

# Занятие 1 — Введение в язык C++, среда программирования Code::Blocks, простейшие алгоритмы и программы

Программирование, численные методы и информатика

Александр Валерьевич Позднеев

Кафедра автоматизации научных исследований  
Факультет вычислительной математики и кибернетики  
Московский государственный университет имени М. В. Ломоносова  
<http://ani.cmc.msu.ru/geol>

Осенний семестр 2011/2012



# Содержание темы

- 1 Вводные замечания
  - Информация о курсе
  - Содержание курса
  - Литература
- 2 Программное обеспечение
- 3 Схема подготовки программы
- 4 Первый запуск среды Code::Blocks
- 5 Первая программа на C++
- 6 Блок-схемы алгоритмов
- 7 Типы алгоритмов
- 8 Линейный алгоритм
- 9 Типы переменных
- 10 Идентификаторы
- 11 Арифметические операторы
- 12 Комментарии
- 13 Задания для самостоятельной работы
- 14 Задания для домашней работы

# Информация о курсе

## Общая информация, цель курса, преподаватели

- ▶ Учебный курс «Программирование, численные методы и информатика» читает кафедра автоматизации научных исследований факультета вычислительной математики и кибернетики МГУ имени М. В. Ломоносова
- ▶ Слушатели — три учебных группы отделения инженерной геологии
- ▶ Цель курса — знакомство с базовыми понятиями и алгоритмами вычислительной математики, их реализацией и современными методами, подходами и технологиями, применяемыми при решении практически важных прикладных задач
- ▶ Преподаватели:
  - ▶ чл.-корр. РАН Дмитрий Павлович Костомаров (числ. методы)
  - ▶ доцент Владимир Вадимович Нефёдов (программирование)
  - ▶ ассистент Александр Валерьевич Позднеев (практические занятия)  
pozdnееv@cmc.msu.ru

# Информация о курсе

## Продолжительность, отчетность

- ▶ Продолжительность:
  - ▶ один семестр
  - ▶ одна лекция в неделю (два академических часа) — 16 лекций
  - ▶ одно практическое занятие в неделю (три академических часа)
- ▶ Итоговая отчетность:
  - ▶ экзамен в конце семестра
  - ▶ наличие отчетов о работе, проделанной при выполнении индивидуальных заданий
- ▶ Индивидуальные задания:
  - ▶ Численные методы вычисления определенных интегралов
  - ▶ Численные методы решения обыкновенных диф. уравнений
  - ▶ Методы решения нелинейных алгебраических уравнений
- ▶ Отчеты:
  - ▶ Каждое из трех выполненных заданий учебного практикума должно быть оформлено в виде специального отчета
  - ▶ Отчеты необходимо иметь с собой на экзамене
  - ▶ Без отчета вы не будете допущены до экзамена

# Содержание курса и практических занятий

- ▶ Численные методы
- ▶ Программирование
  - ▶ Язык программирования C++
  - ▶ Пакет символьных вычислений Maple



Подбельский В. В.

Язык Си++

М.: Финансы и статистика,  
2000, 2004, 2008

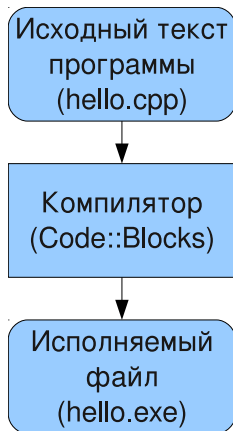


<http://www.finstat.ru/podbelsky-cpp.htm>

## Программное обеспечение

- ▶ Компилятор C++ и среда разработки — мы будем использовать свободную кроссплатформенную среду разработки **Code::Blocks**
- ▶ Пакет **Maple** — мы будем использовать версию 10 или старше
- ▶ Существуют также и другие среды разработки и компиляторы:
  - ▶ Среда Eclipse (Windows/Linux, бесплатная)
  - ▶ Среда Dev-C++ (Windows, бесплатная)
  - ▶ Среда Microsoft Visual C++ (Windows, коммерческая)
  - ▶ Компилятор GCC (Windows/Linux, бесплатный)
  - ▶ Компилятор Intel C (Windows/Linux, в Linux бесплатен для некоммерческого использования)
- ▶ Альтернативные бесплатные пакеты символьных вычислений:
  - ▶ Maxima
  - ▶ Axiom
- ▶ Рекомендуемые веб-браузеры:
  - ▶ Chromium / Google Chrome
  - ▶ Opera
  - ▶ Mozilla Firefox
  - ▶ Safari

