НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++ (ч.3 – условные операторы)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://berkut.homelinux.com

Зачем ветвление?

- Способность управлять программным потоком позволяет делать выборочное выполнение отдельных участков кода, а это весьма ценная особенность программирования.
- Оператор выбора **`if`** позволяет нам выполнять или не выполнять определенные участки кода, в зависимости от того является ли истинным или ложным условие этого оператора.

Зачем ветвление?

Очевидный пример использования if — это проверка введенного пользователем пароля:

- если пароль правильный, программа разрешает пользователю совершить какое-то действие;
- если пароль введен неправильно, то программа не позволит пользователю получить доступ к ограниченным ресурсам.

Логические значения C/C++

Логические значения в C/C++:

- ИСТИНА – TRUE – целое число, не-ноль
- ЛОЖЬ – FALSE – 0 (ноль)

Логические значения C/C++

Примеры:

Выражение **3 == 2** вернет значение **0**, так как три не равно двум.

Выражение **5 == 5** оценивается как истинное и вернет значение **1**.

Логические значения C/C++

Запись	Пояснение	Пример
==	равно	5 == 5 это истина
!=	не равно	3 != 2 и это истина
>	больше	7 > 6 — истина
>=	больше или равно	1 >= 1 истина
<	меньше	5 < 5 — ложь
<=	меньше или равно	3 <= 2 — ложь

Структура if

```
if ( условное выражение )
   // тут располагается один оператор, который выполнится,
   //если условное выражение - истинно
```

Пример использования

```
if ( 7 > 6 )
   cout << ”Семь больше шести”;
```

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Структура if

Структура if с фигурными скобками:

```
if ( условное выражение ) {
   // тут располагаются операторы, которые нужно выполнить,
   // если условие истинно
}
```

Пример использования

```
if ( 7 > 6 ) {
   cout << ”Семь больше шести”;
}
```

- Если не использовать фигурные скобки, то к телу оператора if будет относиться только один, первый оператор.
- Если же необходимо управлять несколькими операторами, то необходимо поместить их в фигурные скобки.

Структура if

Иногда, когда условное выражение — ложное, было бы удобно, чтобы выполнялся некоторый код, отличный от того кода, который выполняется при **ИСТИННОМ** условии.

```
if ( условное выражение ) {
    /* выполняется этот код, если условие истинно  */
}
else {
    /* выполняется этот код, если условие ложно  */
}
```

Конструкция else if

Обычно операторы **`else if`** используют, когда необходим множественный выбор, то есть например определены несколько условий, которые одновременно могут быть истинными, но нам необходимо только одно истинное условное выражение.

```
int age;
cout << "Сколько вам лет? " );
cin >> age;
if ( age < 100 ) {
    cout << "Вы очень молоды!\n";
} else if ( age == 100 ) {
    cout << "Молодость уже позади\n";
} else {
    cout << "Столько не живут\n";
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Использование логических операторов

- Логические операторы позволяют создавать более сложные условные выражения.
- Например, если вы хотите проверить, является ли ваша переменная больше 0 но меньше 10, в таком случае вам достаточно воспользоваться логическим оператором «И»

```
var > 0 and var < 10
```

- В языках C/C++ есть точно такой же оператор, только обозначается он иначе — &&

```
if (var > 0 && var < 10) {
    // что-то сделать
}
```

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Логические операторы С/С++

В С++ существует три логические операции:

- Логическая операция И - &&, нам уже известная;
- Логическая операция ИЛИ - ||;
- Логическая операция НЕ - ! или логическое отрицание.

Операции	Обозначение	Условие	Краткое описание
И	**&&**	a == 3 **&&** b > 4	Составное условие истинно, если истинны оба простых условия
ИЛИ	**||**	a == 3 **||** b > 4	Составное условие истинно, если истинно, хотя бы одно из простых условий
НЕ	**!**	**!**(a == 3)	Условие истинно, если а не равно 3

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Логические операторы C/C++

```
#include <iostream>
using namespace std;

int main()
{
    bool a1 = true, a2 = false; // объявление логических переменных
    bool a3 = true, a4 = false;
    cout << "Tablica istinnosti log operacii &&" << endl;
    cout << "true  && false: " << ( a1 && a2 )   << endl // логическое И
         << "false && true: "  << ( a2 && a1 )   << endl
         << "true  && true: "  << ( a1 && a3 )   << endl
         << "false && false: " << ( a2 && a4 )   << endl;
    cout << "Tablica istinnosti log operacii ||" << endl;
    cout << "true  || false: " << ( a1 || a2 )   << endl // логическое ИЛИ
         << "false || true: "  << ( a2 || a1 )   << endl
         << "true  || true: "  << ( a1 || a3 )   << endl
         << "false || false: " << ( a2 || a4 )   << endl;
    cout << "Tablica istinnosti log operacii !"  << endl;
    cout << "!true: " << ( ! a1 ) << endl // логическое НЕ
         << "!false: "<< ( ! a2 ) << endl;
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Логические операторы C/C++

```
Tablica istinnosti log operacii &&
true  && false: 0
false && true: 0
true  && true: 1
false && false: 0
Tablica istinnosti log operacii ||
true  || false: 1
false || true: 1
true  || true: 1
false || false: 0
Tablica istinnosti log operacii !
!true: 0
!false: 1
Для продолжения нажмите любую клавишу . . .
```

Оператор switch

В C++ еще имеется оператор множественного выбора **switch**:

```
// форма записи оператора множественного выбора switch
switch (/*переменная или выражение*/) {
    case /*константное выражение1*/:
    {
        /*группа операторов*/;
        break;
    }
    case /*константное выражение2*/:
    {
        /*группа операторов*/;
        break;
    }
    //.    .    .
    default:
    {
        /*группа операторов*/;
    }
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Оператор switch

```
switch (op) {
      case 1: {
          cout << a << " + " << b << " = " << a + b << endl;
          break;
      }
      case 2: {
          cout << a << " - " << b << " = " << a - b << endl;
          break;
      }
      case 3: {
          cout << a << " * " << b << " = " << a * b << endl;
          break;
      }
      case 4: {
          cout << a << " / " << b << " = " << a / b << endl;
          break;
      }
      default: // если op равно любому другому значению
      cout << "Invalid input" << endl;
}
```

Демонстрация

Спасибо!

Вопросы?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++ (ч.3 – условные операторы)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://berkut.homelinux.com