

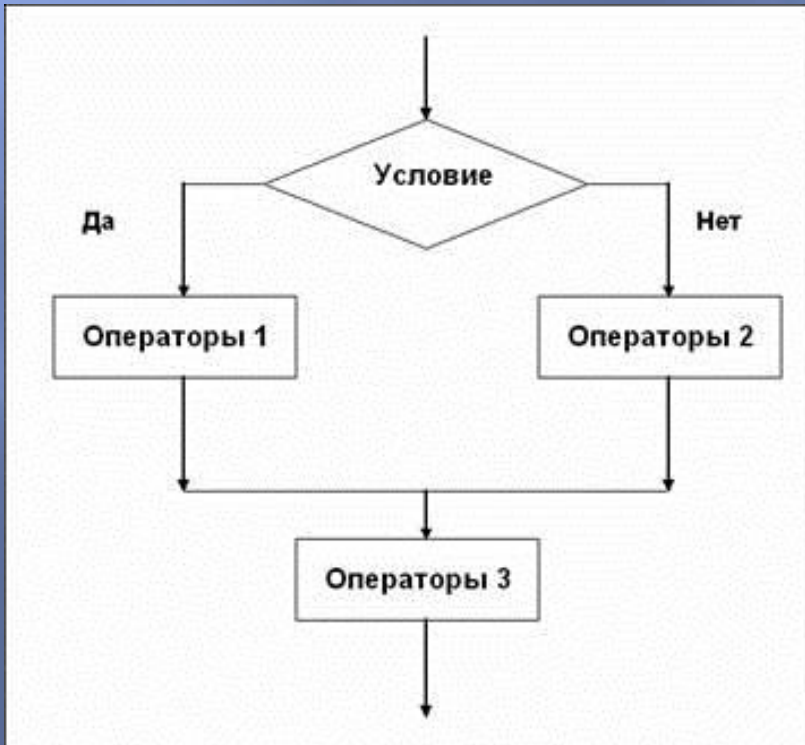
Лекція 2. Керуючі інструкції

Керуючі інструкції – операторні конструкції, за допомогою яких реалізуються точки розгалуження

