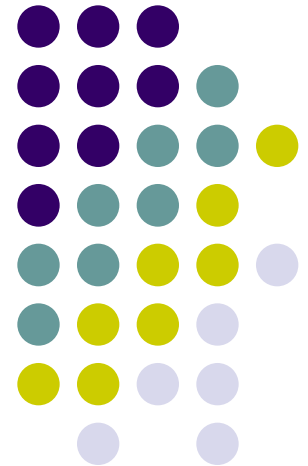


# Оператори мови C++

---

Тема 3





# Оператори дії

- Оператор-вираз

```
a = 1;
```

```
b = 3;
```

```
m = max(a, b);
```

```
x + y - 12;
```

```
func(d, 12, x);
```

- Оператор оголошення імен

```
int x;
```

```
double f;
```

```
const float pi = 3.1415;
```

- Складений оператор

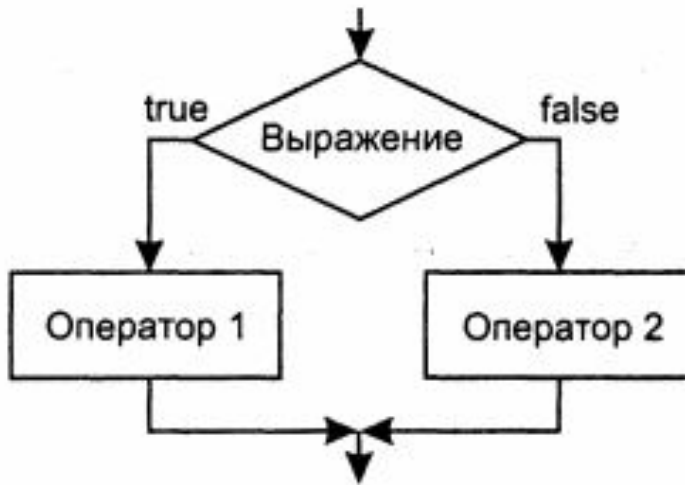
```
{b = 3;
```

```
m = max(a, b); }
```

# Оператори керування . Розгалуження



- Умовний оператор



# Оператори керування . Розгалуження



- Умовний оператор

**if (умова) оператор1 else оператор2**

```
if (x > y)
```

```
  a = x;
```

```
  else  a = y;
```

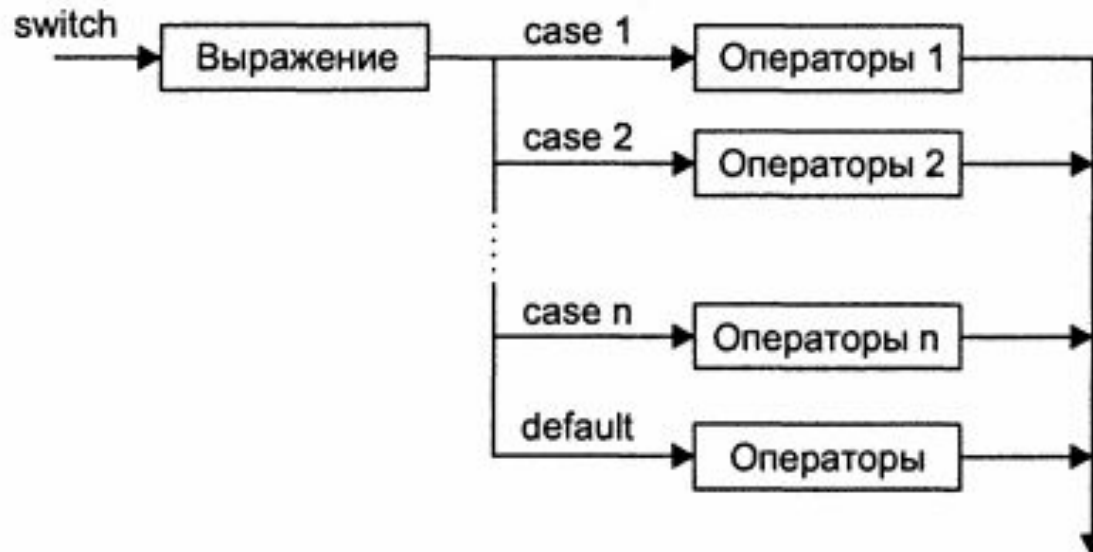
```
if (x < 0)
```

```
{  x = -x;
```

```
  cout << "Змінити значення x на протилежне за  
  знаком";}
```

```
  abs = x;
```

# Оператор выбора



# Оператор вибору



Синтаксис:

```
switch (вираз)
{
case c1_1:
<оператори_1>
break;
.....
case cn_1:
[case cn_2: ...]
<оператори_n>
break;
default:
<оператори>
break;
}
```

# Оператор вибору



Правила запису:

- switch-вираз повинен мати значення, сумісне з цілим типом.
- Значення `i_j` у case-константах мають бути константами типу switch-виразу.
- case-константи не можуть виражатись діапазоном.
- Оператори у case-частинах закінчуються оператором `break`, що забезпечує вихід із switch-блоку. Якщо `break` відсутній, то виконуються оператори наступної case-частини.
- default-частина не є обов'язковою. Вона присутня, якщо перевіряються усі можливі значення switch-виразу. Якщо перевіряються окремі значення switch-виразу, то default-частина не указується. default-частина може розміщуватись у довільній частині switch-блоку.
- Фігурні дужки у case-частині не обов'язкові.