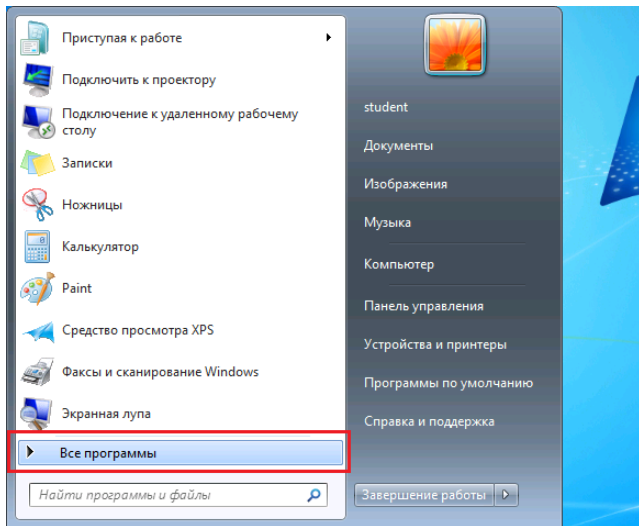
## Упрощенная схема подготовки программы





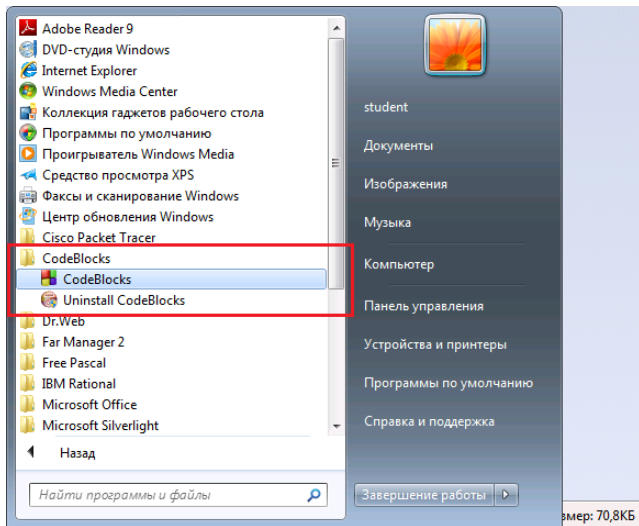
# Первый запуск среды Code::Blocks

Пункт меню «Пуск → Все программы»



# Первый запуск среды Code::Blocks

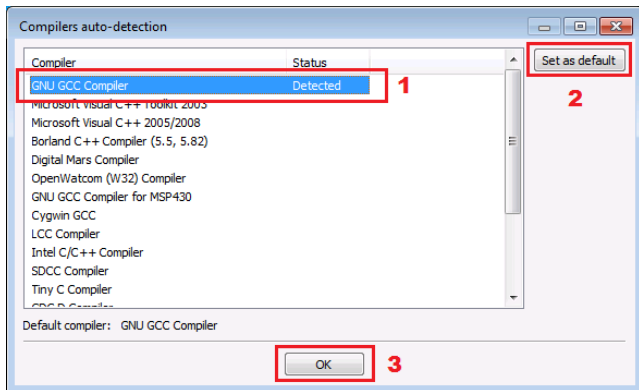
Пункт меню «Пуск → Все программы → CodeBlocks → CodeBlocks»



# Первый запуск среды Code::Blocks

Выбор компилятора «по умолчанию»

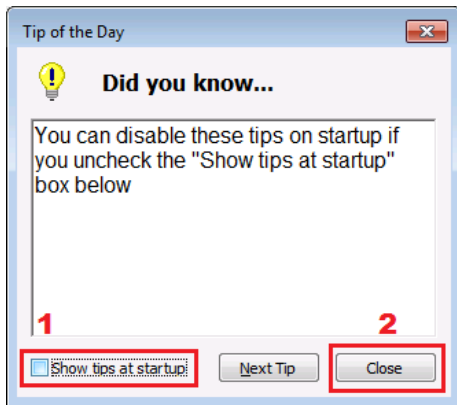
1. Выбрать компилятор GNU GCC Compiler
2. Нажать кнопку «Set as default»
3. Нажать кнопку «OK»



