



Курс ОСНОВИ – C#

ЗАНЯТТЯ №3 - УМОВНІ ОПЕРАТОРИ ТА ЦИКЛИ



A = 10



A < 5

Так?

Ні?

УМОВНІ оператори

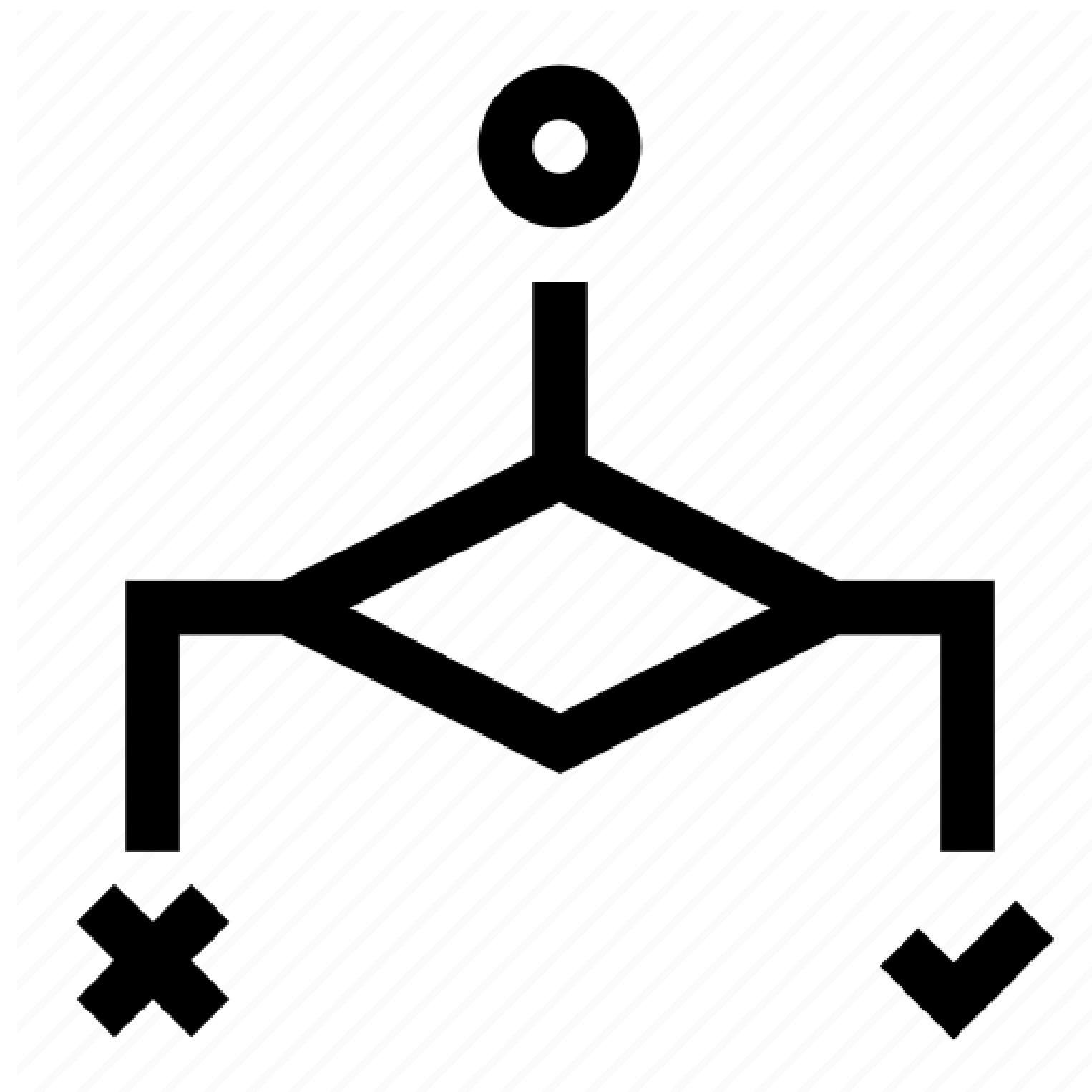
Умовні оператори – це конструкції, що дозволяють керувати ходом виконання програми, залежно від певних умов. У мові C # є два типи таких конструкцій: if...else і switch...case .

