

Введение в язык C++, среда программирования Dev-C++, простейшие алгоритмы и программы

Программирование, численные методы и информатика

А. В. Позднеев

Кафедра автоматизации научных исследований
Факультет вычислительной математики и кибернетики
Московский государственный университет имени М. В. Ломоносова
<http://ani.cmc.msu.ru/geo1>

Осенний семестр 2010/2011



Содержание темы

- 1 Информация о курсе
- 2 Содержание курса
- 3 Литература
- 4 Программное обеспечение
- 5 Схема подготовки программы
- 6 Первая программа на C++
- 7 Блок-схемы алгоритмов
- 8 Типы алгоритмов
- 9 Линейный алгоритм
- 10 Типы переменных
- 11 Идентификаторы
- 12 Арифметические операторы
- 13 Комментарии
- 14 Задания для самостоятельной работы
- 15 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)
 - ▶ доц. Владимир Вадимович Нефёдов (программирование)
 - ▶ 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:
 - ▶ итоговый 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

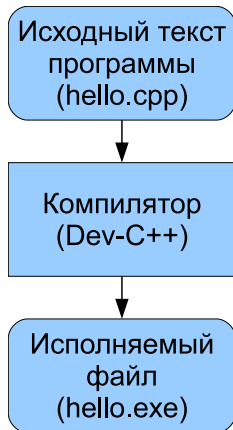


<http://www.finstat.ru/podbelsky-cpp.htm>

Программное обеспечение

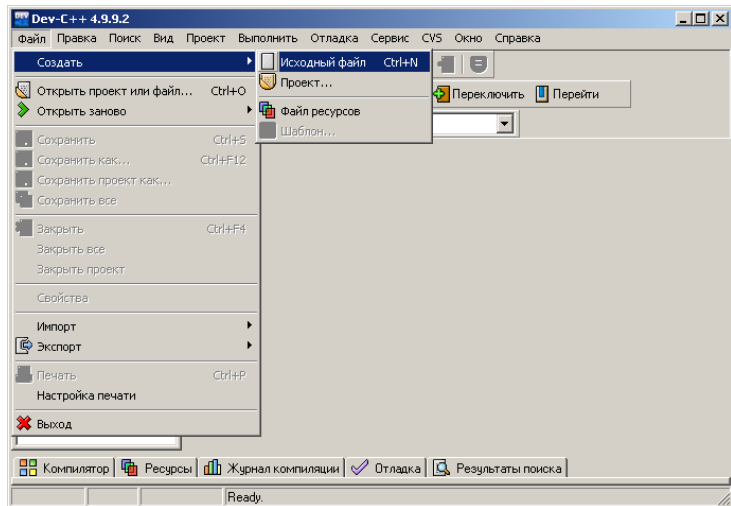
- ▶ Компилятор C++ и среда разработки — мы будем использовать среду программирования **Dev-C++** для операционных систем семейства Windows
- ▶ Пакет **Maple** — мы будем использовать версию 10 или старше
- ▶ Но существуют и другие среды разработки и компиляторы:
 - ▶ Microsoft Visual C++ (Windows, коммерческая)
 - ▶ Среда Eclipse (Windows/Linux, бесплатная)
 - ▶ 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 для ОС семейства Windows:
 - ▶ Mozilla Firefox — если вы не ограничены в вычислительных мощностях и объеме трафика
 - ▶ Opera — если вы работаете на старом «железе» или приходится экономить трафик

