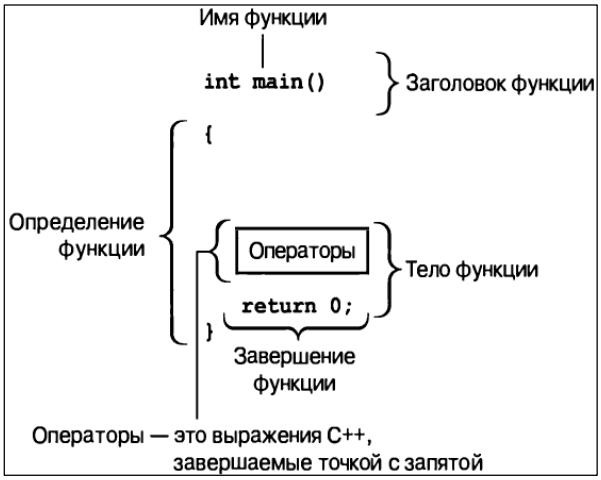
Hello, World !

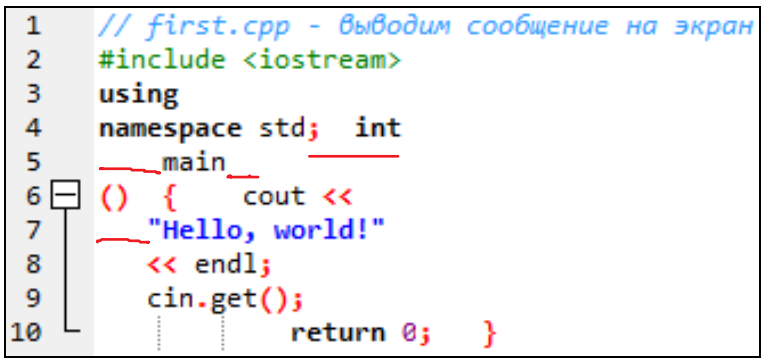
```cpp
// first.cpp - выводим сообщение на экран
#include <iostream>
using namespace std;

int main()
{
   cout << "Hello, world!" << endl;
   cin.get();
   return 0;
}
```

dll

```
main()
main(void)
int  main(void)
void main()        return 0;
```



Форматирование исходного кода C++



```
// first.cpp - выводим сообщение на экран
#include <iostream>
using
namespace std;  int
___main__
() {    cout <<
___"Hello, world!"
   << endl;
   cin.get();
        return 0;   }
```

```
int ma  in()    // пробел в имени; main - лексема
ret
urn 0;          // Enter внутри ключевого слова
"Hello,
   world!"      // Enter внутри строки
```

Правила хорошего стиля оформления кода:

– один оператор в одной строке;

– открывающая и закрывающая фигурные скобки для функции, каждая из которых находится в своей строке;

– операторы в операторных скобках, функциях записаны с отступом от фигурных скобок;

– вокруг круглых скобок, связанных с именем функции, пробельные символы отсутствуют

и др.

Комментарии в C++

```
1  // first.cpp - выводим сообщение на экран
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     /*Но возможно использование многострочных
8     комментариев, ...*/
9     cout << "Hello, world!" /*...в том числе
10    внутри операторов...*/ << endl;
11    cin.get();
12    return /*...и здесь тоже */ 0;
13 }
```

Однострочные комментарии

Многострочный комментарий в стиле C

Препроцессор C++ и файл iostream

*Character **out**put stream (поток вывода символов)*

*Character **in**put stream (поток ввода символов)*

```
1  // first.cpp - выводим сообщение на экран
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7     cout << "Hello, world!" << endl;
8     cin.get();
9     return 0;
10 }
```

Директива препроцессора #include добавляет содержимое файла в код

Определение пространства имён

Препроцессор – это программа, выполняющая обработку файла исходного кода перед началом компиляции.

Пространство имён – способ решения проблемы совпадения имён
CompanyA::count() CompanyB::count()

#include <iomanip>

Грамотный подход использования пространства имён

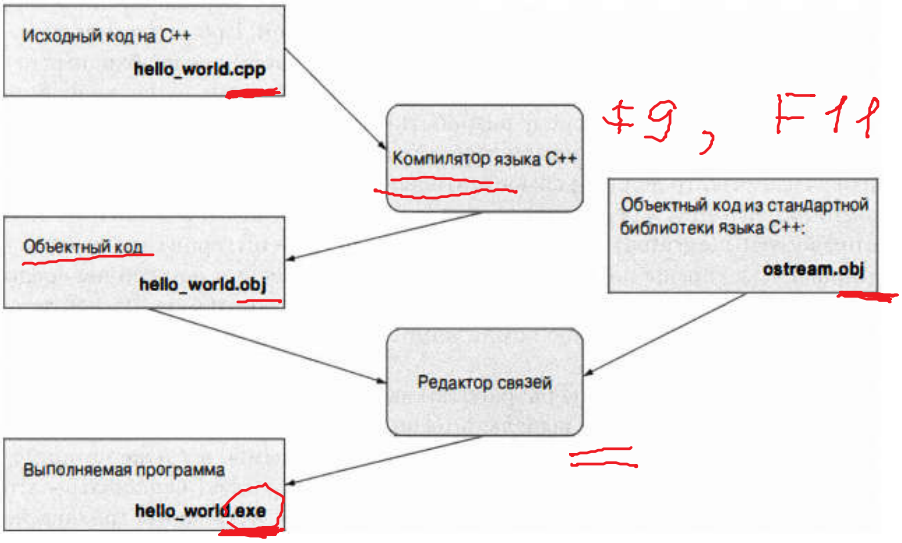
```cpp
// first.cpp - выводим сообщение на экран
#include <iostream>

int main()
{
    std::cout << "Hello, world!";
    std::cout << std::endl;
    std::cin.get();
    return 0;
}
```

```cpp
// first.cpp - выводим сообщение на экран
#include <iostream>
using std::cout;
using std::endl;
using std::cin;
int main()
{
    cout << "Hello, world!";
    cout << endl;
    cin.get();
    return 0;
}
```

