

1.1. Пример программы на языке C++

Рассмотрим в качестве примера программу, которая по двум катетам прямоугольного треугольника определяет его гипотенузу.

Листинг 1.1.

```
1 #include <iostream>           //библиотека для потока ввода-вывода
2 #include <cmath>              //библиотека математических функций
3 using namespace std;
4
5 int main(){
6     setlocale(LC_ALL, "RUS");  //использование русской клавиатуры
7     float x, y;
8     cout << "Введите два катета\n";
9     cin >> x >> y;            //ввод двух катетов
10    float c = sqrt(x * x + y * y); //теорема Пифагора
11    cout << "Гипотенуза равна " << c << endl; //вывод
12    system("pause");           //задержка консоли
13    return 0;
14 }
```

Команда `#include<...>` (1–2 строки Листинга 1.1) называется *Директива препроцессора* и позволяет подключать к программе возможности, описанные в *заголовочных файлах*

Заголовочный файл `iostream.h` содержит данные и функции для работы с потоком ввода-вывода.

Заголовочный файл `math.h` содержит математические функции и математические константы.

Директива `using namespace std` позволяет определить, что подключенные библиотеки относятся к стандартному пространству имен.

Желательно подключать эти библиотеки во все свои проекты.

Программа на C++ начинается с выполнения функции `main()`. Наличие этой функции является обязательным.

Синтаксис функции `main()`:

```
int main(){
    //операторы
    return 0;
}
```

Функция `setlocale(LC_ALL, "Rus");` позволяет использовать для вывода на экран буквы русского алфавита.

Функция `system("pause");` ожидает нажатия какой-либо клавиши для закрытия консольного окна.

1.2. Создание, отладка и запуск проекта

Рассмотрим пример создания проекта в интегрированной среде Visual Studio 2008.

Сначала необходимо создать новый проект (рисунок 1.1):

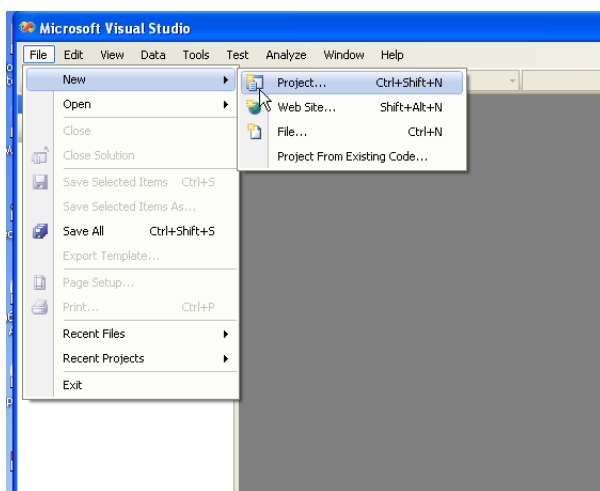


Рис. 1.1. Создание нового проекта

В открывшемся диалоговом окне выбрать `Empty project` и в поле Name ввести имя проекта (рисунок 1.2).

Теперь необходимо добавить в проект файлы. Если нужно создать новый файл, выбираем `Add New Item`, если необходимо подключить существующий — `Add Existing Item` (рисунок 1.3).

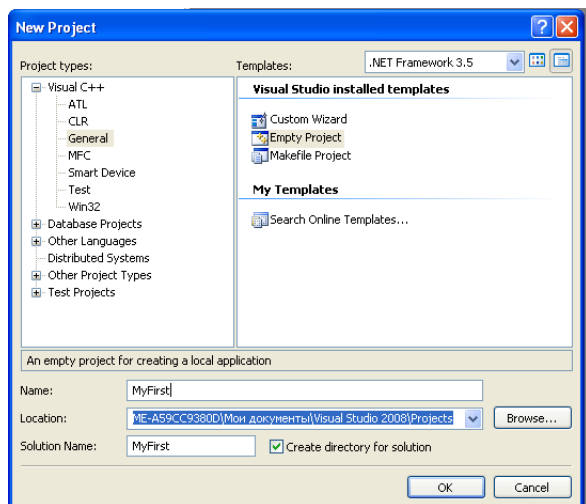


Рис. 1.2. Пустой проект

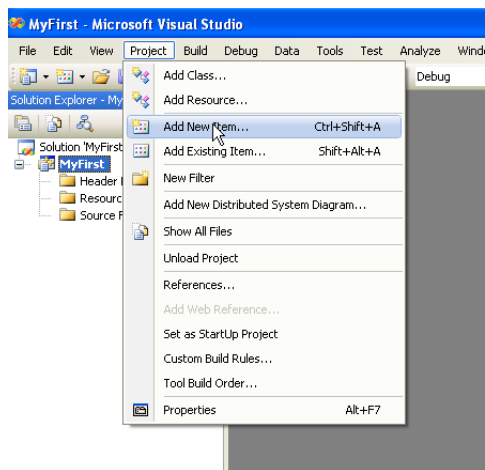


Рис. 1.3. Подключение файла

В открывшемся диалоговом окне выбрать C++ File.cpp и в поле Name ввести имя файла (рисунок 1.4). Имя файла может не совпадать с именем проекта.