Оператор if

Конструкція if...else дозволяє перевірити деяку умову на істинність і, залежно від результату, виконати той чи інший блок коду. Синтаксис умовного оператора if...else :

```
if ( Умова_1 )  
{  
  // Блок виконається, якщо Умова 1 має значення true  
}  
else if ( Умова_2 ) // Необов'язкова частина  
{  
  // Блок виконається, якщо Умова 2 має значення true  
}  
else // Необов'язкова частина  
{  
  // Блок виконається, якщо Умова 1 та Умова 2 мають значення false  
}
```

Як умова може бути змінна типу bool , чи вираз, значення якого має тип bool .

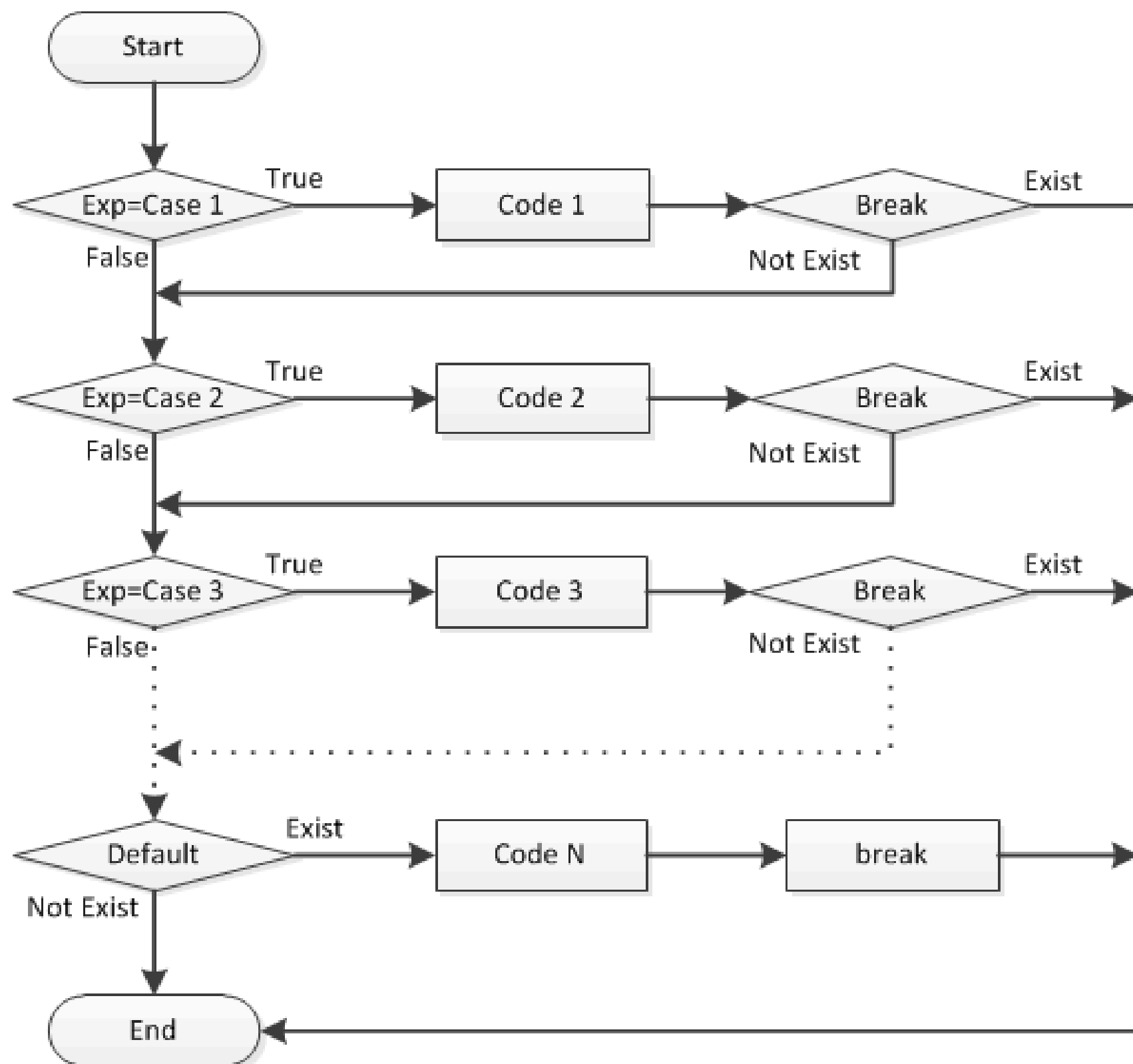


Оператор switch

Конструкція switch...case використовується, коли необхідно, залежно від конкретного значення змінної чи результату певної операції, виконати необхідний блок коду.

