
for(k=1; k<10; k++ )
```

```
{
```

```
for(j=1; j<10; j++) printf("%d * %d=%d\n",
```

```
k, j, k*j);
```

```
getch(); return 0;
```

continue (жалғастыру) — for, while, do — while циклдерінің кез келгенінде пайдаланатын оператор. Оның орындалу түрі: ол оқылған соң жүйе цикл денесінің қалған бөліктерін орындауды тоқтатады да, циклді қайталап орындайды, яғни басқаруды циклдің жаңа кдцамын орындауға өткізеді. Бұл оператордың орындалуы **break** операторының орындалуына қарама-қарсы.

Мысалы 1 .. 150 натурал сандарының ішінен 9-ға еселі сандарды табу керек.

```
#include <stdio.h>
```

```
#include <conio.h> .
```

```
main() (15)
```

```
int k; clrscr();
```

```
for (k=1; k<=150; k++) {if (k % 9) continue; printf("%12d\n",k); } getch (); return 0;
```

```
// Нәтиже (баған бойынша шығатын сандар): 9 18 27 ... 144
```

Ескерту. printf операторының форматтық параметріне **12d** символы енгізілген. Мұндағы 12 саны нәтижені экранның сол жағынан 12 позициялық орын тастап шығару керектігін ғана көрсетеді.

Бакылау сұрактары

- 1 (if) операторының екі үлгіде жазылуы?
- 2 SWITCH операторы?
- 3 Break циклдік конструкциясы?
- 4 C++ тілінде циклдік операторлардың түрі?

Әдебиеттер тізімі

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