# Первый запуск среды Code::Blocks

Отключение советов, показываемых при запуске

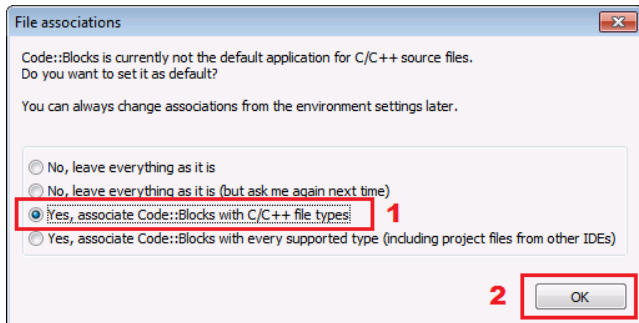
1. Снять галочку «Show tips at startup»
2. Нажать кнопку «Close»



# Первый запуск среды Code::Blocks

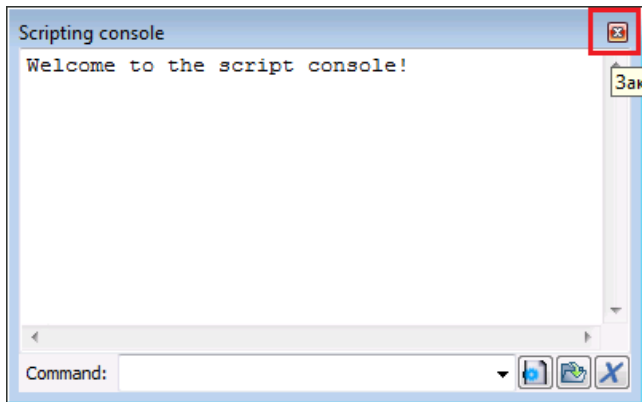
«Связать» в Windows файлы типа C/C++ со средой Code::Blocks

1. Выбрать пункт «Yes, associate Code::Blocks with C/C++ file types»
2. Нажать кнопку «OK»



# Первый запуск среды Code::Blocks

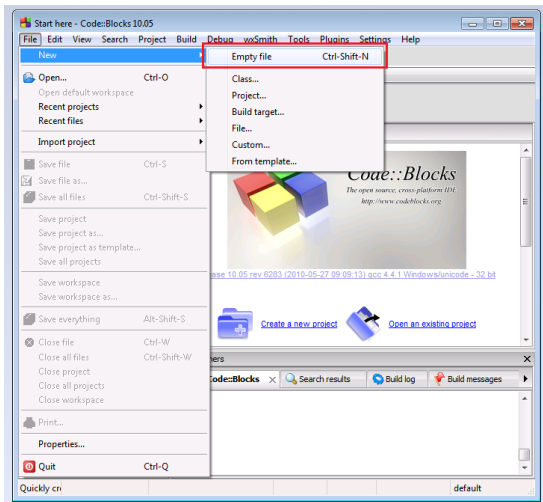
Закреть окно «Scripting console», нажав на крестик в правом верхнем углу



# Первая программа на C++

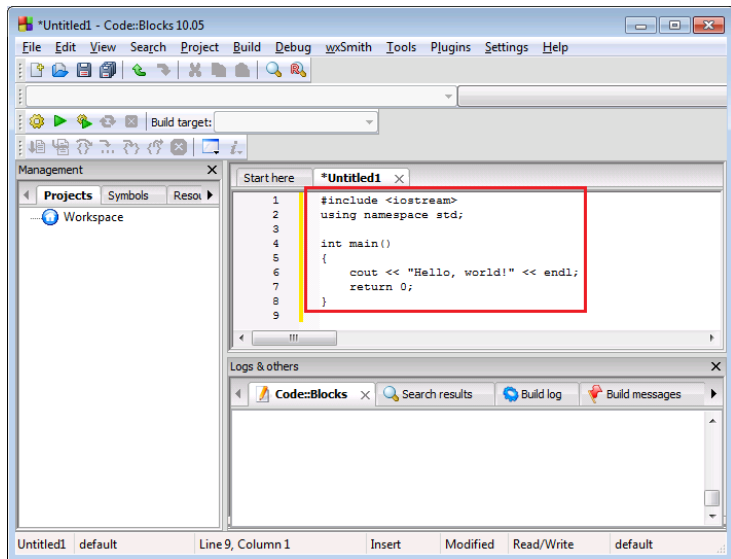
## Создание нового файла

Выбираем пункт меню «File → New → Empty file»