Умовний оператор `if()`

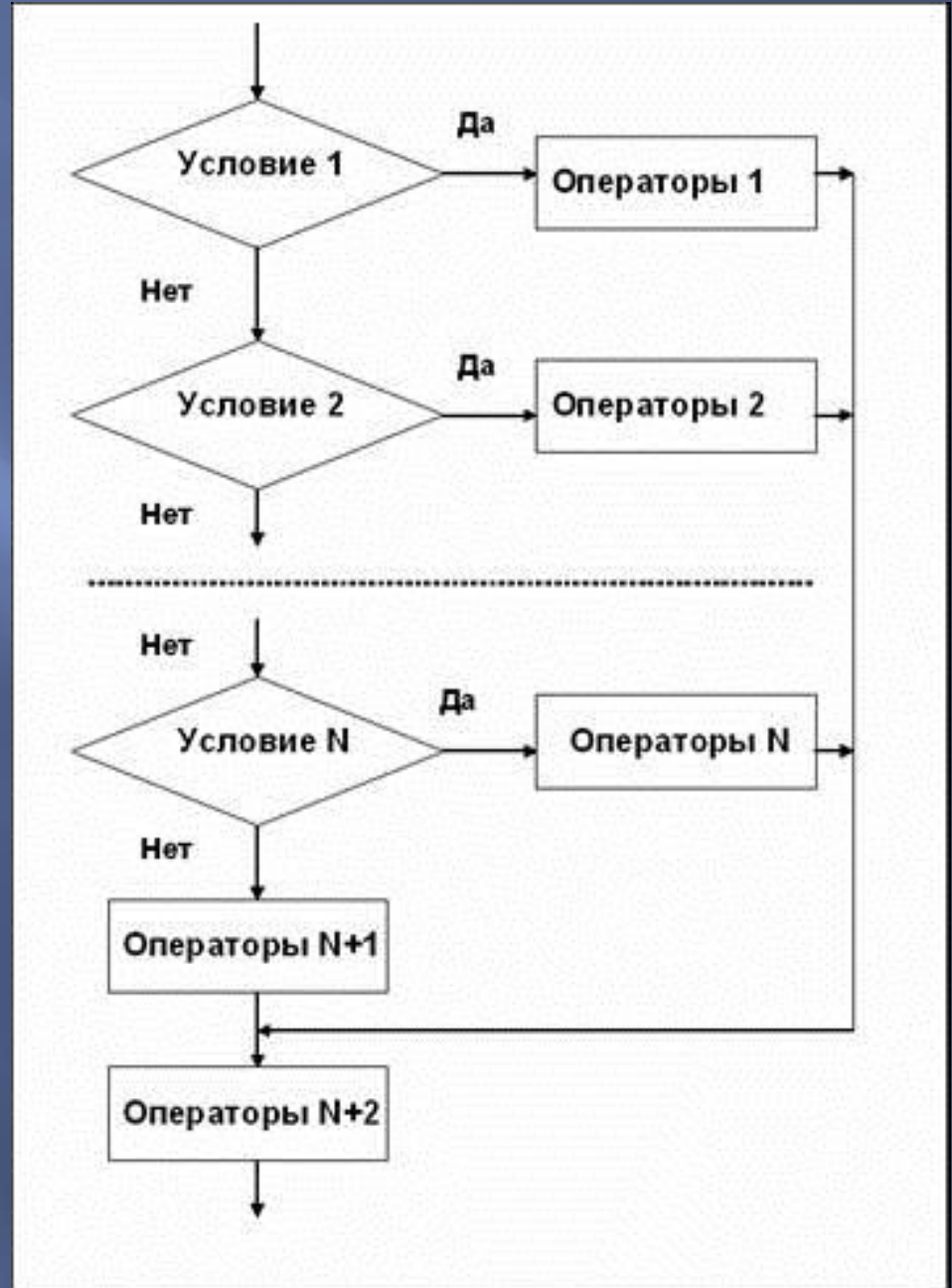
`if(умова) {команди1}`
`else {команди2}`

`if(умова) {команди}`



Блок з вкладених операторів `if()`

```
if(умова1) {команди1}  
else if(умова2) {команди2}  
else if(умова3) {команди3}  
.....  
else if(умоваN) {командиN}  
else {команди(N+1)}
```