Компиляция и компоновка

Запуск программы:

- Исходный код
- Выполняемый (объектный, машинный) код. Компилятор – транслирует программу из текстовой формы, понятной для человека, в форму, понятную для машины.
- Исполняемый код

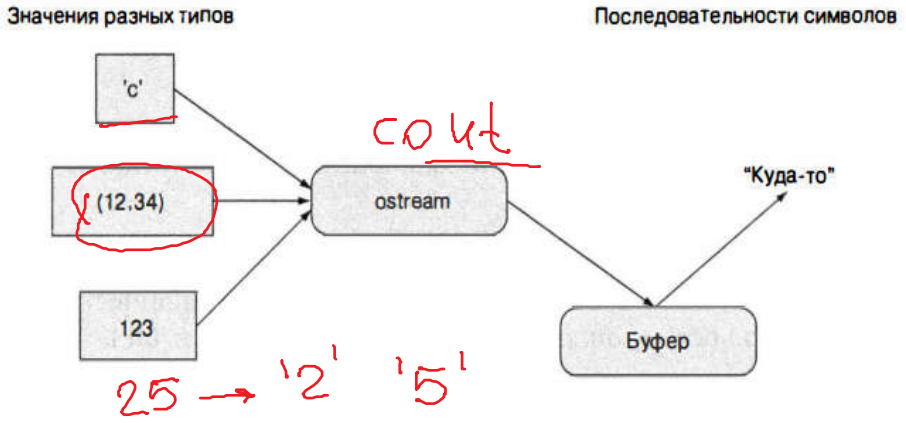


Ошибки:

- времени компиляции
- времени компоновки
- времени выполнения или логические ошибки

Модель ввода/вывода

Вывод данных

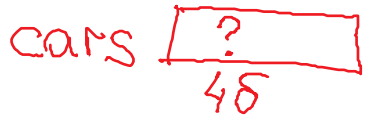


Ввод данных

Программа на C++ – набор функций
Функция – набор операторов

Операторы объявления и переменные

```
// cars.cpp - продажи в автосалоне
// использует и отображает переменную
#include <iostream>

int main() {
    using namespace std;
    int cars;       // объявление переменной целочисленного типа
    cars = 25;      // присваивание значения переменной
    cout << "There are ";
    cout << cars;  // отображение значения переменной
    cout << " cars in the dealership.";
    cout << endl;
    cars = cars - 1; // изменение переменной
    cout << "And now " << cars << " cars." << endl;
    return 0;
}
```




```
There are 25 cars in the dealership.
And now 24 cars.
```

Операторы присваивания

```
// cars.cpp - продажи в автосалоне
// использует и отображает переменную
#include <iostream>

int main() {
    using namespace std;
    int cars;       // объявление переменной целочисленного типа
    cars = 25;      // присваивание значения переменной
    cout << "There are ";
    cout << cars;  // отображение значения переменной
    cout << " cars in the dealership.";
    cout << endl;
    cars = cars - 1; // изменение переменной
    cout << "And now " << cars << " cars." << endl;
    return 0;
}
```

Альтернативный вариант:

```
int a;
int b;
int c;
c = b = a =13;
```

Снова вывод данных

```
// cars.cpp - продажи в автосалоне
// использует и отображает переменную
#include <iostream>

int main() {
    using namespace std;
    int cars;       // объявление переменной целочисленного типа
    cars = 25;      // присваивание значения переменной
    cout << "There are ";
    cout << cars;  // отображение значения переменной
    cout << " cars in the dealership.";
    cout << endl;
    cars = cars - 1; // изменение переменной
    cout << "And now " << cars << " cars." << endl;
    return 0;
}
```

```
cout << "And now ";
cout << cars;
cout << " cars.";
cout << endl;
```

```
cout << "And now "
     << cars
     << " cars."
     << endl;
```

```
// cars.cpp - продажи в автосалоне
// использует и отображает переменную
#include <iostream>

int main() {
    using namespace std;
    int cars;       // объявление переменной целочисленного типа
    cout << "How many cars do you see in dealership?\n";
    cin >> cars;   // чтение значения переменной
    cout << "There are ";
    cout << cars;  // отображение значения переменной
    cout << " cars in the dealership.";
    cout << endl;
    cars = cars - 1; // изменение переменной
    cout << "And now " << cars << " cars." << endl;
    return 0;
}
```