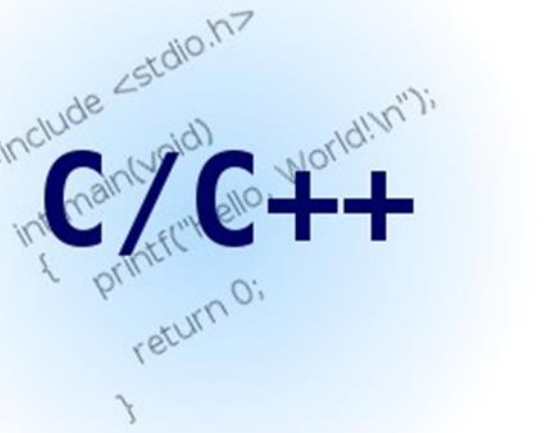
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++
(ч.1 – введение и условные операторы)



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

Язык программирования C++

- C++ является ISO-стандартизированным ЯП
  - В течение некоторого времени, C++ не имел официального стандарта, однако с 1998 года, C++ был стандартизирован комитетом ISO
- C++ компилируемый язык
  - C++ компилируется непосредственно в машинный код, что позволяет ему быть одним из самых быстрых в мире языков
- C++ является строго типизированным ЯП
  - C++ подразумевает, что программист знает, что делает, и позволяет невероятное количество возможностей, ограниченных только лишь фантазией
- C++ поддерживает множество парадигм
  - C++ поддерживает процедурную, обобщённую, и объектно-ориентированную парадигмы программирования…
- C++ является полностью совместимым с языком Си
  - В C++ можно использовать Си библиотеки и они будут исправно работать.

Компиляторы C++

Компилятор — программа, транслирующая исходный (высокоуровневый) код программы в конечный (низкоуровневый) код.

Компиляция — процесс преобразования высокоуровневого исходного текста программы, в эквивалентный текст программы, но уже на низкоуровневом языке.

Компоновщик (Линкер) — программа, которая генерирует исполнимый модуль путём связывания объектных файлов проекта.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

IDE (Интегрированная среда разработки)

- **IDE** — сочетание текстового редактора и компилятора.
- Разработка, компиляция и запуск своих программы осуществляется непосредственно в IDE.
- Интегрированные среды разработки упрощают процесс составления программ, так как написание кода компиляция и запуск программ выполняются в одной программе — IDE.
- Ещё одной важной особенностью IDE является то, что IDE помогает быстро найти и исправить ошибки компиляции.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

IDE (Интегрированная среда разработки)

- **CLion** компании JetBrains – рекомендуемая, новая, перспективная IDE (вообще-то она - платная, но для студентов НУК – бесплатно)
- **Code::Blocks с Mingw** - рекомендуемая, абсолютно бесплатная IDE
  - CLion и Code::Blocks также доступны на Linux и MacOS.
- **Microsoft Visual Studio** — это хорошая среда разработки приложений под ОС Windows.

Code::Blocks

Введение в язык C++

- Каждая программа в C++ имеет одну функцию, её называют главная или **main**-функция, выполнение программы начинается именно с этой функции.
- Из главной функции, вы также можете вызывать любые другие функции, неважно, являются ли они написанными нами, или, как упоминалось ранее, предоставляются компилятором.
- Чтобы получить доступ к стандартным функциям, которые поставляются с компилятором, необходимо подключить заголовочный файл используя препроцессорную директиву - **`#include`**.