Приклад програми “Ділення на нуль” з оператором `if()`

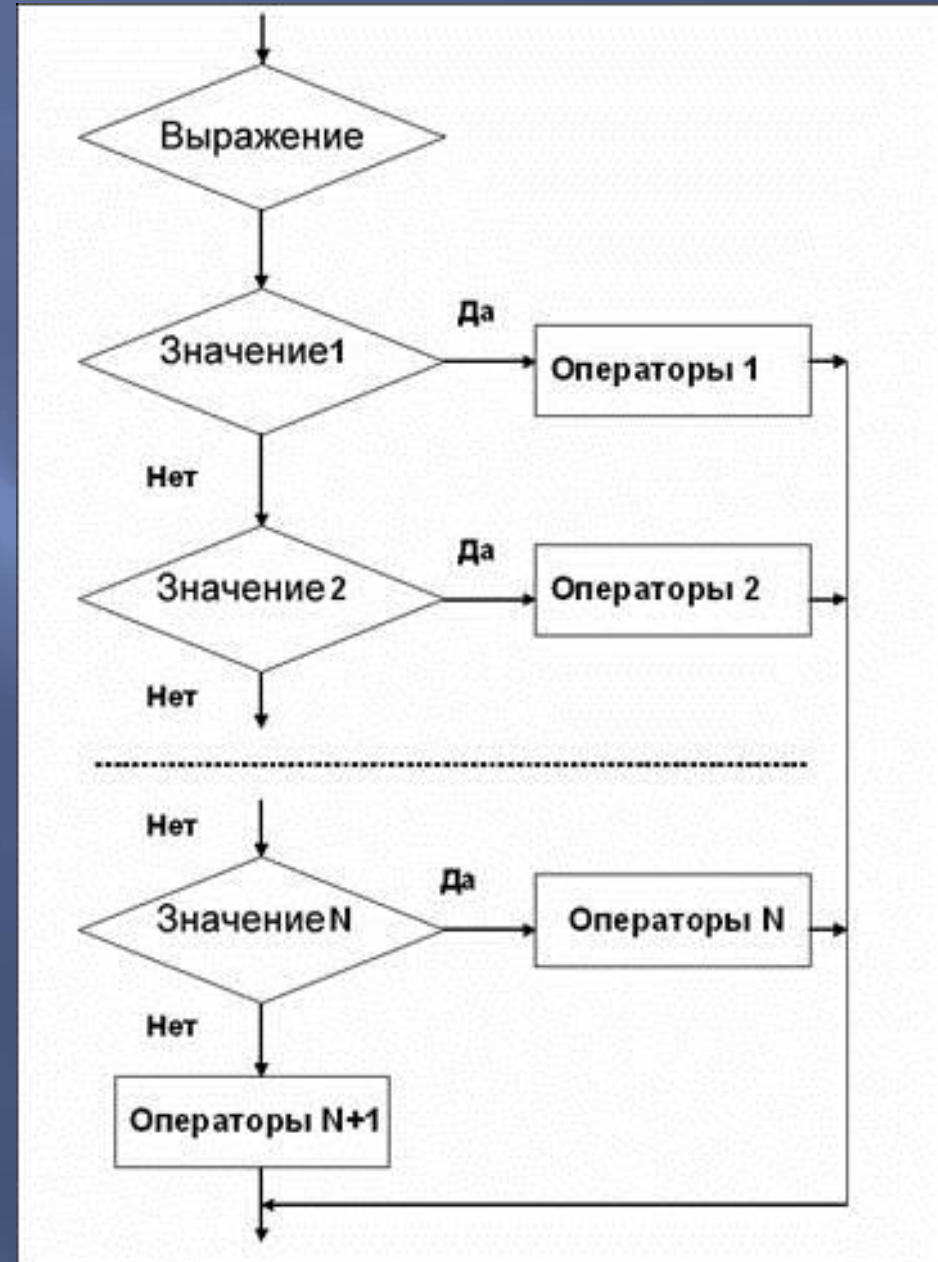
```
#include <iostream>
using namespace std;
int main(){
double x,y;
cout<<"x=";
cin>>x;
cout<<"y=";
cin>>y;
if(y!=0) cout<<"x/y ="<<x/y<<"\n"; // if(y) cout<<"x/y ="<<x/y<<"\n";
else cout<<"Ділення на нуль!\n";
return 0;
}
```

Приклад програми “Вгадування числа” з оп-м `if()`

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(){
int n,m;
n=rand()%100+1;
cout<<"Введіть число m=";
cin>>m;
cout<<"n="<<n<<": ";
if(m>n) cout<<"ваше число більше!\n";
else if(n>m) cout<<"ваше число менше!\n";
else cout<<"ви вгадали число!\n";
return 0;
}
```

Умовний оператор switch()