# Первая программа на C++

Вводим текст программы





# Первая программа на C++

Вводим текст программы

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

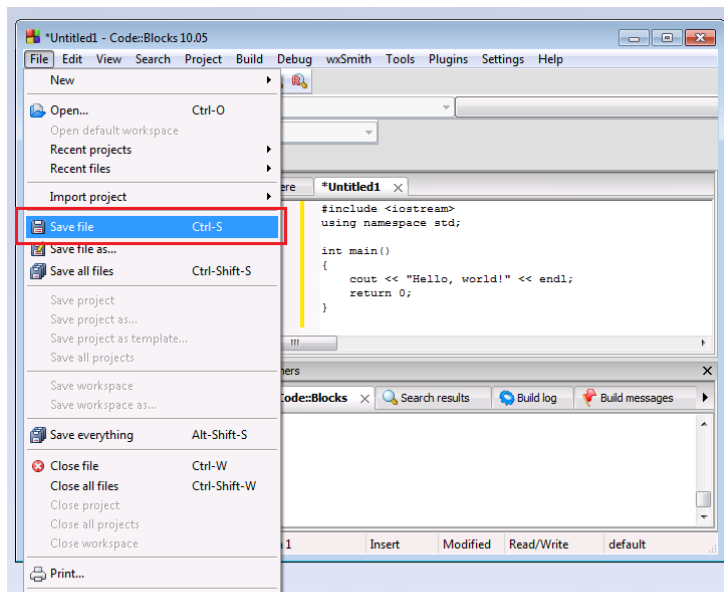
int main()
{
    cout << "Hello, World!" << endl;
    return 0;
}
```

## main vs. Main vs. MAIN

Язык C++ является чувствительным к регистру — заменять `main` на `Main` или `MAIN` нельзя

# Первая программа на C++

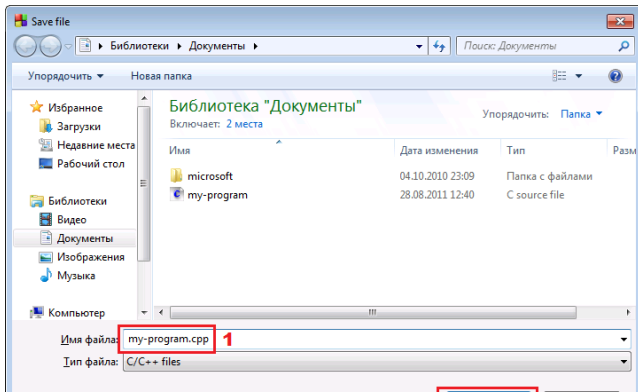
Сохраняем файл (пункт меню «File → Save File»)



# Первая программа на C++

Сохраняем файл как my-program.cpp

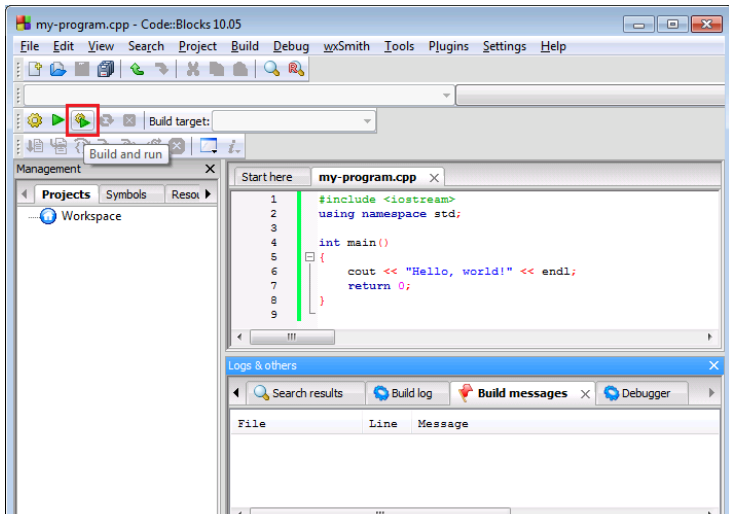
- ▶ Не рекомендуется использовать в именах файлов пробелы; вместо «hello world.cpp» программу лучше назвать «hello-world.cpp»