Первая программа на C++

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

int main()
{
 cout <<"Моя первая программа на C++\n";
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Объявление переменных в C++

Чтобы объявить переменную используется синтаксис тип <имя>;

```
int num;
char character;
float num_float;
```

Допустимо объявление нескольких переменных одного и того же типа в одной строке

```
int x, y, z, d;
```

Распространенные ошибки при объявлении переменных в C++

- Если вы попытаетесь использовать переменную, которую не объявили, ваша программа не будет скомпилирована, и вы получите сообщение об ошибке.
- В C++, все ключевые слова языка, все функции и все переменные чувствительны к регистру.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Использование переменных

```
#include <iostream>

using namespace std;

int main()
{
 int number;

 cout <<"Введите число: ";
 cin >> number;
 cout <<"Вы ввели: " << number << "\n";
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Типы данных в C++

Тип	Размер	Диапазон значений
целочисленный (логический) тип данных		
bool	1	0 / 255
целочисленный (символьный) тип данных		
char	1	0 / 255
целочисленные типы данных		
short int	2	-32 768 / 32 767
unsigned short int	2	0 / 65 535
int	4	-2 147 483 648 / 2 147 483 647
unsigned int	4	0 / 4 294 967 295
long int	4	-2 147 483 648 / 2 147 483 647
unsigned long int	4	0 / 4 294 967 295
типы данных с плавающей точкой		
float	4	±3. 4e-038.. 3. 4e+038
long float	8	±1. 7e-308.. 1. 7e+308
double	8	±1. 7e-308.. 1. 7e+308

Изменение и сравнение величин

Независимо от того, какой тип данных вы используете, переменные не представляют особого интереса без возможности изменения их значения

Операция	Описание	Операция	Описание
+	Сложение	==	Равенство
-	Вычитание	>	Больше
*	Умножение	<	Меньше
/	Деление	!=	Не равно
%	Остаток	>=	Больше или равно
=	Присваивание	<=	Меньше или равно

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Изменение и сравнение величин

```
a = 4 * 6; // а равно 24
a = a + 5; // равно сумме исходного значения и пяти
a == 5     // не присваивается 5, выполняется проверка, а равно 5 или нет
```

```
a < 5  // Проверка, а менее пяти?
a > 5  // Проверка, а больше пяти?
a == 5 // Проверка, а равно пяти?
a != 5 // Проверка, а неравно пяти?
a >= 5 // Проверка, а больше или равно пяти?
a <= 5 // Проверка, а меньше или равно пяти?
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Арифметические операции C++

