**Оператори вибору if та else в C ++**

**Теоретична частина**

 Перед тим, як приступити до вивчення цієї теми, вам вже треба добре знати, що таке логічні операції.

Оператори вибору дозволяють прийняти програму рішення, грунтуючись на істинність або хибність умови. Якщо умова істинна (тобто вона true) означає те, що оператор в тілі if виконується, після чого виконується наступний по порядку оператор. Якщо умова помилкова (тобто вона false) означає, що оператор в тілі if не виконується (ігнорується або пропускається) і відразу ж виконується наступний по порядку оператор. Умова, що перевіряється, може бути будь-яким логічним виразом. Логічні вирази можуть бути утворені операціями рівності і відносини.

| **Операція в С++** | **Умова** | **Смисл записаних умов в С++** |
| --- | --- | --- |
| **==** | a**==**b | a дорівнює b |
| **!=** | a **!=** b | a не дорівнює b |
| **>** | a **>** b | a більше b |
| **<** | a **<** b | a менше b |
| **>=** | a **>=** b | a більше або дорівнює b |
| **<=** | a **<=** b | a менше або дорівнюєb |

Дуже частою помилкою програмування є неправильний запис операцій рівності і співвідношень. Прогалин між знаками ==; ! =; > =; <=; бути не повинно. Також необхідно пам'ятати правильну послідовність запису символів операцій рівності і відносин.

 приклад:

 =! неправильно; ! = Правильний запис;

 => Неправильно; > = Правильна запис;

 = <Неправильно; <= Правильний запис.

 Не можна плутати операцію = (операція присвоювання) і операцію == (операція рівності), оскільки це призводить до логічних помилок, тобто програма навіть скомпілюється без всяких помилок, але працювати буде не правильно.

Ну що ж - приступимо до наших операторам вибору (вони ж умовні оператори, вони ж оператори розгалуження). Ми розглянемо декілька простих прикладів, які допоможуть вам познайомитися з цими операторами і застосовувати їх в подальшому вже в більш складних програмах.

 **Приклад:** Користувач вводить два числа і програма визначає яке з цих двох чисел більше.

|  |  |
| --- | --- |
| **1****2****3****4****5****6****7****8****9****10****11****12****13****14****15****16****17****18****19****20****21****22****23****24****25****26****27****28****29****30****31****32****33****34****35****36****37****38****39** | **#include <iostream>****using namespace std;****int main()****{** **setlocale(LC\_ALL, "rus");** **int variable1 = 0;** **int variable2 = 0;** **cout << "Введите первое число: ";** **cin >> variable1;** **cout << "Введите второе число: ";** **cin >> variable2;** **// сравниваем полученные значения** **// если первое число больше второго - выводим сообщение**  **// если это будет не так, код в фигурных скобках не выполнится** **// и программа перейдет к проверке второго условия if**  **if (variable1 > variable2)**  **{** **cout << "\nРезультат сравнения: Первое число больше чем второе\n";** **}** **// если и это условие не выполняется - переход к следующему if** **if (variable1 < variable2)** **{** **cout << "\nРезультат сравнения: Второе  число больше чем первое\n";**  **}** **// проверяем значения на равенство. Не забывайте об операторе ==** **// если написать variable1 = variable2, то значение**  **// variable2 запишется в переменную variable1** **if (variable1 == variable2)**  **{** **cout << "\nРезультат сравнения: Числа равны\n";** **}** **return 0;****}** |

До 14-го рядка все зрозуміло – користувач ввів два числа, вони записалися у відповідні змінні. Як же їх тепер порівняти і показати результат порівняння? Для цього ми застосуємо оператор if (перекладається -ЯКЩО). Пишемо if, далі у круглих дужках () записуємо умову – порівняння значень змінних (variable1 > variable2).

І якщо variable1 дійсно більше variable2 – це умова поверне true (істина). Тоді виконання програми перейде до коду, який розміщений всередині фігурних дужок {} – рядок 21 – 23. Тобто на екрані ми побачимо повідомлення, про те що перше число більше другого.

Якщо ж умова в круглих дужках () не виконується (повертає false (хибність), то код у фігурних дужках {} буде проігноровано і виконання програми продовжиться з наступного рядка коду під блоком if – з рядка 26. Там ми знову перевіряємо відношення змінних між собою.

