

Pinchuk Ruslan IA-032 code style

Руслан Пинчук ИА-032;

SibSUTIS, email: boss.ruslan1212@inbox.ru

github: NelicePriyatniy

February 2022

1 Introduction

Создание Код-Стайла для C++.

2 C++

2.1 Version

Используется версия C++199711.

2.2 Header Files

Рекомендации разделяются на степень важности: обязательные, настоятельно рекомендуемые и общие.

Основная цель рекомендаций — улучшение читаемости и, следовательно, ясности и лёгкости поддержки, а также общего качества кода.

Рассмотрим общие рекомендации:

1. Допускаются любые нарушения рекомендаций, если это улучшает читаемость.
2. Правила могут быть нарушены, если против них есть персональные возражения. Это попытка создать набор общих рекомендаций, не навязывая всем единый стиль. Опытные программисты обычно всё равно подгоняют стиль под себя.

3. Скобки и пробелы:

3.1 Отделяйте пробелами фигурные скобки:

```
int x = 3;
int y = 7;
double z = 4.25;
```

```
x++;
if (a == b) {
    foo();
}
```

3.2 Ставьте пробелы между операторами и операндами:

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

4. Соглашения об именовании:

4.1 Имена, представляющие типы, должны быть обязательно написаны в смешанном регистре, начиная с верхнего.

```
Line, SavingsAccount
```

4.2 Имена переменных должны быть записаны в смешанном регистре, начиная с нижнего.

```
line, savingsAccount
```

4.3 Названия методов и функций должны быть глаголами, быть записанными в смешанном регистре и начинаться с нижнего.

```
getName(), computeTotalWidth()
```

4.4 Все имена следует записывать по-английски.

```
computeAverage(); // НЕЛЬЗЯ: compAvg();
```

4.5 Следует избегать сокращений в именах.

```
line, savingsAccount
```

5. Базовые выражения C++

5.1 Используйте оператор вывода cout вместо printf

```
cout << "Hello, world!" << endl;
```

5.2 Используйте цикл for, когда вы знаете количество повторений, а цикл while, когда количество повторений неизвестно

5.3 Если у вас есть выражение if / else, которое возвращает логическое значение, возвращайте результаты теста напрямую:

```
return score1 == score2;
```

5.4 Никогда не проверяйте значения логического типа, используя `==` или `!=` с `true` или `false`:

6. Комментарии

6.1 Заглавный комментарий. Размещайте заглавный комментарий, который описывает назначение файла, вверху каждого файла. Предположите, что читатель вашего комментария является продвинутым программистом, но не кем-то, кто уже видел ваш код ранее.

6.2 Заголовок функции / конструктора. Разместите заголовочный комментарий на каждом конструкторе и функции вашего файла. Заголовок должен описывать поведение и / или цель функции.

6.3 Параметры / возврат. Если ваша функция принимает параметры, то кратко опишите их цель и смысл. Если ваша функция возвращает значение — кратко опишите, что она возвращает.

6.4 Исключения. Если ваша функция намеренно выдает какие-то исключения для определенных ошибочных случаев, то это требует упоминания.

6.5 Комментарии на одной строке. Если внутри функции имеется секция кода, которая длинна, сложна или непонятна, то кратко опишите её назначение.

6.6 TODO. Следует удалить все `// TODO` комментарии перед тем, как заканчивать и сдавать программу.

3 Список литературы:

Сайт: <https://tproger.ru/translations/stanford-cpp-style-guide/>

Который использовал материалы :

<https://stanford.edu/class/archive/cs/cs106b/cs106b.1158/styleguide.shtml>

Документ: <https://habr.com/ru/post/172091/>

Использующий материалы:

Code Complete, Steve McConnell — Microsoft Press

Programming in C++, Rules and Recommendations, M Henricson, e. Nyquist,
Ellementel (Swedish telecom): <http://www.doc.ic.ac.uk/lab/cplus/c>

Wildfire C++ Programming Style, Keith Gabryelski, Wildfire Communications
Inc.: <http://www.wildfire.com/ag/Engineering/Development/C++Style/>

C++ Coding Standard, Todd Hoff: <http://www.possibility.com/Cpp/CppCodingStandard.htm>

Doxygen documentation system: <http://www.stack.nl/~dimitri/doxygen/index.html>