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

int main()
{
  double sum, razn, prod, div; // объявление переменных через запятую
  double a1; // отдельное объявление переменной a1
  double a2; // отдельное объявление переменной a2
  cout << "Vvedite pervoe chislo: ";
  cin >> a1;
  cout << "Vvedite vtoroe chislo: ";
  cin >> a2;
  sum  = a1 + a2; // операция сложения
  razn = a1 - a2; // операция вычитания
  prod  = a1 * a2; // операция умножения
  div  = a1 / a2; // операция деления
  cout << a1 << "+" << a2 << "=" << sum  << endl;
  cout << a1 << "-" << a2 << "=" << razn << endl;
  cout << a1 << "*" << a2 << "=" << prod << endl;
  cout << a1 << "/" << a2 << "=" << div  << endl;
  return 0;
}
```

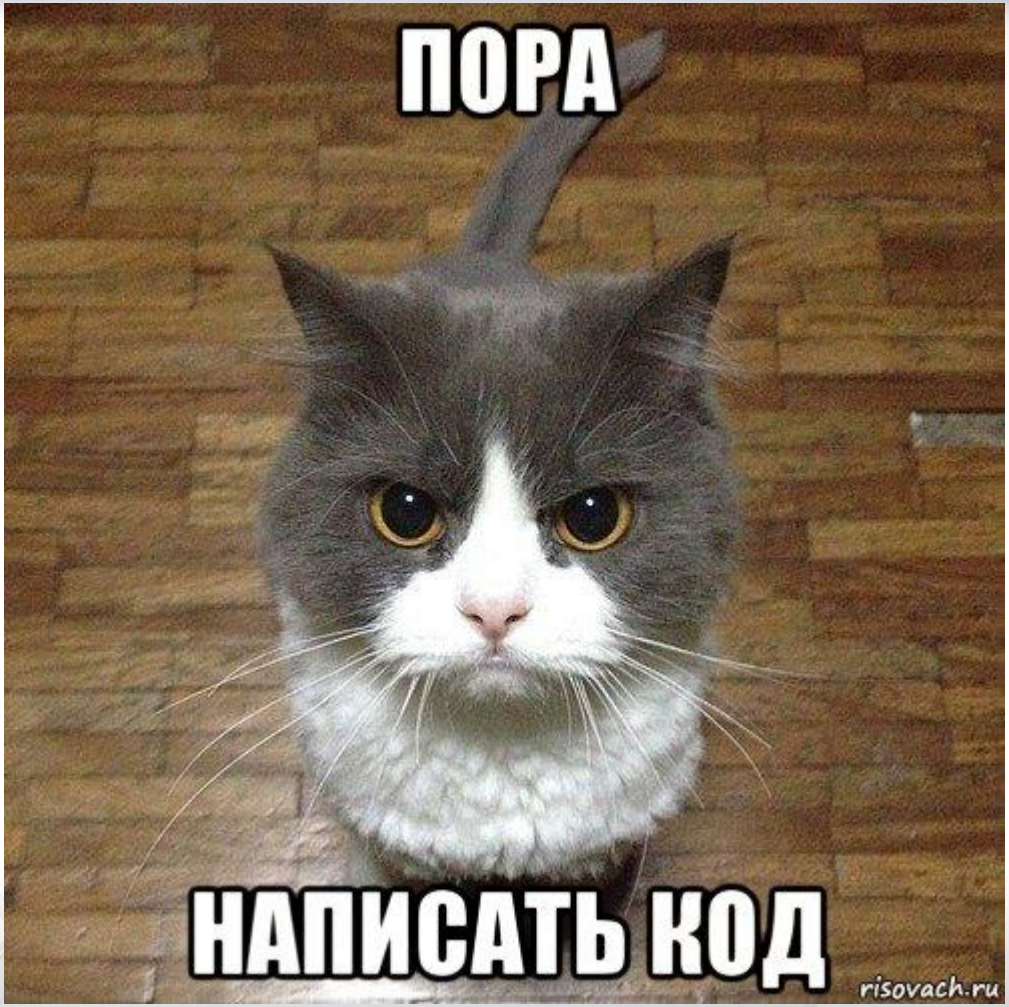
Математические функции

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

int main() {
  cout <<"log10(10)   = " <<log10(10.0)  << endl;
  cout <<"log10(1)    = " <<log10(1.0)   << endl;
  cout <<"log(2.718281) = " <<log(2.718281) << endl;
  cout <<"sqrt(9)     = " <<sqrt(9.0)    << endl;
  cout <<"pow(2,3)    = " <<pow(2.0,3.0) << endl;
  cout <<"abs(0)      = " <<abs(0.0)     << endl;
  cout <<"abs(-5)     = " <<abs(-5.0)    << endl;
  cout <<"ceil(3.14)  = " <<ceil(3.14)   << endl;
  cout <<"ceil(-2.4)  = " <<ceil(-2.4)   << endl;
  cout <<"floor(3.14) = " <<floor(3.14)  << endl;
  cout <<"floor(-2.4) = " <<floor(-2.4)  << endl;
  cout <<"fmod(2.4/2.0) = " <<fmod(2.4,2.0) << endl;
  return 0;
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Операции присваивания в C++

- Для сокращённой записи выражений в языке программирования C++ есть специальные операции, которые называются операциями присваивания.
- Рассмотрим фрагмент кода, с использованием операции присваивания.

