Оператори за цикъл

КН, УП Практикум 2ра група 2023/2024, Богомил Стоянов, Виолета Кастрева

1. Оператор for
   В C++ for цикълът е контролна структура, която ви позволява да изпълнявате блок от код многократно за определен брой итерации или докато не бъде изпълнено определено условие. Цикълът for има следния синтаксис:

```
for (initialization; condition; update) {
    // Код за изпълнение на всяка итерация
}
```

   Нека разбием компонентите на for цикъл:

   Инициализация: Тази част се изпълнява веднъж, преди да започне цикълът. Обикновено се използва за инициализиране на контролна променлива за цикъл (често цяло число) до начална стойност. Например:

```
for (int i = 0; i; i++) {
    // Инициализация: int i = 0;
    // Код за изпълнение на всяка итерация
}
```

   Условие: Това е булев израз, който се оценява преди всяка итерация. Ако условието е вярно, цикълът продължава; ако е невярно, цикълът прекратява. Когато условието стане невярно, цикълът спира. Например:

```
for (int i = 0; i < 5; i++) {
    // Condition: i < 5
    // Код за изпълнение на всяка итерация
}
```

   Актуализация: Тази част се изпълнява след всяка итерация и обикновено се използва за актуализиране на контролната променлива на цикъла. Обичайно е да се увеличава или намалява контролната променлива. Например:

```
for (int i = 0; i < 5; i++) {
    // Увеличаваме: i++
    // Код за изпълнение на всяка итерация
}
```

   Кодът във фигурните скоби {} е блокът от код, който ще бъде изпълнен при всяка итерация. Можете да имате всеки валиден C++ код в този блок. Това е

мястото, където дефинирате какво искате да правите във всяка итерация на цикъла.

Ето пример за прост цикъл for, който отпечатва числа от 1 до 5:

```
#include <iostream>

int main() {
    for (int i = 1; i <= 5; i++) {
        std::cout << i << " ";
    }

    return 0;
}
```

Този код инициализира i = 1, проверява дали i е по-малко или равно на 5 и увеличава i с 1 при всяка итерация. Той ще отпечата числата от 1 до 5 на конзолата.

Можете да използвате for цикли за широк набор от задачи, включително итерация през масиви, обработка на данни и извършване на повтарящи се задачи с известен брой итерации.

2. Оператор while
В C++ цикълът while е контролна структура, която ви позволява многократно да изпълнявате блок от код, докато дадено условие остава вярно. Използва се в ситуации, в които искате да продължите да изпълнявате част от код, докато определено условие вече не бъде изпълнено. Основният синтаксис на цикъла while е както следва:

```
while (condition) {
    // Код за изпълнение, докато условието е вярно
}
```

Ето разбивка на компонентите на цикъла while:

Условие: Това е булев израз, който се оценява преди всяка итерация. Ако условието е вярно, цикълът продължава; ако е невярно, цикълът прекратява. Цикълът ще продължи да се изпълнява, докато условието остава вярно.

Блок: Кодът, ограден във фигурните скоби {}, е блокът от код, който ще се изпълнява при всяка итерация, докато условието е вярно.

Ето пример за прост цикъл while, който брои от 1 до 5:

```
#include <iostream>

int main() {
    int i = 1;
    while (i <= 5) {
        std::cout << i << " ";
        i++;
    }

    return 0;
}
```

В този пример инициализираме променлива i = 1. Цикълът while проверява дали i е по-малко или равно на 5 преди всяка итерация. Докато това условие е вярно, цикълът продължава. Вътре в цикъла отпечатваме стойността на i и след това увеличаваме i с 1. Този цикъл ще отпечата числата от 1 до 5 на конзолата.

Бъдете внимателни, когато използвате цикли while, тъй като ако условието никога не стане невярно, това може да доведе до безкраен цикъл, който може да блокира вашата програма. За да избегнете това, уверете се, че условието в крайна сметка ще стане невярно.

Можете също да използвате цикли while в ситуации, в които не знаете предварително колко итерации са необходими, но искате да продължите да повтаряте, докато не бъде изпълнено определено условие. Например, можете да използвате цикъл while за непрекъснато четене на въведени данни, докато не бъде натиснат конкретен клавиш или за многократно извършване на действие, докато определено условие стане истина.

3. Оператор do/while
   Do/while цикълът е друг тип структура на цикъл, който ви позволява многократно да изпълнявате блок от код, докато дадено условие остава вярно. За разлика от цикъла while, цикълът do/while гарантира, че кодовият блок ще се изпълни поне веднъж, дори ако условието първоначално е невярно. Основният синтаксис на do/while цикъл е както следва:

```
do {
    // Код за изпълнение
} while (condition);
```

Ето разбивка на компонентите на do/while цикъл:

Блок: Кодът, ограден във фигурните скоби {}, е блокът от код, който ще бъде изпълнен поне веднъж.

Условие: Това е булев израз, който се оценява след всяка итерация. Ако условието е вярно, цикълът продължава; ако е невярно, цикълът прекратява. Цикълът ще продължи да се изпълнява, докато условието остава вярно.

Ето пример за прост do/while цикъл, който брои от 1 до 5:

```
#include <iostream>

int main() {
    int i = 1;
    do {
        std::cout << i << " ";
        i++;
    } while (i <= 5);

    return 0;
}
```

В този пример ние инициализираме променлива i = 1 и след това влизаме в блока do. Изпълнява се кодовият блок, който отпечатва стойността на i и я увеличава с 1. След като кодовият блок се изпълни, цикълът проверява условието (i <= 5). Докато условието е вярно, цикълът продължава и кодовият блок се изпълнява отново. Този цикъл ще отпечата числата от 1 до 5 на конзолата.

Една ключова характеристика на do/while цикъла е, че той гарантира поне едно изпълнение на кодовия блок, дори ако условието първоначално е невярно. Това може да бъде полезно в ситуации, в които искате да се уверите, че определена задача е изпълнена, преди да проверите условието.

Въпреки това, подобно на други структури на цикъл, трябва да внимавате с условието, за да избегнете потенциални безкрайни цикли, тъй като цикълът ще продължи да се изпълнява, докато условието е вярно.

4. Накратко:
   Ето обобщение на трите често срещани типа цикли в C++:

   -For:
   Цикъл for се използва за изпълнение на блок от код определен брой пъти.
   Той има три части: инициализация, условие и актуализация.
   Той е идеален за ситуации, в които знаете точния брой необходими повторения.

Контролната променлива на цикъла обикновено се инициализира, проверява спрямо условие и се актуализира при всяка итерация.

```
for (int i = 0; i < x; i++) {
    // Код за изпълнение на всяка итерация
}
```

-While:
Цикълът while се използва за многократно изпълнение на блок от код, докато дадено условие остава вярно.
Условието се проверява в началото на всяка итерация.
Подходящо е, когато броят на итерациите не е известен предварително и може да се изпълни нула или повече пъти.

```
while (condition) {
    //Код за изпълнение, докато условието е вярно
}
```

-Do/while:
Цикълът do/while се използва за многократно изпълнение на блок от код поне веднъж и след това, докато дадено условие остава вярно.
Условието се проверява в края на всяка итерация, като се **гарантира поне едно изпълнение на кодовия блок**.
Полезно е, когато искате да гарантирате, че дадена задача е изпълнена, преди да проверите условието.

```
do {
    // Код
} while (condition);
```

Всеки от тези цикли има своите силни страни и случаи на използване, така че изборът на правилния зависи от специфичните изисквания на вашата програма и проблема, който се опитвате да разрешите.