- ▶ При сохранении файла следует обязательно указывать расширение файла — .cpp



# Первая программа на C++

Компилируем и запускаем на выполнение

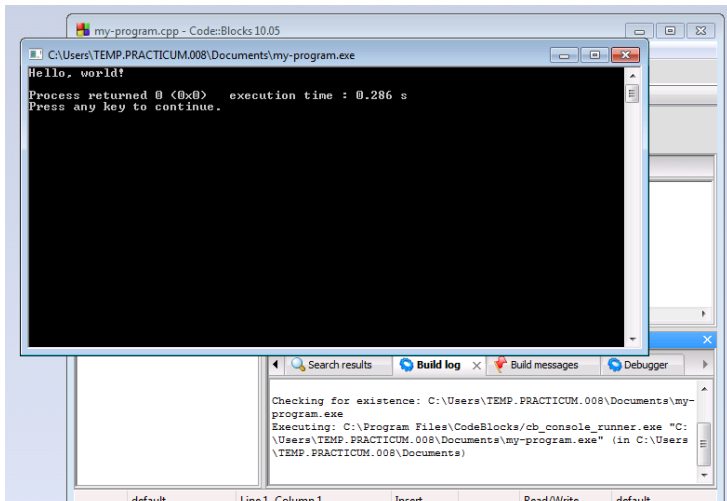
Кликаем на кнопке, на которой изображены желтая шестеренка и зеленый треугольник, либо используем «горячую клавишу» F9



# Первая программа на C++

## Результат запуска

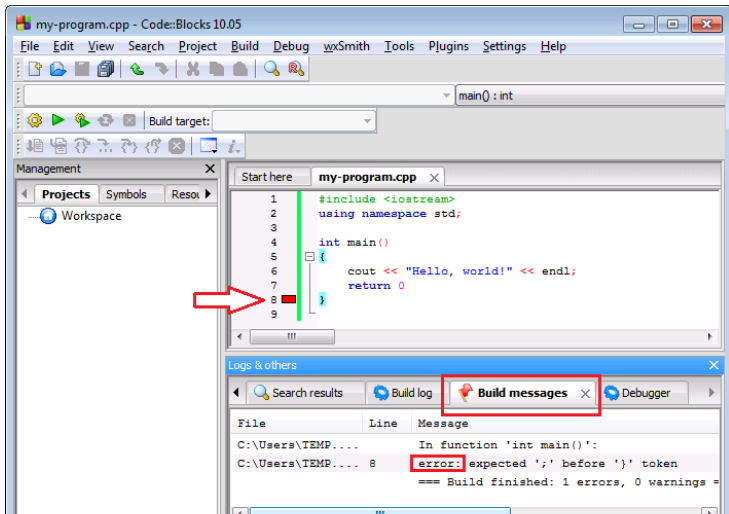
- ▶ Если все правильно, то вы увидите надпись «Hello, World!»
- ▶ Нажмите любую клавишу, чтобы завершить программу



# Первая программа на C++

## Ошибки при компиляции

Если при наборе программы вы совершили ошибку, то среда Code::Blocks сообщит об этом во время компиляции



# Первая программа на C++

## Разбор текста программы

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     return 0;
8 }
```

