

Why was the previous API not independently testable?

The main reason that the previous design was unable to be tested independently was the lack of abstraction. This violated the Dependency Inversion Principle, and resulted in high coupling as classes depended on concrete implementations rather than abstractions. Hence, testing could not be performed independently as classes could not be tested as a single unit, due to their reliance on other concrete classes.

To remedy this issue, I implemented two main changes to the API. The first change was the replacement of each class with an interface, and moving their implementation to a separate file. For example, `Student.class` was moved to `StudentImpl.class`. The second change involved removing any constructor calls within each class, and instead passing in abstract objects. For example, in `EnrolmentSystem`, the method `addFaculty(String name, String abbreviation)` was changed to `addFaculty(Faculty f)`. Below are the specific changes made to each original class.

EnrolmentSystem.class

This class, like all others, was changed to an interface, and its implementation moved to `EnrolmentSystemImpl.class`. Through this, the coupling between the class and other classes, such as `ReportGenerator`, and allow it to be mocked using Mockito.

Additionally, the methods `addStudent()`, `addFaculty()` and `addUnit()` were modified to avoid calling constructors of implemented classes. This was done through passing the instantiated object as an argument rather than the object's parameters. I also added the method `setReportGenerator()` to avoid calling the `ReportGenerator` constructor inside the `EnrolmentSystem` class. This again further reduced coupling between `EnrolmentSystem` and other classes and allowed it to be tested independently from them without the need for mocking.

Student.class

This class, like all others, was changed to an interface, and its implementation moved to `EnrolmentSystemImpl.class`. Through this, the coupling between the class and other classes, such as `EnrolmentSystem`, and allow it to be mocked using Mockito.

Faculty.class

This class, like all others, was changed to an interface, and its implementation moved to `EnrolmentSystemImpl.class`. Through this, the coupling between the class and other classes, such as `EnrolmentSystem`, and allow it to be mocked using Mockito.

Unit.class

This class, like all others, was changed to an interface, and its implementation moved to `EnrolmentSystemImpl.class`. Through this, the coupling between the class and other classes, such as `EnrolmentSystem`, and allow it to be mocked using Mockito.

ReportGenerator.class

This class, like all others, was changed to an interface, and its implementation moved to `EnrolmentSystemImpl.class`. Through this, the coupling between the class and other classes, such as `EnrolmentSystem`, and allow it to be mocked using Mockito.

Main.class

This class was not modified as it was not relevant to the functionality of the API.