

Программирование (C++)

9 класс

Что такое программирование?

Программирование — это создание программ для компьютеров. Этим занимаются **программисты**.

Чем занимаются **программисты**:

анализ задачи (выделение исходных данных, связей между ними, этапов решения задачи)

системные аналитики

разработка **алгоритмов**

алгоритмисты

написание и отладка **программ**

кодировщики

тестирование программ

тестировщики

написание **документации**

технические писатели

Направления в программировании

системный программист

операционные системы,
утилиты, драйверы

прикладной программист

прикладные программы, в
т.ч. для мобильных
устройств

веб-программист

веб-сайты

программист баз данных

системы управления
базами данных

Языки программирования

- К машинным языкам относится ассемблер
- К языкам высокого уровня: Паскаль, Бейсик, Си.
- К языкам визуального программирования : Делфи, Визуал Бейсик, С++, Питон.

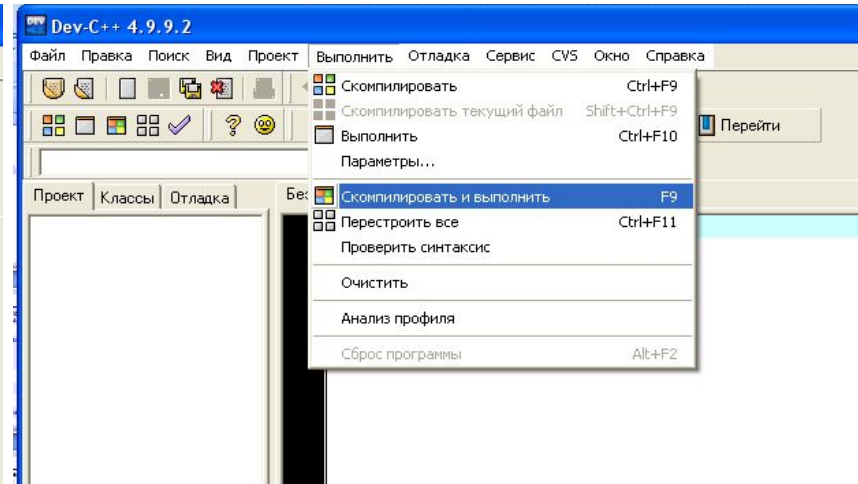
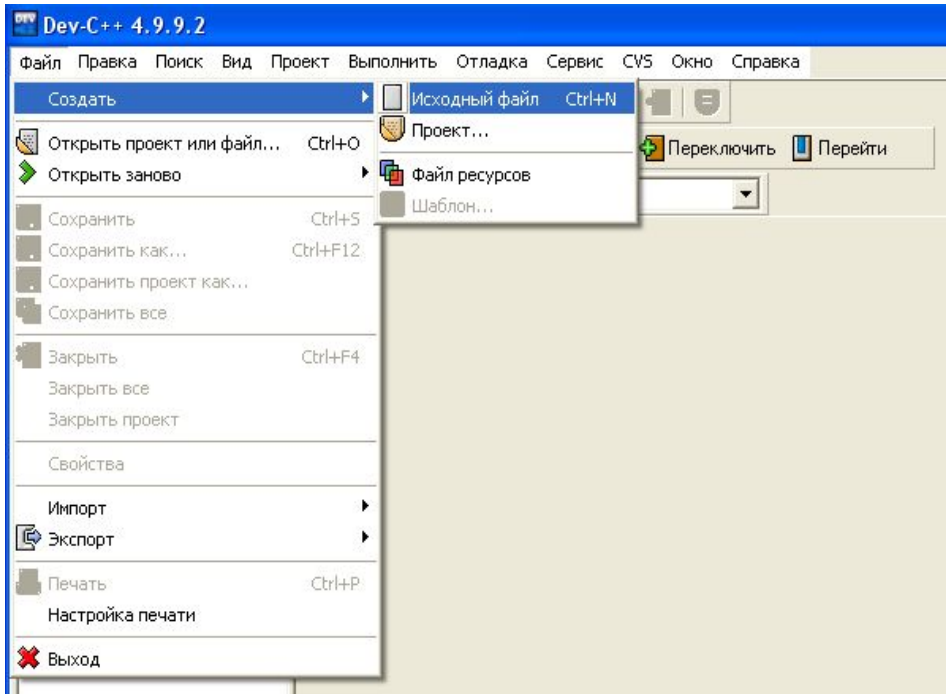
Все языки программирования подразделяются на интерпретаторы и компиляторы.