Упрощенная схема подготовки программы



Первая программа на C++

Запускаем Dev-C++



Первая программа на C++

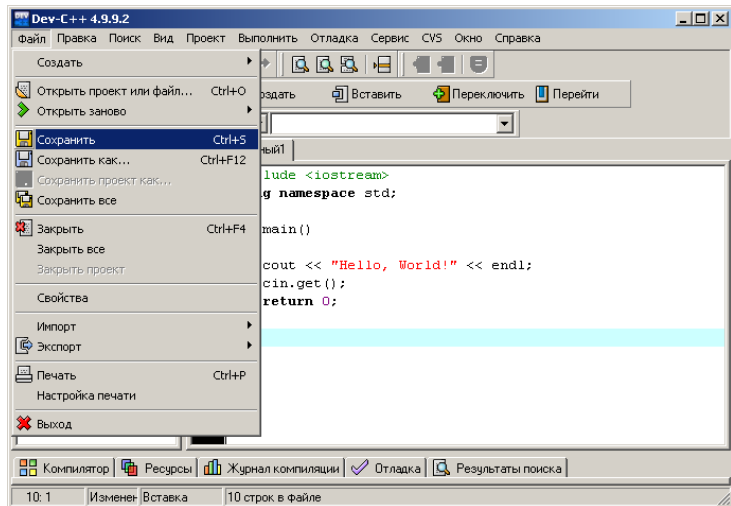
Вводим текст программы

- ▶ Номера строк вводить не нужно — они приведены для удобства
- ▶ Язык C++ является чувствительным к регистру — заменять main на Main или MAIN нельзя

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     system("pause");
8     return 0;
9 }
```

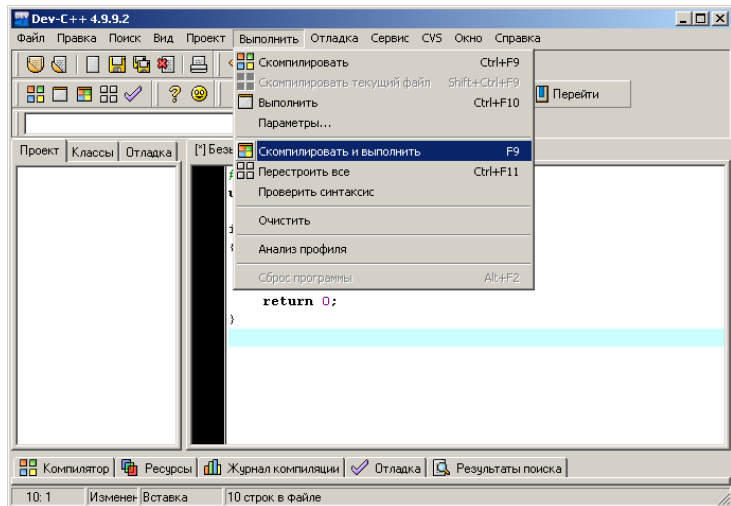
Первая программа на C++

Сохраняем файл как hello.cpp



Первая программа на C++

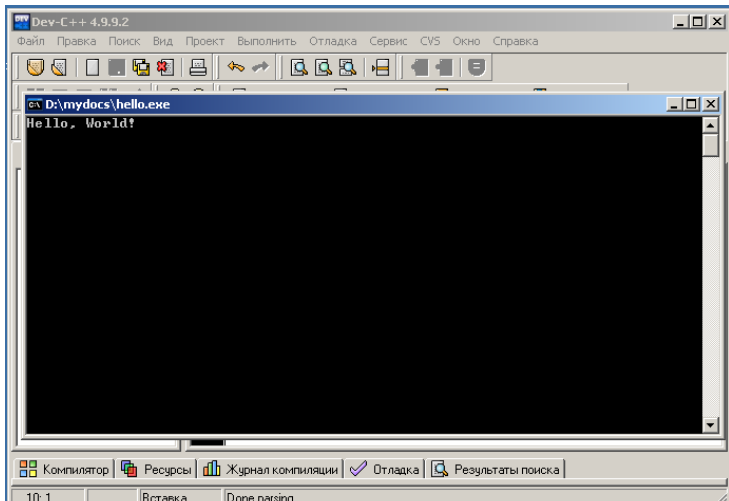
Компилируем и запускаем на выполнение



Первая программа на C++

Результат запуска

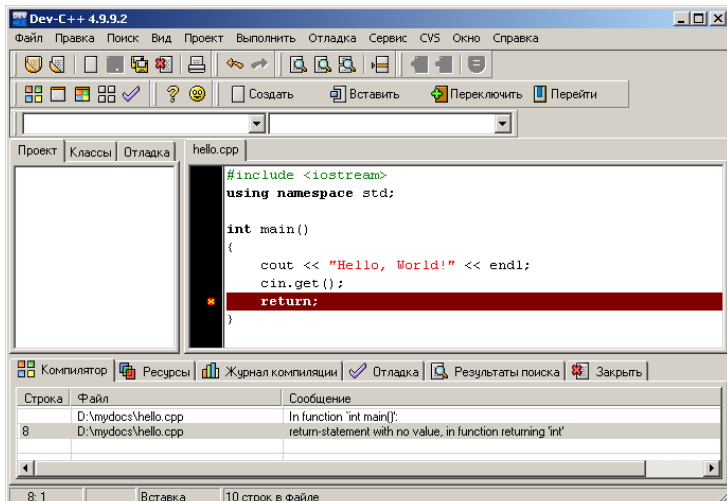
- ▶ Если все правильно, то вы увидите надпись «Hello, World!»
- ▶ Нажмите кнопку «Enter», чтобы завершить программу



Первая программа на C++

Ошибки при компиляции

- ▶ Если при наборе программы вы совершили ошибку, то Dev-C++ 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     system("pause"); // wait for user to press any key
8     return 0;
9 }
```