```
int value = 256;
value = value + 256;
// обычное выражение с использованием двух операций: = и +
value += 256;
// сокращённое эквивалентное выражение
```

Операции присваивания в C++

В C++ существует пять операций присваивания, не считая основную операцию присваивания ”=“

- **+=** операция присваивания-сложения;
- **-=** операция присваивания-вычитания;
- ***=** операция присваивания-умножения;
- **/=** операция присваивания-деления;
- **%=** операция присваивания-остатка от деления

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Операции присваивания в C++

Операция	Обозначение	Пример	Экв.пример
операция присваивания-сложения	+=	var += 16	var = var + 16
операция присваивания-вычитания	-=	var -= 16	var = var — 16
операция присваивания-умножения	*=	var *= 16	var = var * 16
операция присваивания-деления	/=	var /= 16	var = var / 16
операция присваивания-остатка от деления	%=	var %= 16	var = var % 16

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.

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

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Структура if

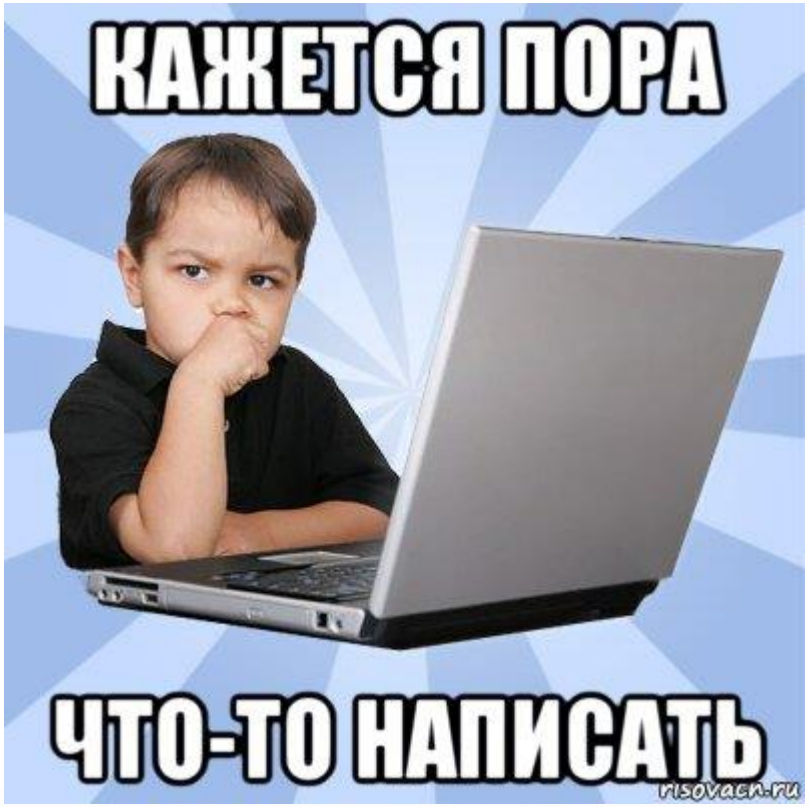
```
if ( условное выражение )
   // тут располагается один оператор, который выполнится,
   //если условное выражение - истинно
```

Пример использования

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

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

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»

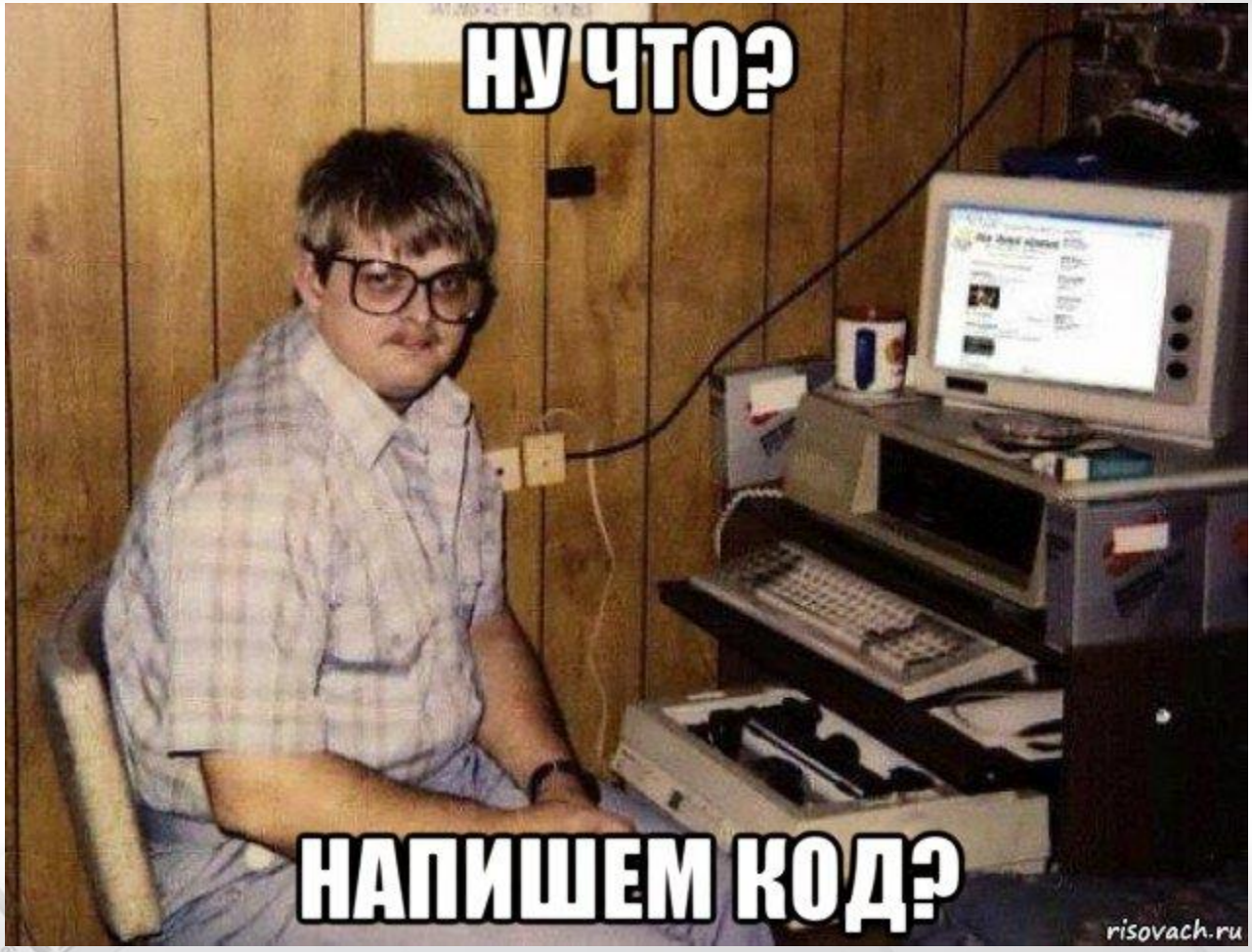
```
var > 0 and var < 10
```

- В языках C/C++ есть точно такой же оператор, и обозначается он знаками &&

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

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

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;
}
```