Синтаксис конструкції має такий вигляд:

```
switch ( Вираз )
{
    case Шаблон_1:
        break ;
    case Шаблон_2:
        break ;
    .....
    default :
        break ;
}
```



`result = (условие) ? выражение1 : выражение2;`

условное выражение

условие проверки

условный оператор

Тернарный оператор

Результатом обчислення тернарного оператора є одна з двох альтернатив, яка вибирається залежно від істинності умови, що перевіряється.

Синтаксис оператора має такий вигляд:

`condition? if_true : if_else`

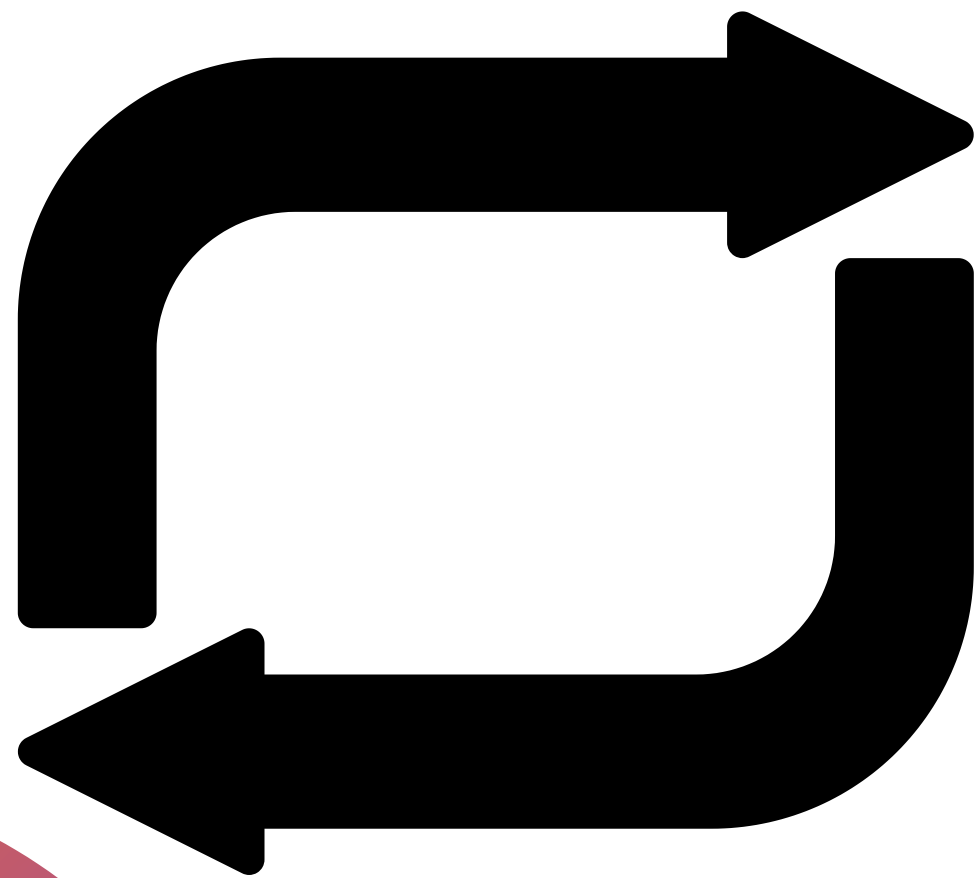
Залежно від значення логічного виразу `condition` повертається результат обчислення виразу `if_true` або `if_else`. Нижче наведено приклад, що демонструє роботу цього оператора:

```
int n8 = 5 ;
```

```
int n9 = 8 ;
```

```
int largerNumber = n8 > n9? n8: n9;
```

