

Правила оформления программ на языке C в данном курсе.

Стиль наименования идентификаторов

Самые распространённые стили наименования идентификаторов это:

camelCase

PascalCase

snake_case

Для языка C будем использовать **snake_case** для почти всего, кроме названий новых типов. Для них будем использовать **Pascal_Snake_Case**.

Стиль расстановки скобок

Можете использовать один из двух следующих вариантов расстановки скобок. Любые другие варианты будут считаться ошибкой оформления.

```
int print_numbers(int n)
{
    while (n > 0)
    {
        cout << i << " ";
        n--;
    }
}
```

```
int print_numbers(int n) {
    while (n > 0) {
        cout << i << " ";
        n--;
    }
}
```

Стиль отступов для однострочной области видимости

В случае если тело области видимости состоит из одной строки, то скобки можно не писать.

```
int calculate_sum(int n)
{
    int sum = 0;
    for (int i = 0; i < n; ++i)
        sum += i;

    return sum;
}
```

```
int calculate_sum(int n)
{
    int sum = 0;
    for (int i = 0; i < n; ++i)
    {
        sum += i;
    }
    return sum;
}
```

Исключение – если одна из областей видимости оператора **if else** состоит из более чем одной строки, то все остальные области видимости этого оператора должны обрамляться скобками.

```
if (a > b)
{
    c = a;
}
else
{
    c = b;
}
```

```
if (a > b)
{
    c = a;
}
else
    c = b;
```

Пробелы после запятых

После запятой или точки запятой обязательно ставится пробел. До запятой или точки с запятой пробел не ставится.

```
int func(int a, int b)
{
    for (int i = a; i < b; ++i)
        printf("%i %i\n", a, b);
}
```

```
int func(int a,int b)
{
    for (int i = a;i < b;++i)
        printf("%i %i\n",a,b);
}
```

Пробелы до и после бинарных операторов

До и после бинарного оператора нужно обязательно ставить пробелы.

```
int a = 10;
int b = 20;
int c = a + b;
int d;
d = a + b * c;
```

```
int a = 10;
int b = 20;
int c = a+b;
int d;
d = a + b*c;
```

Отступы

В качестве отступов используется ровно 4 пробела, а не знак табуляции. Настройте ваш текстовый редактор при нажатии на Tab вставлял 4 пробела а не знак табуляции. Причина, по которой желательно использовать пробелы, заключается в том, что символ табуляции может отображаться по-разному в различных программах и настройках редактора. Если вы случайно смешаете пробелы с табами в одном файле, то при открытии кода в другом редакторе все отступы "поедут" и структура кода нарушится.

Объявление несколько переменных на одной строке

Запрещено объявлять несколько переменных на одной строке.

```
int position = 100;
int velocity = 2;
```

```
int position = 100, velocity = 2;
```

Объявление указателей

При объявлении указателей звёздочка обязательно ставится рядом с названием типа.

```
int a = 100;
int* p = &a;
```

```
int a = 100;
int *p = &a;
```

```
int a = 100;
int * p = &a;
```