- ▶ 1: поддержка потоков (streams) ввода/вывода (Input/Output)
- ▶ 2: использовать имена из стандартной библиотеки C++
- ▶ 3: пустые строки игнорируются; они служат для улучшения читабельности
- ▶ 4: `main()` — любая программа на C++ должна включать функцию с этим именем; ключевое слово `int` означает, что программа возвратит операционной системе целое число

# Первая программа на C++

## Разбор текста программы (продолжение)

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     return 0;
8 }
```

- ▶ 5, 8: тело любой функции заключается в фигурные скобки
- ▶ 6: `cout` — это имя объекта, который обеспечивает вывод информации на экран дисплея
- ▶ 6: то, что нужно вывести, разделяется символами «<<»
- ▶ 6: вывод `endl` приводит к переводу курсора на новую строку
- ▶ 7: точка выхода из программы; операционной системе передается значение 0 (признак того, что программа завершилась успешно)



# Первая программа на C++

## Общие замечания

- ▶ Пустые строки компилятором игнорируются
- ▶ Переводы строк и пробелы, по большому счету, тоже не играют для компилятора значения
- ▶ Следующая программа полностью эквивалентна только что рассмотренной:

```
#include <iostream>
using namespace std;int main(){cout<<
"Hello, World!"<<endl;return 0;}
```

- ▶ Для повышения читабельности кода следует ставить в коде пустые строки и делать отступы

# Первая программа на C++

## Общие замечания (продолжение)

- ▶ Большинство конструкций в C++ завершаются символом «;» — он служит для того, чтобы отделять друг от друга различные операции
- ▶ Конструкция «`#include <iostream>`» буквально означает: «включить в текст программы файл `iostream`»
- ▶ Не используйте, в отличие от Подбельского, суффикс «.h» при подключении стандартных файлов («`#include <iostream.h>`») — это устаревший подход

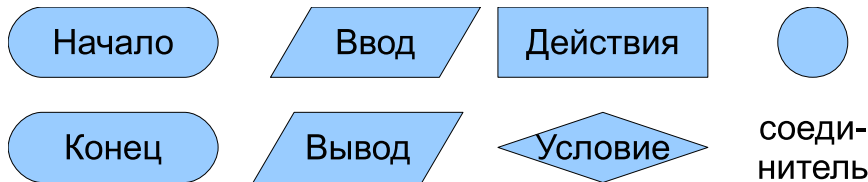
# Первая программа на C++

## Задания на самостоятельную работу

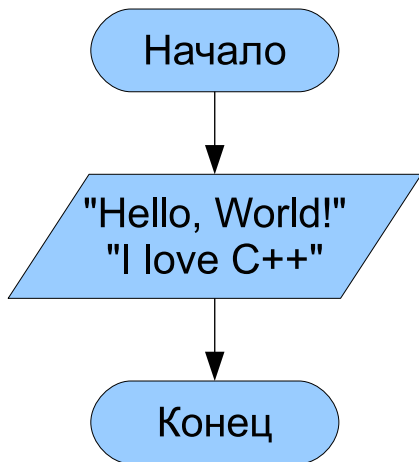
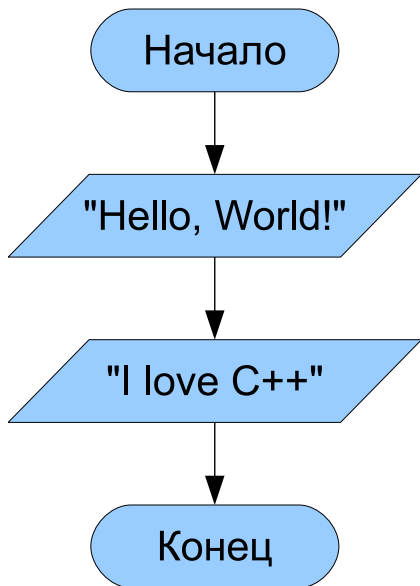
- ▶ Измените программу так, чтобы она напечатала *еще одну* строчку: «I love C++»
- ▶ Сколько раз пришлось воспользоваться объектом `cout`?
- ▶ Можно ли обойтись лишь одним обращением к объекту `cout`?

# Основные элементы блок-схем алгоритмов

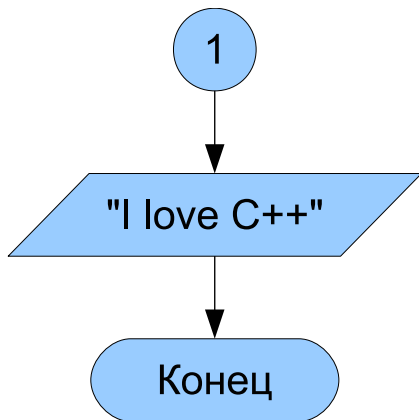
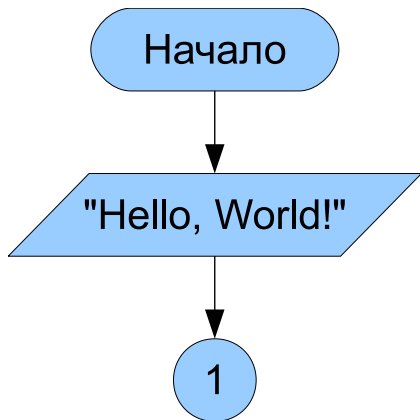
- ▶ *Алгоритм* — организованная последовательность действий, приводящая к результату
- ▶ Алгоритм оформляется с помощью блок-схем, языка программирования
- ▶ *Блок-схема* — описание шагов алгоритма в виде блоков различной формы, соединенных между собой стрелками
- ▶ Блок-схема — это модель алгоритма
- ▶ Основные элементы схем алгоритма:



## Блок-схема программы «Hello, World!»



## Блок-схема программы «Hello, World!» (продолжение)



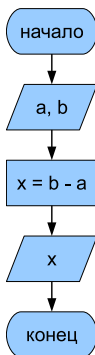
# Основные типы алгоритмов

- ▶ Различают три типа алгоритмов:
  - ▶ Линейный
  - ▶ Разветвляющийся
  - ▶ Циклический

# Линейный алгоритм

Решение уравнения  $a + x = b$

Необходимо вывести на экран решение уравнения  $a + x = b$ , значения  $a$  и  $b$  пользователь должен ввести с клавиатуры