Все аналогічно – якщо умова істина – код виконається цього блоку if, якщо хибна – перехід нижче. У рядку 33 перевірка останнього можливого варіанту – перевірка на рівність. Обов'язково застосовуємо оператор ==(дорівнює), а не = (привласнити). Інакше код цього блоку if буде працювати з помилкою. Наприклад variable1 = 3, а variable2 = 4 , то якщо ми помилково запишемо в умови if (variable1 = variable2) , значення 4 запишеться у variable1.

Отримуємо if (4). А будь-яке значення відмінне від 0, як ви пам'ятаєте, компілятор сприймає як true . У підсумку цей блок виконається. і ми побачимо повідомлення про те, що значення рівні, хоча це спочатку було не так. Будьте уважні.

Цей код можна значно скоротити. По-перше, можна додати оператор else (подивимося нижче). По друге – якщо блоки else if або містять тільки один рядок коду у фігурних дужках {} , то ці фігурні дужки можна не писати. Подивимося, як зміниться наш код:

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324 | **#include <iostream>****using namespace std;****int main()****{** **setlocale(LC\_ALL, "rus");** **int variable1 = 0;** **int variable2 = 0;** **cout << "Введите первое число: ";** **cin >> variable1;** **cout << "Введите второе число: ";** **cin >> variable2;** **if (variable1 > variable2) // как и в первом примере** **cout << "\nРезультат сравнения: Первое число больше чем второе\n";** **else if (variable1 < variable2) // иначе если variable1 < variable2** **cout << "\nРезультат сравнения: Второе  число больше чем первое\n";**  **else // иначе** **cout << "\nРезультат сравнения: Числа равны\n";** **return 0;****}** |

Ви бачите, що фігурні дужки в цьому прикладі ми вже не використовуємо. Ще один плюс - якщо перша умова if виконується, то інші блоки if else і else, вже не перевіряються (тобто час на виконання програми скорочується). Ну а якщо перша умова невірне, програма йде в наступний блок else if (інакше якщо), і перевіряє його умова - (variable1 <variable2) - рядок 18. Якщо і тут невірно, то виконається останній блок else, в якому умова можна вже не перевіряти, так як ми до цього перебрали всі можливі варіанти відносини двох значень і залишається тільки їх рівність.

 Результат роботи цих програм однаковий. Але в другому випадку і код компактніший, і роботи компілятору менше.



Важливо запам'ятати:

- синтаксис if:

 

- синтаксис if else:

 

або якщо необхідно більше умов:

 

- конструкція блоків if else - нерозривна. Якщо вона зустрічається в коді, її не можна розділяти стороннім кодом. наприклад:

 

 - принцип роботи оператора вибору if - якщо умова в круглих дужках істина (true), то код блоку виконається. Якщо хибна (false) - ігнорується і виконання програми починається з рядка коду, наступної під блоком if.

 - умова оператора вибору в круглих дужках може бути звичайним: if (variable> 0) або комбінованим: if (variable> 0 && variable <100). Комбінована умова поверне true, якщо обидві умови будуть вірними.

 - умова оператора вибору може бути арифметичним виразом: if (variable1 - variable2). Код блоку if виконається, якщо результат обчислення не буде дорівнює 0. Наприклад: if (5 - 7), if (2 + 5). А ось якщо if (2 -2) - код блоку буде проігнорований.

 - якщо блоки if, else if або else містять тільки один рядок коду, фігурні дужки можна не використовувати. Але для новачків, поки ви вчитеся, рекомендую використовувати фігурні дужки завжди.

Подивіться ці відео. У них розглянуті умовні оператори if і else:

<https://www.youtube.com/watch?v=BypNQczB86o>

<https://www.youtube.com/watch?v=RDUqvXapk4E>

**Практична частина**

1. Дано два цілих числа, кожне число записано в окремому рядку. Виведіть найбільше з даних чисел.

2. Придумати самостійно завдання з використанням if або if-else та скласти алгоритм його рішення.

**Завдання виконати до 13.04.2020 р. включно. Відповіді** **надсилати на електронну пошту** **paleeva88@ukr.net** **у вигляді файлів CodeBlocks або файлу Word з прописаним кодом у вікні CodeBlocks.**