Design Patterns

Exercise

1) Pattern Abstraction

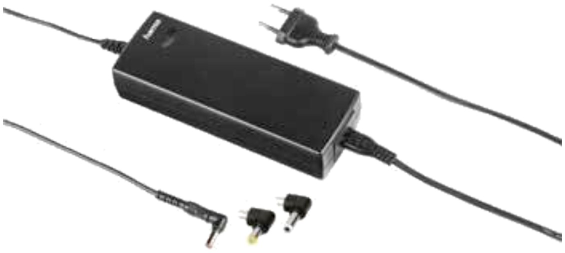

Figure 1: Notebook power supply

Imagine a notebook power supply (see Figure 1). Discuss the similarities to different patterns used in software engineering. Which pattern fits best? Justify your solution.

2) Validation Controller

Write a validation runner, which iterates through a Unicase project and validates every functional requirement and every use case with your Validator, implemented in Exercise 2. The validation runner returns a set of model elements (functional requirements and use cases) that failed the validation. Write a test case to test the validation runner.
Hint: Project supports a method *getAllModelElementsByClass* returning all model elements of a certain class in the project. See video tutorial on validation controller for details.

3) Adapter Pattern

a) Validation in Unicase

Unicase has a view to show the result of the build-in validation. You can open it by right-clicking a project and the selecting “Other” => “Validate Project”. Validate your project and try to solve all the resulting warnings.

b) Adapting your Validator

To show a validation warning in the existing view in Unicase, a Validator has to extend the abstract class `AbstractModelConstraint`. Use the adapter pattern to show the result of the “Weak Phrases”-Validator in the existing Unicase Validation view.
Hint: Watch the video tutorial on validation for details.

4) Observer Pattern

The Observer pattern allows decoupling a class that is being observed from an observer. In unicase all model elements are contained in a project. To listen to changes on model elements on a project, you can register as ProjectChangeObserver at the project. After registration an observer will subsequently be notified on any change on a model element in the respective project.
Implement a test case where you register as an observer to the project and check whether you are notified when a new functional requirement is added to the project.
Hint: Watch the video tutorial on observer for details.

This exercise is due on June 3rd, however one additional task will be published on exercise sheet 6, so please plan for this!