```
#include <iostream>
using namespace std;
int main()
{
    double a, b;

    cout << "Input a: ";
    cin >> a;

    cout << "Input b: ";
    cin >> b;

    double x = b - a;
    cout << "Root of equation a+x=b is " << x << endl;

    return 0;
}
```



# Линейный алгоритм

Решение уравнения  $a + x = b$  (замечания)

- ▶ Ввод пользователя и промежуточные данные должны быть где-то сохранены
- ▶ Для этого служат переменные
- ▶ Все переменные в языке C++ должны быть объявлены перед использованием
- ▶ В языке C++ переменная может быть объявлена в любом месте программы, но
  - ▶ нельзя многократно объявлять переменные с одинаковыми именами в одном и том же блоке
  - ▶ переменную можно использовать сразу же после объявления и до завершения текущего блока (грубо говоря, до ближайшей закрывающей фигурной скобки)

# Линейный алгоритм

## Разбор текста программы

```
3 int main()
4 {
5     double a, b;
6
7     cout << "Input a: ";
8     cin >> a;
9
10
11
12
13     double x = b - a;
14     cout << "Root of equation a+x=b is " << x << endl;
```

- ▶ 4: объявление переменных  $a$  и  $b$ , представляющих из себя вещественные числа двойной точности
- ▶ 9: запись числа, введенного пользователем, в переменную  $a$
- ▶ 14: объявление переменной  $x$  с одновременным присвоением ей значения
- ▶ 15: вывод строки, вещественного числа и перевод строки

# Линейный алгоритм

## Задания для самостоятельной работы

- ▶ Вывести на экран сумму  $a + b$  чисел  $a$  и  $b$ ; значения  $a$  и  $b$  пользователь должен ввести с клавиатуры
- ▶ Необходимо ли здесь вводить дополнительную переменную, в которой будет сохраняться значение суммы?

# Типы переменных в языке C++

- ▶ Целые числа — `int`
  - ▶ Как правило, занимает в памяти 4 байта
  - ▶ Способно хранить числа от  $-2\,147\,483\,648$  до  $2\,147\,483\,647$
- ▶ Вещественные числа — `double`
  - ▶ Как правило, занимает в памяти 8 байт
  - ▶ Способно хранить числа до  $1,797693 \cdot 10^{308}$
  - ▶ Минимальное отличное от нуля число —  $2,225074 \cdot 10^{-308}$
- ▶ Символы — `char`
  - ▶ Как правило, занимает в памяти 1 байт
  - ▶ Может принимать значения символов латинского алфавита, кириллицы, знаков препинания и некоторые другие
- ▶ Существуют некоторые другие базовые типы данных, но очень маловероятно, что нам придется использовать их на практических занятиях

# Идентификаторы переменных и функций в языке

Имена переменных и функций включают

- ▶ символы латинского алфавита «a–z, A–Z»
- ▶ символ подчеркивания «\_»
- ▶ цифры «0–9»

Имена переменных и функций

- ▶ не должны начинаться с цифры
- ▶ не рекомендуется начинать и заканчивать символом «\_»
- ▶ не должны совпадать с ключевыми словами (int, return, ...)

Примеры:

- ▶ Допустимые идентификаторы: a, b, x1, y\_42, day\_of\_week, num\_of\_students, route\_66
- ▶ Недопустимые идентификаторы: 3rd\_law, double
- ▶ Нерекомендуемые идентификаторы: \_temp\_var, new\_var\_

# Арифметические операторы

- ▶ Арифметическая инструкция — это некоторое выражение, состоящее из констант, идентификаторов переменных и арифметических операторов, которая завершается точкой с запятой
- ▶ Оператор присваивания «=» присваивает одной переменной, идентификатор которой указывается слева от оператора «=» значение некоторого выражения, которое стоит справа:
  - ▶ + — оператор сложения
  - ▶ − — оператор вычитания
  - ▶ \* — оператор умножения
  - ▶ / — оператор деления ( $6/4 \rightarrow 1$ ,  $6.0/4 \rightarrow 1.25$ ,  $6/4.0 \rightarrow 1.25$ )
  - ▶ % — оператор взятия остатка от деления целых чисел ( $6 \% 4 \rightarrow 2$ )

# Комментарии

## Комментарии в стиле C++

- ▶ Комментарий — это последовательность символов, которая воспринимается компилятором как отдельный пробельный символ или, иными словами, игнорируется
- ▶ Комментарии, которые начинаются символами «//», включают в себя весь последующий текст после этих символов и продолжаются до конца строки