У прикладі за допомогою оператора `?` визначається, яке з двох чисел більше, і відповідне значення присвоюється змінною `largerNumber`.



Цикли

Цикли в мовах програмування призначені для побудови конструкції, що виконує заданий блок коду кілька разів, що визначається тією чи іншою умовою. С # надає чотири різні варіанти побудови циклів.

Цикл for

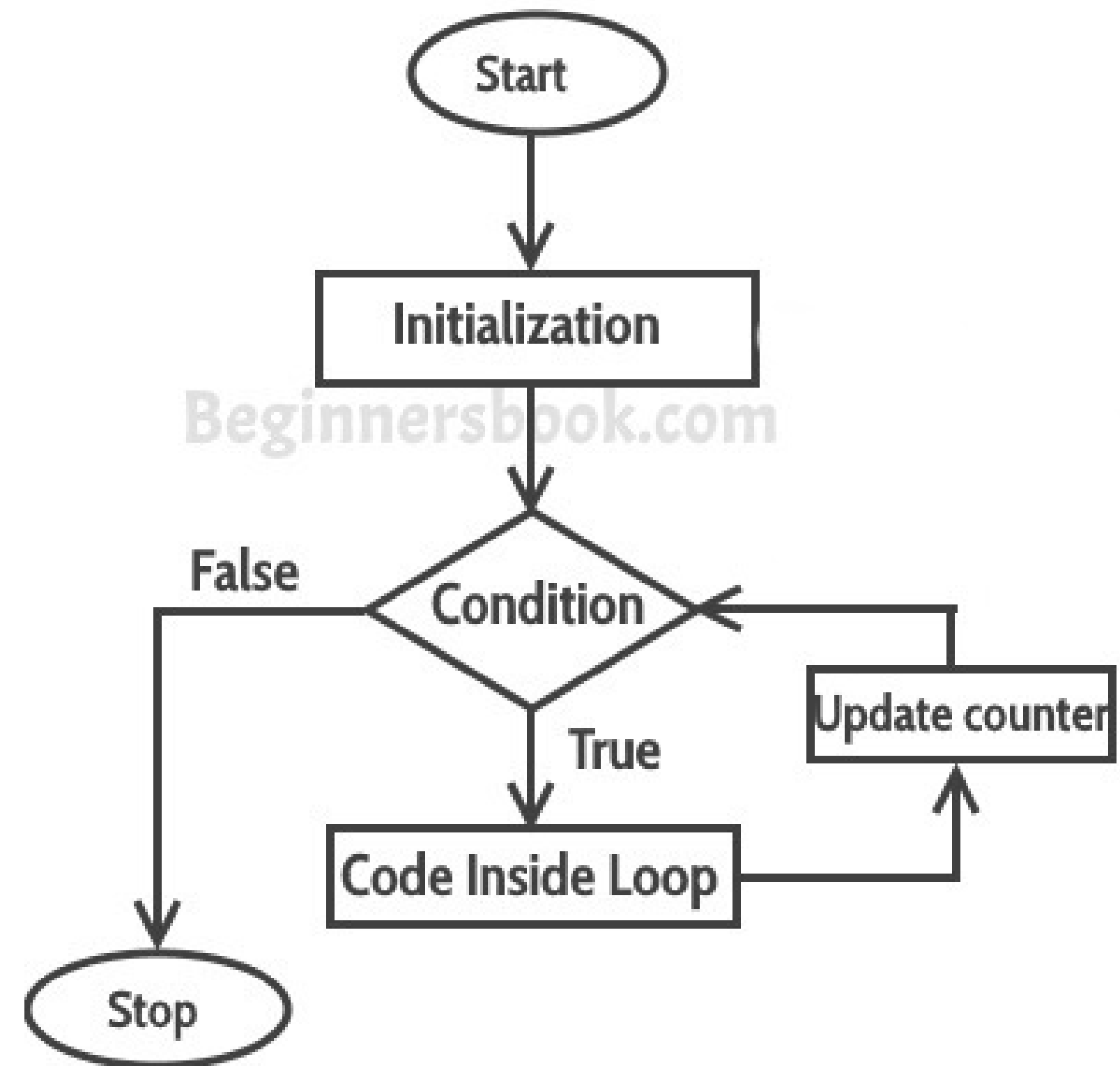
Цикл for має наступний синтаксис:

```
for ( ініціалізація лічильника; умова; ітератор )  
{  
    // оператор (набір операторів)  
}
```