Интерпретатор(работает медленно)проверяет на ошибки и выполняет программу построчно.(например Питон)

Компилятор сначала проверяет всю программу на ошибки, а затем выполняет ее.

Турбо С++ является компилятором.

sourceforge.net/projects/orwelldevcpp/ — бесплатная среда DevC++ для программирования на C++ в Windows;
<https://visualstudio.microsoft.com/ru/vs/community> — бесплатная среда *Visual Studio Community* для программирования на C++ в Windows и macOS;



Простейшая программа на C++

результат – целое
число (integer)

название программы
main – главный

```
int main ()
```

```
{
```

```
// это основная программа
```

```
/* здесь записывают
```

```
операторы */
```

```
}
```

комментарии после
// не обрабатываются

комментарии внутри
/* */ не обрабатываются



Что делает эта программа?

Вывод на экран

ПОДКЛЮЧИТЬ
библиотеку
`iostream`

input-output streams –
ПОТОКИ ВВОДА И
ВЫВОДА

```
#include <iostream>
using namespace std;
int main() {
    cout << "Привет!"; // вывод текста
    cin.get(); // ждать Enter
}
```

ИСПОЛЬЗОВАТЬ
пространство имён
`std`

`cout` — ПОТОК ДЛЯ ВЫВОДА СИМВОЛОВ
(*character output stream*)

`cin` — ПОТОК ДЛЯ ВВОДА СИМВОЛОВ
(*character input stream*)

Вывод на экран

оператор
вывода

Оператор — это команда
языка программирования.

```
cout << "Привет!";
```

```
cout << "Привет!";  
cout << "Вася!";
```

или так:

```
cout << "Привет!" << "Вася!";
```

```
cout << "Привет, " << "Вася!";
```



Символьные строки
записывают в кавычках!

Переход на новую строку

```
cout << "Привет, Вася!";  
cout << "Привет, Петя!";
```

ожидание:

```
Привет, Вася!  
Привет, Петя!
```

реальность:

```
Привет, Вася!Привет, Петя!
```

перейти
на новую
строку

Решение:

```
cout << "Привет, Вася!" << endl;  
cout << "Привет, Петя!" << endl;
```

end of line

Пример задачи

Задача. Ввести два числа и вычислить их сумму.

```
int main()  
{  
  // ввести два числа  
  // вычислить их сумму  
  // вывести сумму на экран  
}
```

Псевдокод – алгоритм на русском языке с элементами языка программирования.



Компьютер не может исполнить псевдокод!

Зачем нужны переменные?

```
int main()  
{  
  // ввести два числа  
  // вычислить их сумму  
  // вывести сумму на экран  
}
```

Где запомнить?

Переменная — это величина, которая имеет имя, тип и значение. Значение переменной может изменяться во время выполнения программы.

```
int a, b, c;
```

объявление переменных



ячейки памяти

Имена переменных

Идентификатор — это имя программы или переменной.

```
int a, b, c;
```

заглавные и строчные буквы **различаются**

МОЖНО использовать

- латинские буквы (A-Z, a-z)
- цифры



Имя не может начинаться с цифры!

- знак подчеркивания _

НЕЛЬЗЯ использовать ~~скобки, знаки ", &, |, *, +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася "PesBarbos"
TU154 [QuQu] _ABBA A+B

Типы переменных

- **int** – целые

```
int a = 1, b, c = 0;
```

начальные значения

- **float** – вещественные (могут иметь дробную часть)

```
float x = 1.234, y = 3.0, z = 0.576;
```



Целая и дробная части отделяются точкой!

