

CodeStyle

ИА-031 Шолохов Владислав

github: @SBZ03; email: sholokhov.02@list.ru

February 2022

1 Введение

Code style как стандарт разработки. Code style помогает обеспечивать линейное развитие проекта, значительно ускоряет адаптацию новых сотрудников и, в целом, формирует и воспитывает культуру разработки. [1]

2 Язык программирования C/C++

2.1 Отступы и пробелы

Для отступа используется табуляция (размер таба 4 пробела).

```
1 int main(){  
2     ...  
3     return 0;  
4 }
```

Операторы и операнды разделяются пробелом.

```
1 int a = (b + c) * e;
```

Перед открывающей скобкой и после знаков препинания используется пробел.

```
1 for (...; ...; ...)
```

Перед двоеточием в конце строки(если оно нужно) пробел не ставится.

2.2 Объявление переменных

Переменным даются описательные имена, такие как `firstName` или `counter`.

Избегаются однобуквенные названия вроде `x` или `s`, за исключением итераторов вроде `i,j,k`.

Если переменная используется лишь внутри определенного `if`, то делайте её локальной, объявляя в том же блоке кода, а не глобальной.

Выбирается подходящий тип данных для ваших переменных. Если переменная содержит лишь целые числа, то она определяется как `int`, а не `double`.

```
1 int counter;  
2 double average;
```

Используется текстовая строка, стандартная для C++.

```
1 string str = "Hello there;
```

Если определенная константа часто используется в коде, то она обозначается как `const` и мы всегда ссылаемся на данную константу, а не на её значение.

```
1 const int VOTINGAGE = 18;
```

2.3 Функции

Между функциями и группами выражений остаётся пустая строка.

```
1 void fun1(...) {  
2     ...  
3 }  
4 //пустая строка  
5 void fun2(...) {  
6     ...  
7 }
```

Имена функций должны быть логическими и записаны с нижнего регистра, но могут быть записаны в смешанном регистре.

Порядок их написания не должен вызывать ошибок, функции должны быть записаны в логическом порядке

```
1 int createMatrix(...) {  
2     ...  
3 }  
4  
5 void deleteMatrix(...) {  
6     ...  
7 }  
8  
9 void printMatrix(...) {  
10     ...  
11     deleteMatrix(...);  
12 }
```

2.4 Операторы управления

Итерационные операторы (while, for, do/while).

Используются отступы и переходы на новую строку, если есть в этом необходимость. Для счётчика используются итераторы вроде i,j,k.

```
1 while (i < N){  
2     ...  
3 }
```

```
1 for (int j = 0; j < N; j++){  
2     ...  
3 }
```

```
1 do{  
2     ...  
3 } while(k < N);
```

Операторы выбора(if else,switch). Используются отступы и переходы на новую строку, если есть в этом необходимость. Для if фигурные скобки ставятся в любом случае. Для else фигурные скобки ставятся случае если действий более одного.

```
1 if (x > y){  
2     cout « "Первое число больше";  
3 } else cout « "Второе число больше";
```

```
1 switch (x){  
2     case 1:  
3         ...;  
4         continue;  
5     case 2:  
6         ...;  
7         break;  
8 }
```

2.5 Структуры

Структуры, объявляются с помощью typedef. Сами названия пишутся с большой буквы.

```
1 typedef struct Node {  
2     int vaule;  
3     struct Node *left;  
4     struct Node *right;  
5 } Node;
```

2.6 Классы

Отделяйте ваши объекты, делая все поля данных в вашем классе private. Всегда размещайте объявления классов и их частей в собственные файлы, ClassName.h.

```
1 class MyClass{  
2     public:  
3         int value;  
4     private:  
5         int getValue(){  
6             return value;  
7         }  
8 }
```

3 Вывод

Я освоил LaTeX и научился им пользоваться. Составил CodeStayle.

Список литературы

[1] Code style как стандарт разработки. - URL:
<https://habr.com/ru/company/manychat/blog/468953/>

[2] Code style как стандарт разработки. - URL:
<https://tproger.ru/translations/stanford-cpp-style-guide/>

[3] Code style как стандарт разработки. - URL:
<https://habr.com/ru/company/ruvds/blog/574352/>