Department of Computer Science 3
RWTH Aachen University

Software Language Engineering, Summer 2020
Exercise-Sheet 2

Exercise Sheet 2: Visitors and Context Conditions

Organizational issues

Keep in mind that the chosen **representative** of your group hands in your solution as Moodle does **not yet support group solutions**. Always make sure to include your names in a groupmembers.txt in your solution.
Hint:
If you have questions/problems solving this task, use the reference manual available here: http://www.se-rwth.de/publications/MontiCore-5-Language-Workbench-Edition-2017.pdf to find further explanations.

Exercise 2.1: Develop a PrettyPrinter (14 Points)

In this task you will develop a *PrettyPrinter* based on the Visitor concept. Unzip the provided project and make yourself familiar with the given travel costs modeling language project. Similar to the former projects the grammar file to start with is located in "`src/main/grammars`". Furthermore, there are 3 example models in "`src/test/resources/examples`" that are valid models of the grammar. The skeleton of the Java class for the *PrettyPrinter* `TravelCostsPrettyPrinter` is located in the package `prettyprint`, a corresponding – currently ignored but failing – test `TravelCostsPrettyPrinterTest` for your pretty printer also exists.

a) Implement the missing parts of the PrettyPrinter implementation. Make use of
   - the printing abilities of the printers super class and the contained `IndentPrinter`
   - the traversal strategy provided as default implementations in the visitor interface.

   However, carefully decide when to implement/override visit, endVisit, handle or traverse methods for printing. If your PrettyPrinter implementation is correct, the corresponding test should not fail anymore. Remove the `@Ignored` annotation such that your implementation is tested.

b) Which parts of your PrettyPrinter and thus of your grammar are not tested by the given example models? Provide this answer in an exercise2.1b.txt file and provide an example model "Example4.tc" that covers the missing modelling elements. Extend the PrettyPrinter test such that your model is used for testing as well.

Exercise 2.2: Developing Context Conditions (8 Points)

The grammar only provides the context-free syntax of the language, but there are several additional constraints that need to be fulfilled to have well-formed models. For the travel costs modeling language this mainly concerns time constraint, e.g., the arrival must be completed before the departure may take place. Therefore, the language must be accompanied by a set of context condition. In this task you will develop some of these conditions.

a) The project already has a skeleton `ArrivalBeforeDepartureCoCo` of a context condition to check that the arrival happened before the departure. A corresponding – currently ignored but failing – test `ArrivalBeforeDepartureCoCoTest` for this context condition also exists. Complete the implementation of this context condition. Make use of the `LocalDate` and `LocalDateTime` classes provided by Java to compare dates. If your implementation is correct, the corresponding test should not fail anymore. Remove the `@Ignored` annotation such that your implementation is tested.

b) Add two more context conditions `MinutesValidCoCo` and `HoursValidCoCo` that check that the highest value for minutes is 59 and for hours 23. Provide one invalid model per context condition with invalid time for workdays and implement corresponding tests `MinutesValidCoCoTest` and `HoursValidCoCoTest` that checks the correctness of your context conditions similar to the test `ArrivalBeforeDepartureCoCoTest`.
   - What happens if you use invalid minute or hour values for arrival or departure? Provide this answer in an exercise2.2b.txt file.

As a solution of this task, zip and hand in your project **without** the generated target, i.e., after running `mvn clean`. Make sure the generated files are reproducible by just running `mvn install`. **Not working projects, e.g., projects that have compile errors, will not gain you any points**.