

Форматирование

1. Используйте табуляцию, а не пробелы для отступов.
2. Открывающая и закрывающие фигурные скобки на отдельной строке.
3. Слева и справа от операторов `.`, `->` не ставятся пробелы.
4. Унарные операторы `--`, `++`, `*`, `&`, `...` не отделяются от аргумента пробелом.
5. После запятой ставится пробел, а перед `—` нет. Аналогично для точки с запятой внутри выражения `for`.
6. Пустой оператор `[]` не отделяется пробелами.
7. Пустые скобки `()` не отделяются пробелами.
8. С использованием круглых скобок и квадратных ставятся пробел после открытия и перед закрытием скобки.
9. В классах и структурах, модификаторы доступа пишутся на том же уровне, что и `class/struct`, а остальной код с отступом.
10. Не пишите несколько выражений в одной строке.
11. Внутри функций группируйте блоки кода, отделяя их не более, чем одной пустой строкой.
12. `const` (относящийся к значению) пишется до имени типа.
13. При объявлении указателя или ссылки, символы `*` и `&` пишутся слитно с типом данных
14. Нельзя объявлять несколько переменных разных типов в одном выражении.

Имена

1. Имена функций — `camelCase`, имена начинаются с глагола.
2. Имена публичных методов — `Pascal Case`, , имена начинаются с глагола
3. Имена классов (структур) — `Pascal Case`
4. Имена констант, элементов перечисления — `Upper case`
5. Имена локальных переменных - `camelCase`
6. Для `private` полей используется префикс `_` и далее `camelCase`
7. `Get Set` методы начинаются с префикса `Get`, `Set` соответственно.