

Code Style

ИА-031 Сысюк З. Н.

Февраль 2022

1 Введение

В документе содержится описание моего кодстайла

2 Основная часть

Используемые языки программирования

- C/C++

Пробелы и отступы

Операторы и операнды разделяются пробелом:

```
1 int x = ( a + b ) * c / d;
```

Так же пробелом отделяются и фигурные скобки:

```
1 if ( a == 5 ) { return; }
```

Оформление циклов и операторов управления

При использовании циклов или операторов управления используются отступы и переходы на новую строку, если это нужно:

```
1 while(1) {  
2     if(a == 2) {  
3         return;  
4     }  
5     a++;  
6 }
```

Разделение функций и блоков кода

Функции и разные по смыслу блоки кода разделяются пустой строкой:

```
1 int addFirst (...) {
2     ....
3 }
4
5 int addLast (...) {
6     ...
7 }
```

Названия функций и переменных

Названия функций и переменных должны быть логичными, не однобуквенными (названия итераторов могут состоять из одной буквы:

```
1 int c; // BAD
2 int counter; // GOOD
```

Названия функций должны быть записаны в смешанном регистре и начинаться с нижнего:

```
1 int addFirst (...) {
2     ....
3 }
4
5 int addLast (...) {
6     ...
7 }
```

Именованные константы должны быть записаны в верхнем регистре с нижним подчеркиванием в качестве разделителя:

```
1 const int MAX_SIZE = 1000;
2 const int N = 10;
```

Лучшие практики

Используйте текстовую строку, стандартную для C++, а не C. C++ путает тем, что имеет два вида текстовых строк: класс `string` из C++ и старый `char*` (массив символов) из C.

```
1 // Bad practice
2 char* str = "Hello_ there";
3
4 // Good practice
```

```
5 string str = "Hello_there";
```

C++ основан на C, поэтому всегда есть вариант решить задачу «путем C++» и «путем C». Например, когда вы желаете вывести что-либо на системную консоль, вы можете сделать это «путем C++», используя оператор вывода `cout`, в то время как «путем C» вы бы использовали глобальную функцию вроде `printf`.

```
1 // Bad practice
2 printf("Hello ,_world!\n");
3 // Good practice
4 cout << "Hello ,_world!" << endl;
```

Файлы исходных кодов

Класс следует объявлять в заголовочном файле и определять (реализовывать) в файле исходного кода, имена файлов совпадают с именем класса:

```
1 myClass.h myClass.cpp
```

Все определения должны находиться в файлах исходного кода:

```
1 class MyClass
2 {
3 public:
4     int getValue () {return value_;} // WRONG
5     ...
6
7 private:
8     int value_;
9 }
```

Заголовочные файлы объявляют интерфейс, файлы исходного кода его реализовывают.

Не следует объявлять переменные класса как `public`. Вместо этого нужно использовать переменные с модификатором `private` и соответствующие функции доступа.

3 Вывод

В ходе работы я описал свой стиль написания кода в редакторе *LATEX*.

4 Список литературы

1. <https://tproger.ru/translations/stanford-cpp-style-guide/>
2. <https://habr.com/ru/post/172091/>
3. <https://habr.com/ru/company/ruvds/blog/574352/>