

Правила оформления исходного кода

1. Использование стандартов к оформлению кода.

У каждого языка программирования есть свой стандарт оформления кода, который говорит, как надо делать отступы, где ставить пробелы и скобки, как называть объекты, как комментировать код и т.д. Например, стандарты оформления кода приведены <https://github.com/google/styleguide>.

2. Использование для переменных (констант, объектов, классов, процедур, функций, методов, пакетов и т.д.) понятных имён.

Имена переменных, констант, объектов, классов и типов должны быть существительными. Название метода или функции должно содержать глагол. Если метод или функция определяет, является ли какая-то информация об объекте истинной или ложной, его имя должно начинаться с «is». Методы, которые возвращают свойства объектов, должны начинаться с «get», а устанавливающие значения свойств — «set». Константы указываются большими буквами.

3. Использование комментариев.

Каждый метод и процедура должна начинаться с описания в комментарии того, что данный метод, процедура или функция делает, а также параметров, возвращаемого значения и возможных ошибок, и исключений. Опишите в комментариях роль каждого файла и класса, содержимое каждого поля класса и основные шаги сложного кода. Комментарии рекомендуется писать по мере разработки кода.

4. Использование подпрограмм.

Подпрограмма — поименованная или иным образом идентифицированная часть, содержащая описание определённого набора действий. Подпрограмма может быть многократно вызвана из разных частей программы. Подпрограммы необходимы для:

- Декомпозиции программной задачи на простые(понятные) шаги;
- Сокращение (в идеале полное устранение) дублированного кода;

- Возможности повторного использования кода в других программах;
- Улучшения прослеживания.

5. Недопустима обфускация кода

Обфускация (от лат. *obfuscare* — затенять, затемнять; и англ. *obfuscate* — делать неочевидным, запутанным, сбивать с толку) или запутывание кода — приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.