- **double** – вещественное с двойной точностью
- **char** – один символ (в апострофах)

```
char c = 'ю';
```

- **string** – символьная строка (в кавычках)

```
string s = "молоко", q = "я";
```

Работа с переменными

Присваивание (запись значения)

```
a = 5;
```

оператор
присваивания

$a \leftarrow 5$

```
a = 5;  
a = 18;
```

В переменной a будет храниться последнее значение 18

Вывод на экран

```
cout << a;
```



В чём разница?

```
c = 14;  
cout << c;
```

14

```
c = 14;  
cout << "c";
```

c

Работа с переменными

Изменение значения

```
i = i + 1;
```

увеличить на 1

$$i \leftarrow i + 1$$

```
a = 4;
```

```
b = 7;
```

```
a = a + 1;
```

```
b = b + 1;
```

```
a = a + b;
```

```
b = b + a;
```

```
a = a + 2;
```

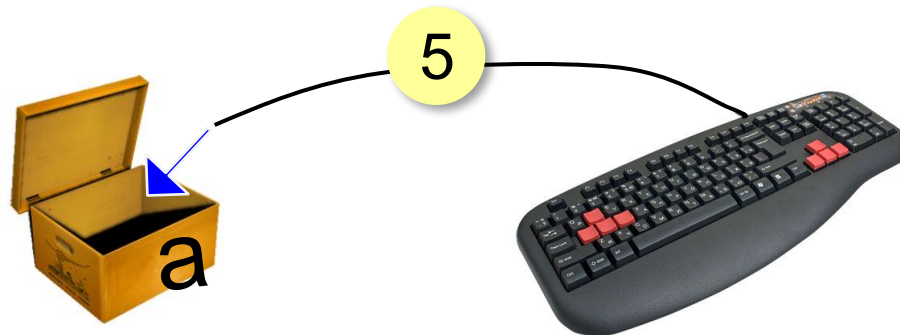
```
b = b + a;
```

a	b
4	
	7
5	
	8
13	
	21
15	
	36

Ввод с клавиатуры

Цель – изменить исходные данные, не меняя программу.

```
cin >> a;
```

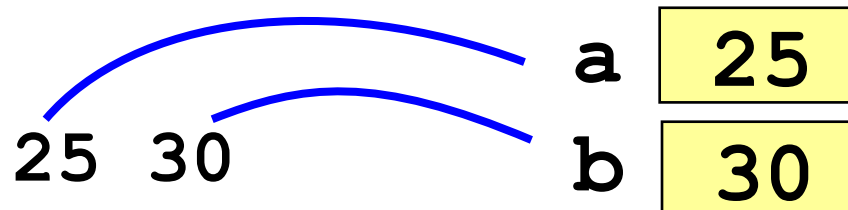


1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

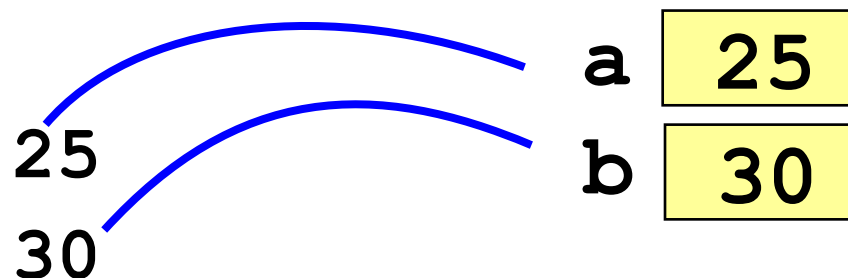
Ввод с клавиатуры

```
cin >> a >> b;
```

через пробел:



через *Enter*:



Программа сложения чисел

```
int main()
{
  int a, b, c;
  cin >> a >> b; // ввести два числа
  c = a + b; // вычислить их сумму
  cout << c; // вывести сумму на экран
}
```



Что плохо?

ожидание:

Введите два числа: 5 7
5+7=12

реальность:

5 7
12



Как улучшить диалог?

cout << данных с текстом

значение *a*

значение *b*

значение *c*

5+7=12

ТЕКСТ

```
cout << a;  
cout << "+" ;  
cout << b;  
cout << "=" ;  
cout << c;
```

```
cout << a << "+"  
    << b << "=" << c;
```

Программа сложения чисел

```
int main()
{
    int a, b, c;
    cout << "Введите два числа: ";
    cin >> a >> b;
    c = a + b;
    cout << a << "+" << b << "=" << c;
}
```



Как переделать для 3-х чисел?