После написания программы необходимо проверить ее на наличие ошибок, произвести сборку и только после этого можно запускать программу. В сложных проектах исправление ошибок не всегда изменяет собранное решение, поэтому лучше всего при изменениях в программе пересобирать решение (рисунок 1.5).

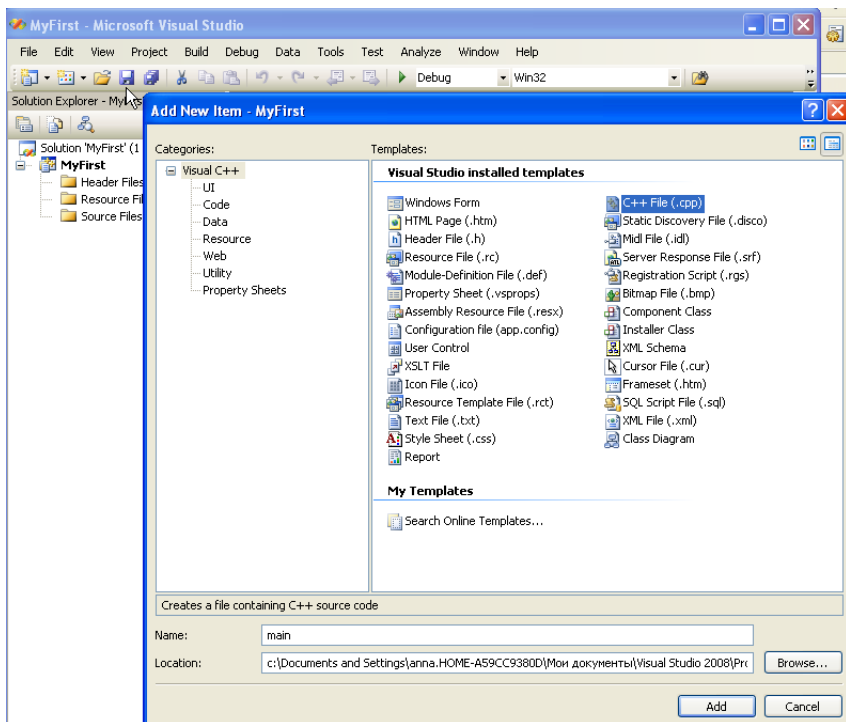


Рис. 1.4. Файл .cpp

В случае корректной записи программы, необходимо нажать **Ctrl + F5**. В открывшемся консольном окне выполнять действия (рисунок 1.6).

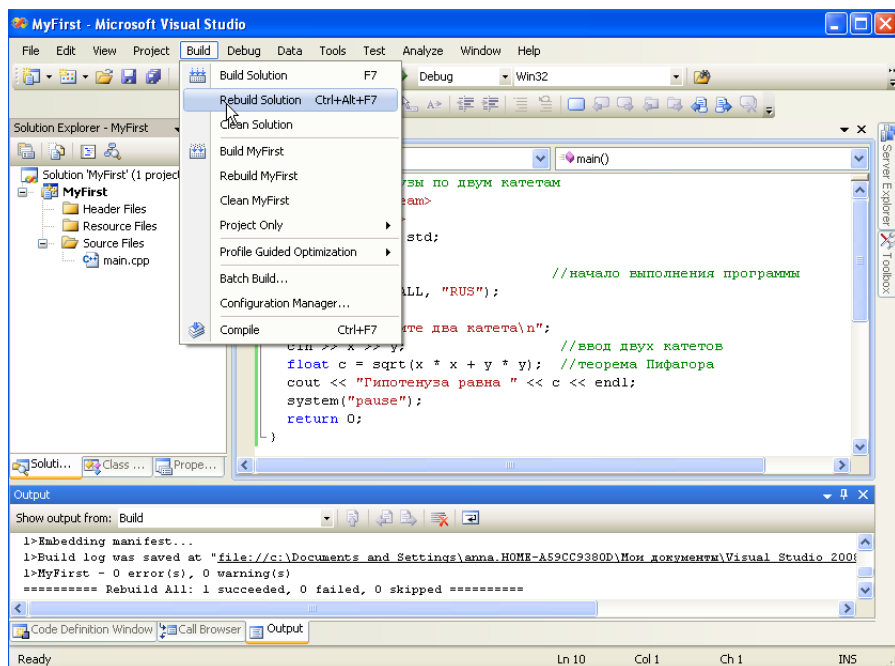


Рис. 1.5. Сборка решения

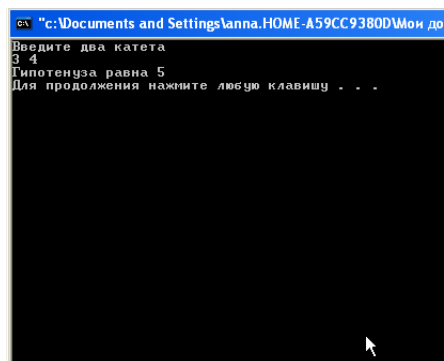


Рис. 1.6. Выполнение программы (Листинг 1.1)