Ініціалізатор – це вираз, який обчислюється перед виконанням тіла циклу. Зазвичай тут ініціалізується локальна змінна, яка використовуватиметься як лічильник.

Умова - це вираз, який перевіряється перед кожною новою ітерацією циклу. Якщо значення виразу дорівнює true , то буде виконано тіло циклу, якщо false - відбудеться вихід із циклу і виконається наступна інструкція.

Ітератор - це вираз, що обчислюється після кожної ітерації. Зазвичай тут відбувається зміна локальної змінної, оголошеної в ініціалізаторі.





Цикли while та do/while

У C # цикл while має таку конструкцію:

```
while ( умова )  
{  
    // Тіло циклу  
}
```

У цьому циклі перевіряється умова, і якщо воно є істинним, то виконується набір операторів усередині тіла циклу. Обов'язково переконайтеся, що змінюєте змінну, від якої залежить умова, інакше цикл стане нескінченним, якщо, звичайно, це не є метою. Приклад роботи з циклом while :

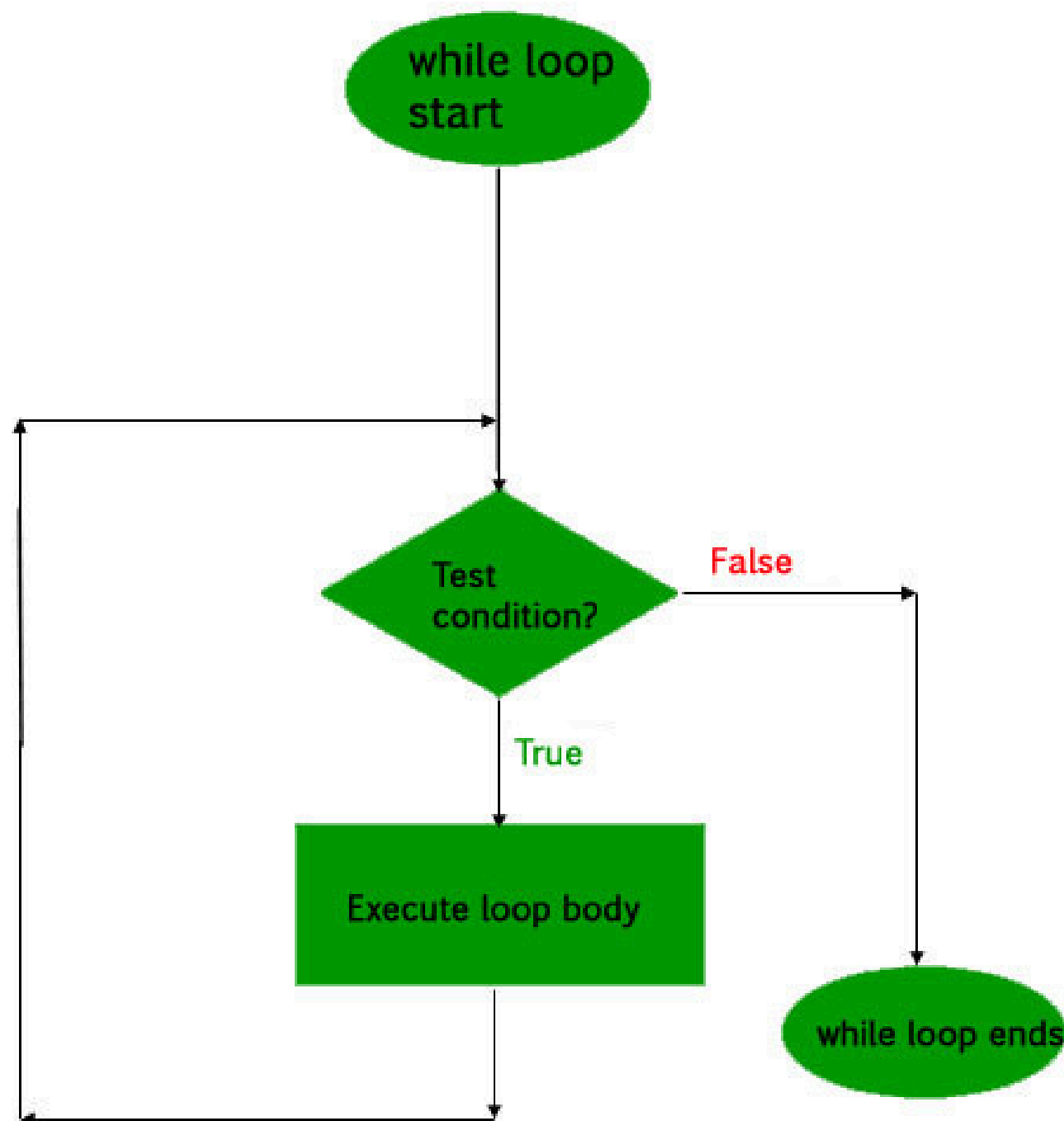
```
int i = 0 ;  
int number = 3 ;  
while ( i <= number ) {  
    Console.WriteLine ( "Ітерація циклу номер" + i ) ;  
    i++;  
}
```