```
switch(вираз){  
  case значення1:  
    команди1  
    break;  
  case значення2:  
    команди2  
    break;  
  ...  
  default:  
    команди  
}
```



Приклад програми з оператором `switch()`

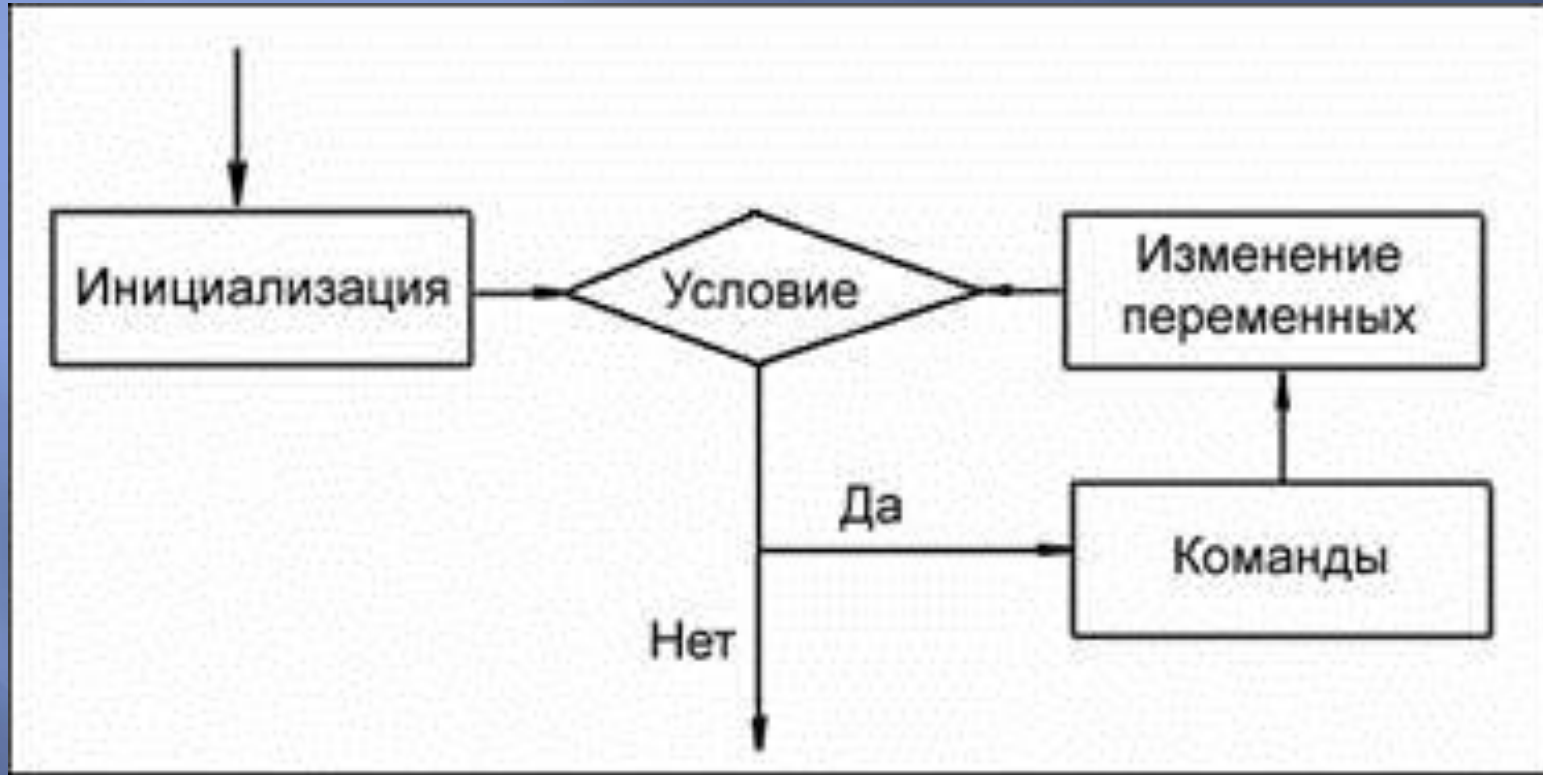
```
int main(){
int n;
cout<<"Введіть n=";
cin>>n;
switch(n){
case 1:
cout<<"Перший case-блок\n";
break;
case 2:
cout<<"Другий case-блок\n";
break;
case 3:
cout<<"Третій case-блок\n";
break;
default:
cout<<"За умовчанням\n";}
return 0;}
```

Приклад програми “Числа Фібоначі” з оп-м `switch()`