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

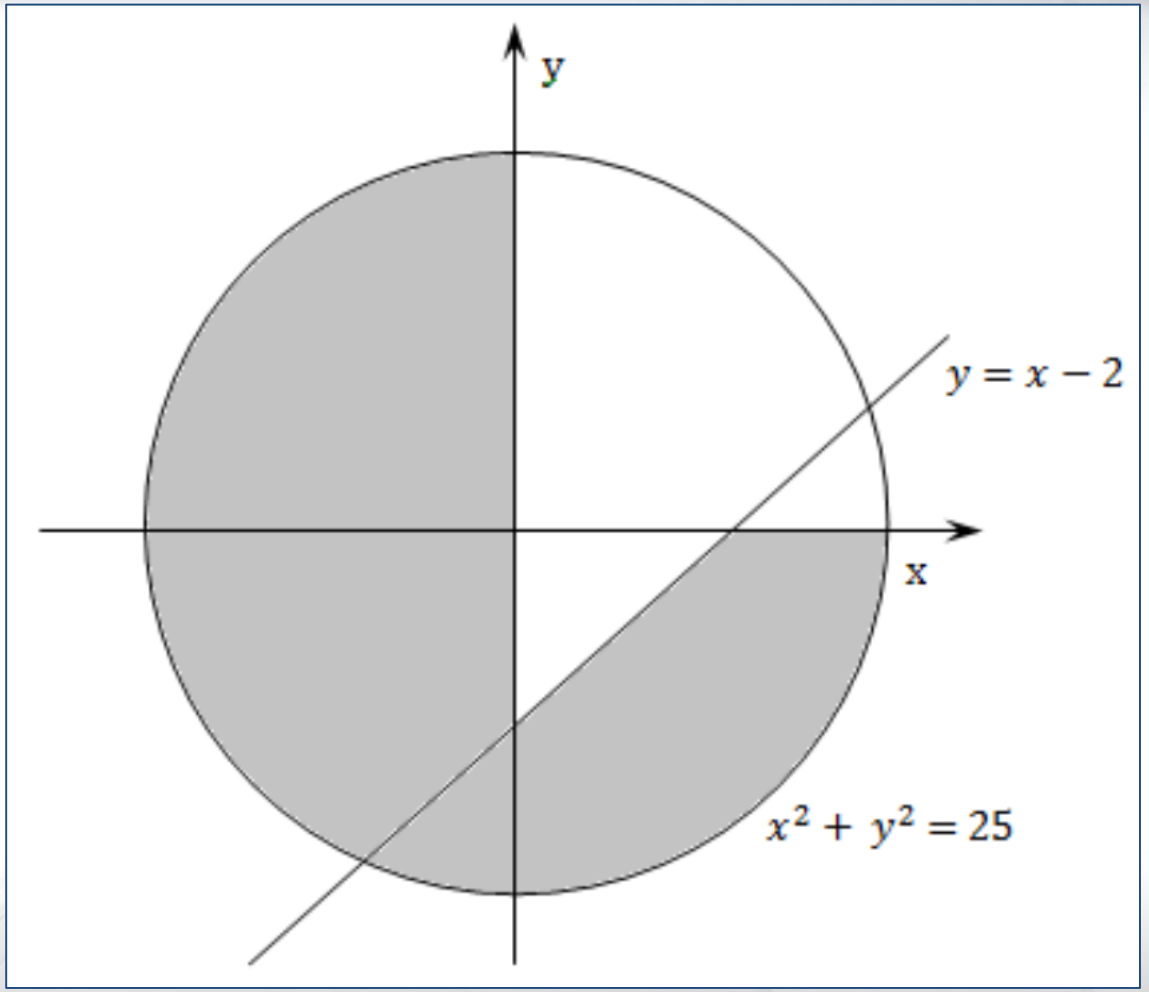
```
switch (op) {
      case 1: {
          cout << a << " + " << b << " = " << a + b << endl;
          break;
      }
      case 2: {
          cout << a << " - " << b << " = " << a - b << endl;
          break;
      }
      case 3: {
          cout << a << " * " << b << " = " << a * b << endl;
 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

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Демонстрация



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

e-olymp.com

Ввод-вывод, простая математика, ветвления:

https://www.e-olymp.com/ru/contests/10770

Но сначала нужно:

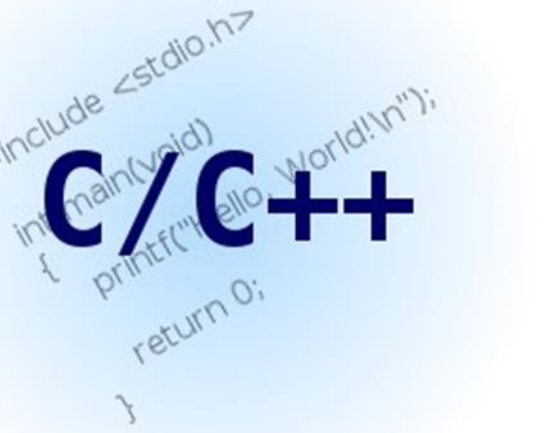
1) зарегистрироваться на e-olymp – выбрать логин и придумать пароль
2) прислать свой логин в сообщении пользователю “berkut” ☺
3) получив приглашение в группу, принять его

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++
(ч.1 – введение и условные операторы)



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