

Правила оформления исходного кода на языке C++

Зарезервированные слова языка C++ (int, return, if, else, while, for и т. д.) пишутся в нижнем регистре.

Названия переменных и функций пишутся в lowerCamelCase (например, helloWorld), имена классов — в UpperCamelCase (например, HelloWorld).

```
C/C++
// Правильно ✓
int main()
{
    char ch;
    int maxValue;

    return 0;
}

void sayHello()
{
}

class HelloWorld {
public:
    void printMessage() {
        int messageCount = 1;
        std::cout << "Hello, world!" << std::endl;
    }
};

// Неправильно ✗
class hello_world {
public:
    void Print_Message() {
        int Message_Count = 1;
        std::cout << "Hello, world!" << std::endl;
    }
}
```

После разделителей (запятая, точка с запятой, двоеточие) ставится пробел, перед разделителем пробел не ставится.

C/C++

// Правильно ✓

```
int a = 1, b = 2;
std::cout << "a: " << a << ", b: " << b << std::endl;
```

// Неправильно ✗

```
int a = 1 ,b = 2 ;
std::cout << "a :" <<a<<",b :" <<b<<std::endl ;
```

Операторы присваивания, математические и логические операторы выделяются с двух сторон пробелами.

C/C++

```
int main() {
    int a = 1;
    int b = 2;

    // Пробелы вокруг оператора сравнения
    if (a < b) {
        // двоеточие в выводе
        std::cout << "a is less than b:" << std::endl;
    }

    return 0;
}
```

Скобки { } переносятся на новую строку, кроме случаев продолжения выражения (например, лямбда-функции или списки инициализации).

```
C/C++
// Правильно ✓
int main()
{
    if (x > 0)
    {
        std::cout << "Positive" << std::endl;
    }

    auto lambda = []() { return 42; };

    std::vector<int> numbers = {1, 2, 3};
}

// Неправильно ✗
int main(){
    if (x > 0){
        std::cout << "Positive" << std::endl;
    }

    auto lambda = []()
    {
        return 42;
    };

    std::vector<int> numbers =
    {
        1, 2, 3
    };
}
```

Внутри блоков кода {} вложенные операторы пишутся каждый на новой строке.

```
C/C++
// Правильно ✓
if (x > 0) {
    std::cout << "Positive" << std::endl;
    x = x - 1;
}

// Неправильно ✗
if (x > 0) {
    std::cout << "Positive" << std::endl; x = x - 1;
}
```

Оператор if оформляется так: if и else пишутся на одном уровне, вложенные операторы внутри if и else отделяются отступом.

```
C/C++
// Правильно ✓
if (x > 0)
{
    std::cout << "Positive" << std::endl;
}
else
{
    std::cout << "Non-positive" << std::endl;
}

// Неправильно ✗
if (x > 0){
std::cout << "Positive" << std::endl;
}
else{
std::cout << "Non-positive" << std::endl;
}
```

Оператор `while` и `for` оформляется так: тело цикла пишется на новой строке с отступом.

```
C/C++
```

```
// Правильно ✓
```

```
for (int i = 0; i < 5; ++i)
{
    std::cout << "i = " << i << std::endl;
}
```

```
int x = 3;
while (x > 0)
{
    std::cout << "x = " << x << std::endl;
    --x;
}
```

```
// Неправильно ✗
```

```
for (int i = 0; i < 5; ++i) std::cout << "i = " << i << std::endl;
```

```
int x = 3;
while (x > 0) {
    std::cout << "x = " << x << std::endl;
    --x;
}
```

Фигурные скобки {} всегда используются для блоков if, else, while, for, даже если внутри одна строка.

```
C/C++
// Правильно ✓
if (x > 0)
{
    std::cout << "Positive" << std::endl;
}
else
{
    std::cout << "Non-positive" << std::endl;
}

for (int i = 0; i < 3; ++i)
{
    std::cout << i << std::endl;
}

while (x > 0)
{
    --x;
}

// Неправильно ✗
if (x > 0)
    std::cout << "Positive" << std::endl;
else
    std::cout << "Non-positive" << std::endl;

for (int i = 0; i < 3; ++i)
    std::cout << i << std::endl;

while (x > 0)
    --x;
```

Максимальная длина строки — 120 символов (для удобного просмотра diff в программах с двумя вертикальными панелями).

Количество файлов исходного кода не должно превышать одного, т.е. один файл для одной программы