# Оператор вибору

- Приклад:

**switch** (code)

**{case 0:**

cout << "код нуль"; x = x + 1; break;

**case 1 :**

cout << "код один"; y = y + 1; break;

**case 2:**

cout << "код два"; z = z + 1; break;

**default:**

cout << "Значення не обробляється " ;}



# Оператори керування .

## Повторення



- Цикл з параметром  
**for (ініц. парам.; умова продовж.; зміна парам.)**  
**тіло циклу**

```
for (i = 1; i <= 100; i = i + 1)           sum  
    = sum + i;
```

```
for(printf("Введіть число! \n"); num != 6; )  
scanf("%d",&num);
```

# Оператори керування . Повторення

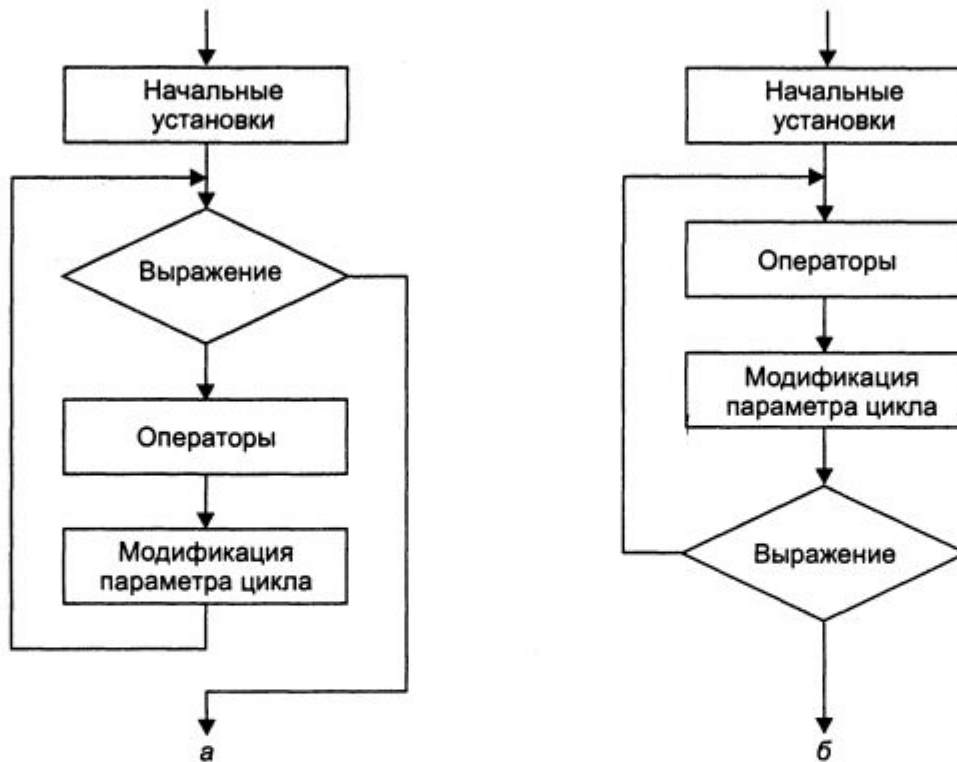


Рис. 1.8. Структурные схемы операторов цикла: а — цикл с предусловием;  
б — цикл с постусловием

# Оператори керування . Повторення



- Цикл з передумовою  
**while (умова) оператор;**

```
char ch;  
ch = getch();  
while (ch != '*')  
{ ch = getch();}
```

# Оператори керування . Повторення



- Цикл з постумовою

**do оператор while ( умова);**

do

```
scanf("%d", &number);
```

```
while (number!=50);
```

```
do { ch = getch(); } while (ch != '*');
```



# Інші оператори керування

- Оператор ***break***

```
while((ch=getchar()) != '*')  
  { if(ch=='\n') break;  
    putchar(ch); }
```

- Оператор ***continue***

```
while((ch=getchar()) != '*')  
  { if(ch=='\n') continue;  
    putchar(ch); }
```



# Інші оператори керування

- Оператор переходу

**goto мітка;**

```
if ( x >= 0) goto positiv;
```

```
  x = -x;
```

```
  positiv: abs = x;
```

- Оператор повернення

**return вираз;**

```
return a*b;
```

```
return 1;
```

```
return;
```