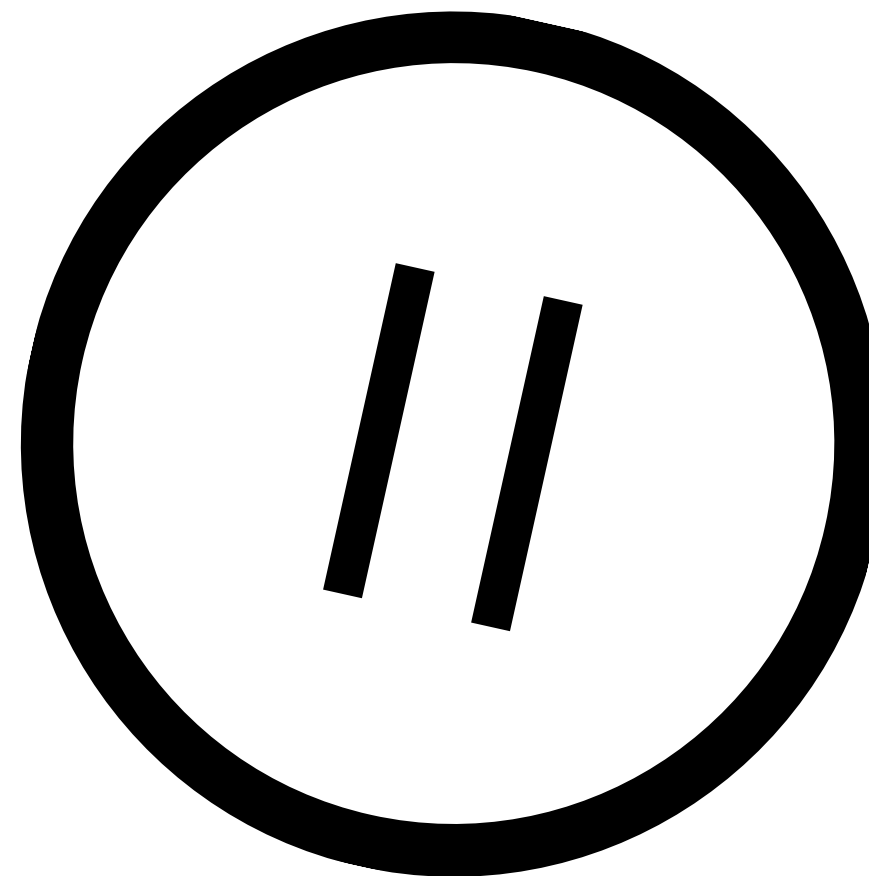
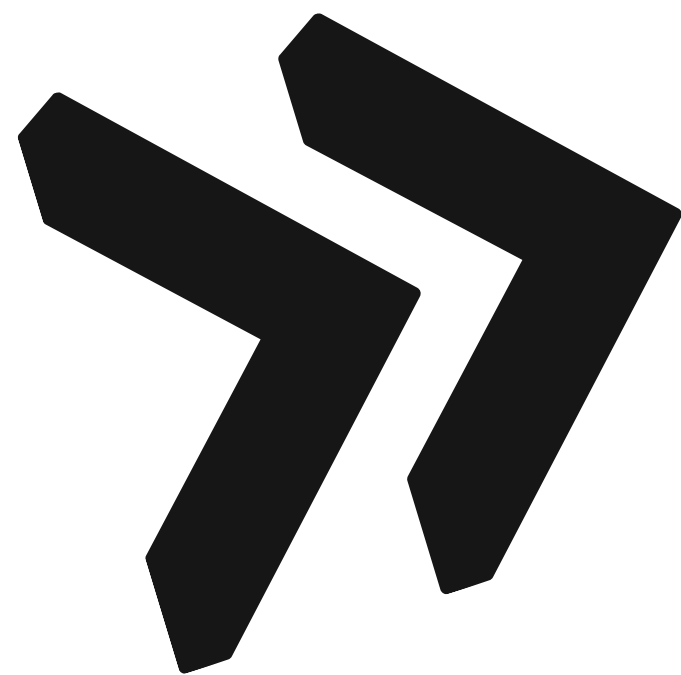
Console.ReadKey () ;

Цикл do/while має наступний синтаксис:

```
do {  
    //Тіло циклу  
} while ( умова ) ;
```

Відмінність do/while полягає в тому, що перевірка умови відбувається після тіла циклу, що призводить до того, що незалежно від умови цикл виконається хоча б один раз.





Оператори переходу

Мова C # надає спеціальні оператори для переривання виконання всього циклу та для примусового завершення поточної ітерації з переходом до наступної. Перше завдання вирішує оператор `break`. Якщо в програмі використовується кілька вкладених циклів, то при використанні `break` вихід буде виконаний тільки з того циклу, де цей оператор був викликаний.

Оператор `continue` примусово завершує поточну ітерацію циклу та переходить до наступної. При цьому для циклу `while` та `do/while` відбувається перехід до умовного виразу, а в циклі `for` спочатку обчислюється ітераційний вираз, а потім перевіряється умова



Дякую за увагу