**Чистый код**

Код который мы написали 1 раз будет много раз перечитываться, он нужен для легкого визуального понятия.

Почему?

* Быстрое чтение
* Легко исправить (быстро и дешево)
* Менее сложный
* Меньше дефектов
* Выглядит профессионально

Как поддерживать чистоту кода?

* Поставить целью команды
* Регулярная задача
* Стандарты (официальные код-конвеншены)
* Практики (практика парного программирования, практика код-ревью, пул-реквест)
* Инструменты (инструменты, которые позволят выявить ошибки в автоматическом режиме без участия человека)

**Code convention**

Соглашения по кодированию — это набор рекомендаций для конкретного языка программирования, в которых рекомендуются стиль, практика и методы программирования для каждого аспекта программы, написанной на этом языке.

**Как правильно называть переменные**

Избегать очевидное название переменных как result, status

Избегать цифр в названии переменных

Сокращать название переменных нельзя

Не использовать однобуквенные переменные

**Как правильно назвать классы**

Имена классов должны начинаться с латинского символа и могут содержать в себе символ дефиса (-) и подчеркивания (\_). Использование русских букв в именах классов недопустимо.

**Как правильно называть функции**

Нужно использовать длинные имена и глаголы

Нельзя: делать 2 действия в одном методе. 1 метод – 1 задача

Название метода должно соответствовать своей задаче

Не создавать функцию с большим количеством параметров

**Комментарии**

Использовать можно

- указывать лицензию

Не стоит

/\*\*/

//TODO

Комментирование понятных частей кода

Комментирование if, while, else, try

Закомментированный код нельзя комментировать

Нельзя чтобы у каждого метода был свой комментарий

**Избыточный код**

Если мы хотим что-то вернуть, то сразу же возвращаем. Не создаем отдельных переменных.

Применяем тернарные операторы

Не нужно:

- не рассчитываем на чтение с консоли

- не сравнивать Boolean с Boolean

-не создаем переменные для ее возвращения

**Исключения**

Стоит использовать!

из названия исключения должно быть понятно, что произошло не так

Не стоит: оставлять пустой catch

**Тесты**

Нужно:

Д.б assert

1 тест – 1 assert

Правильные и хорошие название тестов, по которому можно понять суть этого теста

Д. запускаться в любом порядке(как бы не запустил тест результат один и тот же, тесты не должны зависить друг от друга)

Избегать:

В названии тестов указывать тестовые данные

Избегать for, while, if/else, switch, try, catch

**Дизайн-принципы**

2 принципа Kiss и Dry

Kiss (оставь это простым)

Dry (не повторяйся)

Избегать:

-Слепое копирование

-Не писать в main а потом разбивать на классы

-сразу исправлять грязный код

**Инструменты**

Code convection

Code Review (парное программирование)

Static Code Analyzers (специальные программы, которые читают код и пытаются найти ошибки)