```
int main(){
int n;
cout<<"Введіть число від 0 до 10: ";
cin>>n;
switch(n){
case 0:
cout<<"Користувачем введений нуль!\n";
break;
case 1:
case 2:
case 3:
case 5:
case 8:
cout<<"Введене число Фібоначі!\n";
break;
default:
cout<<"Введене ціле число!\n";}
return 0;}
```

Оператор цикла `for()`

`for(ініціалізація; умова; зміна значень){команди}`



Приклад програми “Сума чисел” з оп-м циклу `for()`

```
#include <iostream>
using namespace std;
int main(){
int n,i,s=0;
cout<<"Введіть число n=";
cin>>n;
for(i=1;i<=n;i++){
s+=i;
}
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад програми “Сума чисел - 2” з оп-м циклу `for()`

```
#include <iostream>
using namespace std;
int main(){
int n,i=1,s=0;
cout<<"Введіть число n=";
cin>>n;
for(;i<=n;i++){
s+=i;
}
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад програми “Сума чисел - 3” з оп-м циклу `for()`

```
#include <iostream>
using namespace std;
int main(){
int n,i=1,s=0;
cout<<"Введіть число n=";
cin>>n;
for(;i<=n;){
s+=i;
i++;
}
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад програми “Сума чисел - 4” з оп-м циклу **for()**

```
#include <iostream>
using namespace std;
int main(){
int n,i=1,s=0;
cout<<"Введіть число n=";
cin>>n;
for(;;){
s+=i;
i++;
if(i>n) break;
}
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад програми “Сума чисел - 5” з оп-м циклу **for()**

```
#include <iostream>
using namespace std;
int main(){
int n;
cout<<"Введіть число n=";
cin>>n;
for(int i=1, s=0;i<=n;s+=i++);
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад пр-ми “Вгадування букви” з оп-м циклу `for()`

```
#include <iostream>
using namespace std;
int main(){
for(char x='a';x!='z';){
cout<<"Вгадайте букву: ";
cin>>x;
}
cout<<"Ви вгадали!\n";
return 0;
}
```

Приклад пр-ми “Натуральні числа” з оп-м циклу `for()`

```
#include <iostream>
using namespace std;
int main(){
int i,j;
for(i=1;i<=3;i++){
for(j=1;j<=5;j++) cout<<3*(j-1)+i<<" ";
cout<<"\n";}
return 0;}
```

Результат:

```
1 4 7 10 13
2 5 8 11 14
3 6 9 12 15
```

Приклад пр-ми “Дві індексні змінні” з оп-м циклу **for()**

```
#include <iostream>
using namespace std;
int main(){
int i,j;
for(i=10,j=90;i<j;i+=5,j-=10)
cout<<i<<" "<<j<<"\n";
return 0;}
```

Результат:

10 90

15 80

20 70

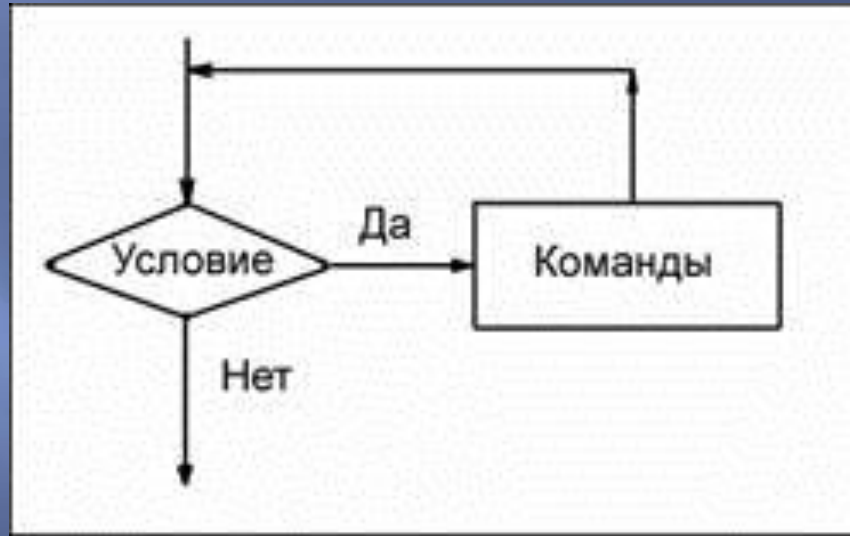
25 60

30 50

35 40

Оператор цикла while()

```
while(умова){  
  команди  
}
```



```
do{  
  команди  
}  
while(умова);
```