- ▶ 1: поддержка потоков (streams) ввода/вывода (Input/Output)
- ▶ 2: использовать имена из стандартной библиотеки C++
- ▶ 3: пустые строки и комментарии // игнорируются; они служат для повышения читабельности и документирования
- ▶ 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     system("pause"); // wait for user to press any key
8     return 0;
9 }
```

- ▶ 5, 9: 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: печать endl приводит к переводу курсора на новую строку

Первая программа на C++

Разбор текста программы (окончание)

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!" << endl;
7     system("pause"); // wait for user to press any key
8     return 0;
9 }
```

- ▶ 7: конструкция `system("pause")` ожидает от пользователя нажатия любой кнопки и не дает окну программы закрыться
- ▶ 8: точка выхода из программы; операционной системе передается значение «0»

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:

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

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

Первая программа на C++

Общие замечания (продолжение)

- ▶ Комментарии, начинающиеся с символов «//», длятся до конца строки
 - ▶ Обязательно оставляйте комментарии к нетривиальным операциям, которые вы выполняете в программе!
- ▶ 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

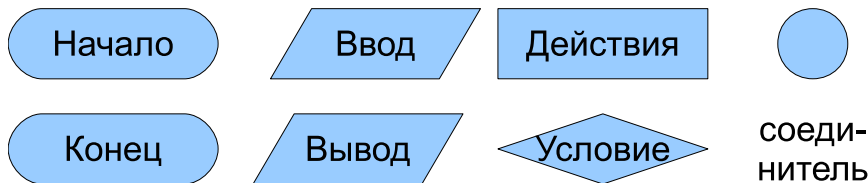
Первая программа на C++

Задания на самостоятельную работу

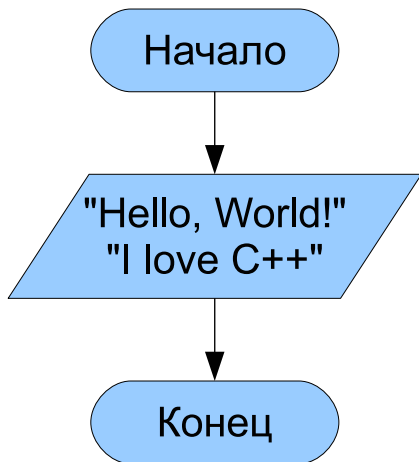
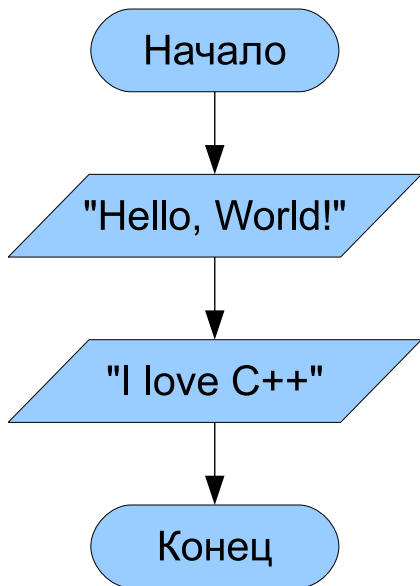
- ▶ Измените программу так, чтобы она напечатала еще одну строчку: «I love C++»
- ▶ Сколько раз пришлось воспользоваться объектом `cout`?
- ▶ Можно ли обойтись лишь одним обращением к объекту `cout`?

