## Identify SOLID Principles

Choose a large open source project written in TypeScript preferably (choose another OOP language if you want: Java, C#, Ruby). You may want to look at **GitHub**. A project may be considered large enough if it contains at least 30 classes.

Try to identify **at least** **3 or more (preferably)** examples of the SOLID principles. Document them by filling in the table below. Note that you don't need to give code examples itself, just provide the link to file with line numbers range (or whatever you want to clearly recognize the example you describe, e.g. **file** *hello.ts:15-36*, **class** *Foo*), you can also write free text.

|  |  |
| --- | --- |
| **Principle** | **Examples** |
| Single Responsibility Principle |  |
|  |
| Open / Closed Principle |  |
|  |
| Liskov Substitution Principle |  |
|  |
| Interface Seggregation Principle |  |
|  |
| Dependency Inversion Principle |  |
|  |

## Violations of SOLID and Other Principles

Try to find **at least 3 or more (preferably)** violations of the SOLID principles in the project you have chosen for Problem 1 and document it. Additionally, you can describe other (DRY / KISS / YAGNI / etc.) violations.

You may provide short descriptions about how to refactor/improve such violations.

**Optionally**, you can add small examples with results of such refactoring using pseudo-code or real code.

**Evaluation Criteria:**

**0** – Nothing was done

**2** – Examples of the SOLID principles was provided but contains less then 3 cases for each of them

**3** – Examples of the SOLID principles was fully done, there are no examples of violations

**4** – Violations of the SOLID principles was provided as well, but each table contains less then 3 examples.

**5** – All principles covered with at least 3 or more examples of SOLID cases and violations

**5+** – Results of refactoring and additional violations (DRY, KISS, YAGNI, etc.) was provided