

6 лабораторная работа по принципам S.O.L.I.D.

Краткие теоретические сведения. (1)

S.

S - Single Responsibility Principle - принцип единственной ответственности. Каждый класс должен иметь только одну зону ответственности.

O.

O - Open closed Principle - принцип открытости-закрытости. Классы должны быть открыты для расширения, но закрыты для изменения.

L.

L - Liskov substitution Principle - принцип подстановки Барбары Лисков. Должна быть возможность вместо базового (родительского) типа (класса) подставить любой его подтип (класс-наследник), при этом работа программы не должна измениться.

I.

I - Interface Segregation Principle - принцип разделения интерфейсов. Данный принцип обозначает, что не нужно заставлять клиента (класс) реализовывать интерфейс, который не имеет к нему отношения.

D.

D - Dependency Inversion Principle - принцип инверсии зависимостей. Модули верхнего уровня не должны зависеть от модулей нижнего уровня. И те, и другие должны зависеть от абстракции. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций

Задание I.

Посмотреть код, приведенный в файле `uartworker.cpp`. Применить принципы SOLID (можно не все), аргументировать.

Задание II.

Применить принципы SOLID(можно не все) на своём коде лабораторных работ 4-5. Аргументировать выбор.

Полезные ссылки:

1. <https://habr.com/ru/articles/688530/>
2. <https://javarush.com/groups/posts/osnovnye-principy-dizajna-klassov-solid-v-java>
3. https://habr.com/ru/companies/productivity_inside/articles/505430/
4. <https://www.youtube.com/watch?v=TxZwqVTaCmA>