Основные элементы блок-схем алгоритмов

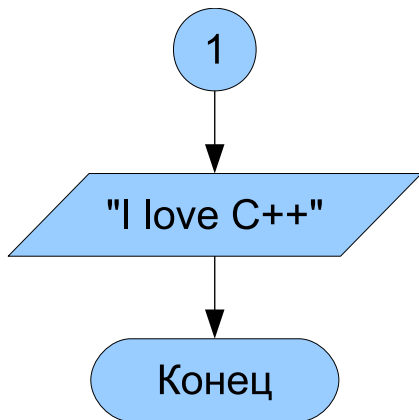
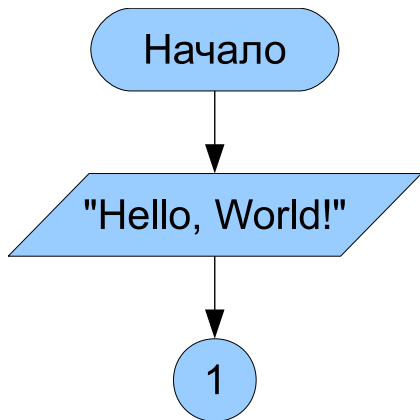
- ▶ *Алгоритм* — организованная последовательность действий, приводящая к результату
- ▶ Алгоритм оформляется с помощью блок-схем, языка программирования
- ▶ *Блок-схема* — описание шагов алгоритма в виде блоков различной формы, соединенных между собой стрелками
- ▶ Основные элементы схем алгоритма:



Блок-схема программы «Hello, World!»



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



Основные типы алгоритмов

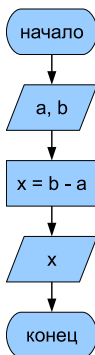
- ▶ Различают три типа алгоритмов:
 - ▶ Линейный
 - ▶ Разветвляющийся
 - ▶ Циклический

Линейный алгоритм

Решение уравнения $a + x = b$

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

Решение:



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     double a, b;
7
8     cout << "Input a: ";
9     cin >> a;
10
11    cout << "Input b: ";
12    cin >> b;
13
14    double x = b - a;
15    cout << "Root of equation a+x=b is " << x << endl;
16
17    system("pause");
18    return 0;
19 }
```

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

```
1 int main()  
2 {  
3     double a, b;  
4  
5     cout << "Input a: ";  
6     cin >> a;
```

```
1     double x = b - a;  
2     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

```
1 int main() { // the beginning of the main function
2     double pi, e; // the pi and the e numbers
3     return 0; // exit point of the function
4 // you may comment empty lines as well
5 }
```


Комментарии

Классические комментарии в стиле языка C

- ▶ Комментарий может иметь и следующую форму представления:
`/*<characters>*/`

```
1 /* Some long comment,  
2 that could explain,  
3 what this function is doing  
4 */  
5 int main() {  
6     /* such comments are useful */  
7     /* to disable some parts of the code */  
8     /*  
9     double a, b, c;  
10    cout << a << ' ' << b << ' ',  
11        << c << endl;  
12    */  
13    return 0;  
14 }
```

Задания для самостоятельной работы

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

```
1 int a = 13/5;  
2 int b = 13%5;  
3 int c = 13.0/5;  
4 double d = 13/5;  
5 double e = 13%5;  
6 double f = 13.0/5;  
7 double g = 13/5 + 2/5;  
8 double h = 13.0/5 + 2.0/5;  
9 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.