```
int main() { // the beginning of the main function  
    double pi, e; // the pi and the e numbers  
    return 0; // exit point of the function  
// you may comment empty lines as well  
}
```

# Комментарии

## Классические комментарии в стиле языка C

- ▶ Комментарий может иметь и следующую форму представления:

```
/*<characters>*/
```

```
/* Some long comment,  
that could explain,  
what this function is doing  
*/
```

```
int main() {  
    /* such comments are useful */  
    /* to disable some parts of the code */  
    /*  
    double a, b, c;  
    cout << a << ' ' << b << ' ',  
        << c << endl;  
    */  
    return 0;  
}
```



## Задания для самостоятельной работы

Напишите программу, содержащую объявление следующих переменных. Выведите на экран значения всех объявленных переменных. Объясните, почему получился именно такой результат:

```
int a = 13/5;
int b = 13%5;
int c = 13.0/5;
double d = 13/5;
double e = 13%5;
double f = 13.0/5;
double g = 13/5 + 2/5;
double h = 13.0/5 + 2.0/5;
int i = 13.0/5 + 2.0/5;
```

## Задания для самостоятельной работы

1. Даны две целые переменные  $a$  и  $b$ . Составить программу, после работы которой значения переменных поменялись бы местами (т.е. новое значение переменной  $a$  равно старому значению переменной  $b$  и наоборот)
2. Дано действительное число  $x$ . Вычислите число  $x^4$ . Какое наименьшее число операций умножения необходимо для этого?
3. Дано число  $x$ . Вычислите число  $x^7$  при помощи четырех операций умножения.
4. Дано число  $x$ . Вычислите число  $x^{13}$  при помощи пяти операций умножения.
5. Дано натуральное число. Выведите его последнюю цифру.
6. Дано натуральное число. Найдите число десятков в его десятичной записи (то есть вторую справа цифру его десятичной записи).

## Задания для домашней работы

1. Дано число  $x$ . Вычислите число  $x^6$  при помощи трех операций умножения.
2. Дано число  $x$ . Вычислите число  $x^8$  при помощи трех операций умножения.
3. Дано число  $x$ . Вычислите число  $x^{21}$  при помощи шести операций умножения.
4. Дано двузначное число. Найдите число десятков в нем.
5. Дано трехзначное число. Найдите сумму его цифр.

Задания повышенной сложности:

1. Даны две целые переменные  $a$  и  $b$ . Составить программу, после работы которой значения переменных поменялись бы местами, но не используя каких-либо других дополнительных переменных
2. Даны три целых числа:  $h$ ,  $m$ ,  $s$ . Определите угол (в градусах) между часовой стрелкой на циферблате часов в момент времени « $h$  часов,  $m$  минут,  $s$  секунд» и между часовой стрелкой в полночь.