public class TextManipulator { private String text; public TextManipulator(String text) { this.text = text; } public String getText() { return text; } public void appendText(String newText) { text = text.concat(newText); } public String findWordAndReplace(String word, String replacementWord) { if (text.contains(word)) { text = text.replace(word, replacementWord); } return text; } public String findWordAndDelete(String word) { if (text.contains(word)) { text = text.replace(word, ""); } return text; } public void printText() { System.out.println(textManipulator.getText()); } }

SOLID — это акроним, образованный из заглавных букв первых пяти принципов ООП и проектирования.

S(Single Responsibility Principle) - принцип единственной ответственности - каждый класс выполняет лишь одну задачу. Легкая модификация в будущем, простое тестирование, класс не имеет зависимостей на другие классы.

O(Open Closed Principle) - принцип открытости/закрытости - программные сущности открыты для расширения и закрыты для модификации. Чтобы не сломать логику в классе-родителе, мы унаследуемся от него и реализуем что-то своё, и используем свой класс.

L(Liskov’s Substitution Principle) - принцип подстановки барбары лисков - объекты в программе можно заменить их наследниками без изменения свойств программы.

I(Interface Segregation Principle) - принцип разделения интерфейса - много специализированных интерфейсов лучше, чем один общий

D(Dependency Inversion Principle) - принцип инверсии зависимостей - зависимость на абстракциях. Модули верхних уровней не должны зависеть от модулей нижних уровней. Оба типа модулей должны зависеть от абстракций. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.

[https://docs.google.com/spreadsheets/d/1owGOr08-YSywk3KVIAdMLX8\_tZyvPm9UJWubEeOcgD8/htmlview#](https://docs.google.com/spreadsheets/d/1owGOr08-YSywk3KVIAdMLX8_tZyvPm9UJWubEeOcgD8/htmlview) методичка