Приклад програми “Сума чисел - 6” з оп-м `while()`

```
#include <iostream>
using namespace std;
int main(){
int n,i=1,s=0;
cout<<"Введіть число n=";
cin>>n;
while(i<=n){
s+=i;
i++;}
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад програми “Сума чисел - 7” з оп-м **do-while()**

```
#include <iostream>
using namespace std;
int main(){
int n,i=1,s=0;
cout<<"Введіть число n=";
cin>>n;
do{
s+=i;
i++;
}while(i<=n);
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;
}
```

Приклад пр-ми “Сума чисел - 8” з інстр. **goto**

```
#include <iostream>
using namespace std;
int main(){
int n,i=1,s=0;
cout<<"Введіть число n=";
cin>>n;
// Використовується мітка
mylabel:
s+=i;
i++;
// Команда з інструкцією безумовного переходу
if(i<=n) goto mylabel;
cout<<"Сума натуральних чисел: "<<s<<"\n";
return 0;}
```

Приклад “Розрахунок синуса”

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + \frac{(-1)^n x^{2n-1}}{(2n-1)!} + \dots$$

```
#include <iostream>
using namespace std;
//Границя ряду:
const int N=100;
int main(){
//Аргумент функції і “робочі” змінні:
double x,q,s=0;
//Індексна змінна:
int n;
cout<<"Enter x = ";
cin>>x;
q=x;
//Розрахунок синуса:
for(n=1;n<=N;n++){
s+=q;
q*=(-1)*x*x/(2*n)/(2*n+1);}
//Результат:
cout<<"sin("<<x<<" ) = "<<s<<endl;
return 0;}
```

Резюме

1. Умовний оператор **if()** дозволяє виконувати різні блоки команд в залежності від того, виконується певна умова, чи ні. Умова вказується аргументом оператора **if()**, після чого йде блок команд, які виконуються при істинній умові. Команди, що виконуються при хибній умові вказуються після ключового слова **else**.

2. В умовному операторі **switch()** перевіряється значення деякого виразу (аргумент оператора **switch()**). В залежності від значення виразу виконуються різні блоки команд. Можливі значення виразу (число чи символ) вказуються після ключового слова **case**. В блоці **default** розміщуються команди, що виконуються у випадку, коли жодне з представлених в **case**-блоках значень не реалізоване.

3. Оператор циклу **for()** зазвичай використовують, коли необхідно виконати певну кількість раз блок команд. Аргумент оператора **for()** складається з трьох блоків. Блоки розділяються крапкою з комою. Зазвичай в першому блоці виконується ініціалізація індексної змінної, в другому вказується умова, що перевіряється, а в третьому виконується зміна індексної змінної. Блок команд розміщують в фігурних дужках після інструкції **for()**.

Резюме (продовження)

4. В операторе циклу **while()** перевіряється умова, передана аргументом оператору. Якщо умова істинна, виконується блок команд після інструкції **while()** (команди розміщені в фігурних дужках). Умова перевіряється після кожного циклу. Робота оператора завершується, коли умова стає хибною. Оператор має модифікацію **do-while()**. Принципова відмінність полягає в тому, що в першому випадку спочатку перевіряється умова, а в другому спочатку виконуються команди.

5. В C++ існує інструкція безумовного переходу **goto**, що дозволяє переходити до поміченого міткою місця програми. Інструкцію можна використовувати для реалізації циклів.