Правила оформления программы

1. Добавляйте комментарии. Желательно, чтобы сначала было описано краткое условие задачи.
2. Выбирайте имена переменных и функций так, чтобы они имели смысл. (Например, `Sqr` — площадь фигуры.)
3. Фигурные скобки пишите следующим образом: Открывающая скобка на той же строке, что и конструкция, для которой она применяется, закрывающая скобка — на новой строке. Либо и открывающая, и закрывающая фигурные скобки должны находиться на отдельных строках.
4. Между операндами и знаками операции обязательно должны быть пробелы. Это делает код более читаемым. Например, запись вида `x=2*a+6.3+b-3*c`; менее наглядная, чем `x = 2 * a + 6.3 + b - 3 * c`;
5. Операторы, находящиеся внутри блока (между открывающей и закрывающей фигурными скобками) должны быть записаны с некоторым отступом. Если сразу не отформатировали, выделяйте нужный код, нажимаете сначала `Ctrl + K`, потом `Ctrl + F`, тогда код будет отформатирован автоматически.
6. Если используете открывающие элементы (`{`, `(`, `"`), сразу же пишите закрывающие элементы (`}`, `)`, `"`), а потом между ними код.

Примеры решения задач

Пример 3.1. Даны два числа x, y . Вычислить значение выражения $F(x, y) = \frac{2}{3} \sqrt{\frac{\cos x}{y + 1}}$.

Решение.

Поскольку дано математическое выражение, переменные и результат будут вещественного типа.

2 и 3 — неименованные константы типа `int`, поэтому `2 / 3` будет равно нулю, так как результат деления двух целых чисел — целое число. Следовательно, необходимо преобразовать одну из этих констант в вещественный тип: `2.0 / 3`.

Необходимо также расставить скобки для изменения приоритета вычислений.

В итоге, на языке C++ выражение имеет вид:

```
float c = 2.0 * sqrt(cos(x) / (y + 1)) / 3;
```

Листинг 3.1.

```
1 //вычислить значение функции
2 #include <iostream>
3 #include <cmath>
4 using namespace std;
5
6 int main(){
7     setlocale (LC_ALL, "RUS");
8     float x, y;
9     cout << "x = ";
10    cin >> x;
11    cout << "y = ";
12    cin >> y;
13    float c = 2.0 * sqrt(cos(x) / (y + 1)) / 3;
14    cout << "Результат: " << c << endl;
15    system("pause");
16    return 0;
17 }
```

□

Пример 3.2. Дано трехзначное число X . Определить кратна ли сумма его цифр числу Y .

Решение.

Поскольку необходимо определить цифры числа, определить кратность числа, следовательно, используемые переменные должны быть типа `int`.

Трехзначное число можно представить в виде: $X = 100 \times a + 10 \times b + c$.

Соответственно, остаток от деления на 10 числа позволяет найти последнюю цифру числа, а частное от деления на 10 отбрасывает младший разряд числа. Этот процесс повторяется до тех пор, пока все цифры не будут определены.

То есть необходимо повторять следующие действия:

```
int X, a;  
a = X % 10;  
X /= 10;
```

Лучше всего использовать циклы, но пока будем последовательно отделять цифры от младшего разряда к старшему.

Для того, чтобы определить, кратно ли число A числу B , необходимо найти остаток от деления A на B .

Для поиска ответа на вопрос задания, воспользуемся тернарной функцией:

```
(S % Y == 0)? cout << "Yes\n": cout << "No";
```

Пока будем предполагать, что вводимые данные корректны (X — трехзначное число, Y — однозначное). В дальнейшем требуется проводить такие проверки.

Листинг 3.2.

```
1 //кратна ли сумма цифр числа X числу Y  
2 #include <iostream>  
3 #include <cmath>  
4 using namespace std;  
5  
6 int main(){  
7     setlocale (LC_ALL, "RUS");  
8     int x, y;  
9     cout << "Введите трехзначное число\n";  
10    cin >> x;  
11  
12    //разбиваем на цифры  
13    int a, b, c;  
14    a = x % 10; //3 цифра
```



```
15  x /= 10;
16  b = x % 10; //2 цифра
17  c = x / 10; //1 цифра
18
19  //сумма
20  int sum = a + b + c;
21  cout << "Введите однозначное число\n";
22  cin >> y;
23
24  (sum % y == 0) ? cout << "YES\n" : cout << "NO\n";
25  system("pause");
26  return 0;
27 }
```

□