Арифметические выражения

$$a \leftarrow \frac{c + b - 1}{2} \cdot d$$

Линейная запись (в одну строку):

```
a = (c+b-1)/2*d;
```

Операции: + - * – умножение / – деление

Порядок выполнения операций:

- 1) действия в скобках
- 2) умножение и деление, слева направо
- 3) сложение и вычитание, слева направо

```
6 5 2 1 3 4  
a = c + (1 - 2 * b) / 2 * d;
```

Сокращённая запись операций

Полная запись:

```
a = a + b;  
a = a + 1;  
a = a - b;  
a = a - 1;  
a = a * b;  
a = a / b;
```

Сокращённая запись:

```
a += b;  
a += 1;  
a -= b;  
a -= 1;  
a *= b;  
a /= b;
```

```
a++;
```

```
a--;
```

Особенность деления

```
int a = 7, b = 8;  
float x = a / b;
```



Чему равен **x**?



Результат деления целого числа на целое – это целое число (остаток отбрасывается)!

```
int a = 3, b = 4;  
float x;  
x = a / b;  
x = 10 / b;  
x = a / 2;  
x = 10. / b;  
x = a / 2. ;  
x = float(a) / b;
```

Частное и остаток при делении целых

`/` – деление нацело (остаток отбрасывается)

`%` – остаток от деления

175 сек = 2 мин 55 сек



Как получить 2 и 55?

```
int t, m, s;
```

```
t = 175;
```

```
m = t / 60;
```

```
s = t % 60;
```


Частное и остаток при делении целых

 Что получится?

```
n = 123
```

```
d = n / 10;
```

```
k = n % 10;
```

При делении на 10 нацело отбрасывается последняя цифра числа.

Остаток от деления на 10 – это последняя цифра числа.

Форматирование вывода

```
int a = 1, b = 2, c = 3;
cout << a << b << c;
```

123

```
cout << a << " "
     << b << " " << c;
```

1 2 3

```
#include <iomanip>
```

...

```
cout << a
     << setw(3) << b
     << setw(5) << c;
```

1 2 3

3

5

КОЛИЧЕСТВО ЗНАКОВ
НА ВЫВОД ЧИСЛА



Сколько знаков для вывода *a*?

Форматный вывод

```
float x = 12.34567891;  
cout << x;
```

вариант:

12.3457

6 значащих цифр
по умолчанию

```
#include <iomanip>
```

манипуляторы

```
cout << fixed << setw(10)  
      << setprecision(3) << x;
```

12.346

10

в дробной
части

всего на
число

ФОРМАТНЫЙ ВЫВОД

```
float x = 12.34567891;  
cout << fixed;
```

```
cout << setw(8) << setprecision(2)  
      << x;
```

___12.34

```
cout << setw(2) << setprecision(2)  
      << x;
```

12.34

```
cout << setw(0) << setprecision(1)  
      << x;
```

МИНИМАЛЬНО
ВОЗМОЖНОЕ

12.3

Научный формат чисел

```
float x = 123456789;
cout << x;
```

1.23457e+08

$1,23457 \cdot 10^8$

```
float x = 0.0000123456789;
cout << x;
```

1.23457e-005

$1,23457 \cdot 10^{-5}$

```
float x = 0.0000123456789;
cout << scientific
     << setw(10)
     << setprecision(3)
     << x;
```

3
 1.235e-05
 10

Операции с вещественными числами

int – целая часть числа (дробная часть отбрасывается)

round – округление к ближайшему целому

ceil – округление «вверх»

```
#include <cmath>
```

```
float x = 1.6;  
cout << int(x);
```



1

```
cout << round(x);
```



2

```
cout << ceil(x);
```



2

Операции с вещественными числами

$$1/3 = 0,33333\dots$$

бесконечно много знаков



Большинство вещественных чисел хранятся в памяти компьютера с ошибкой!

```
float x, y, z;  
x = 1./2;  
y = 1./3;  
z = 5./6; // 5/6=1/2+1/3  
cout << x+y-z;
```

5.96046e-08

Пробный вариант

1. Найти площадь трапеции. С клавиатуры вводятся три числа a, b, h (a, b – основания, h – высота).

2. С клавиатуры вводиться трехзначное число. Вывести на экран его цифры через запятую.

Решение: (один из возможных вариантов)

Задание 1:

```
#include <iostream>
using namespace std;
int main() {float a,b,h,s;
cin>>a>>b>>h;
s=0.5*(a+b)*h;
cout<<s; cin>>a;
}
```

Задание 2:

```
#include <iostream>
using namespace std;
int main() {int a,z1,z2,z3;
cin>>a;
z3=a%10; a=a/10;
z2=a%10; z1=a/10;
cout<<z1<<','<<z2<<','<<z3;
cin>>a;
}
```