Generating Software from Specifications WS 2013/14 - Assignment 1

Published: Oct 15 -- Turn in until Oct 23, 12h

What to turn in:

Please, prepare for this and every subsequent homework a brief documentation. It should contain your statements, questions, and brief explanations on the following items:

- Which task did you work on?
- What were the main steps of the task?
- Did you manage to perform them successfully?
- Did you encounter any problems? Describe them briefly.
- Could you solve them?
- What did you learn?
- Did any questions remain open?

In order to produce a meaningful documentation of your work you need to take notes while you work on an assignment! Do **not turn any technical material** you produced when solving the task, unless you are explicitly asked to, or if it is necessary to document a problem.

Please, email such a documentation of your work to uwe@upb.de. The mail should reach me until noon on the day before the next lecture.

Use for all those emails a subject of the form **Subject: GSS Assignment{nr} {Name}**.

Exercise 1 (Explore the Wrapper Generator)

The Wrapper Generator is a tool, that supports developing programs in C++. It generates wrapper classes for given type names. Such a class wraps an arbitrary value of the given type into an object of the class. That is particularly useful for basic types and for pointer types. All these classes will a common super-class named Object. It may be used to create heterogeneous containers, as in Java.

In the directory `blatt1/wrapper` you find all files you need to create and to use the Wrapper Generator:

- the specification for Eli to create the generator: `Wrapper.fw`,
- a main program `trywrapper.cc` and an interface file `Pair.h` (both in C++); they use a module to be produced by the Wrapper Generator.

Execute the following steps to generate and to explore the Wrapper Generator. Do not forget to take notes for your work documentation when you execute the steps!

1. Open the manuals of Eli at

   ```
   http://eli-project.sourceforge.net
   ```

2. Create a directory for your work in GSS; create a sub-directory for your work on the Wrapper Generator, and download the files provided for this homework.
3. Start Eli and let it create a cache in the "/tmp" directory:

   ```
   /comp/eli/current/bin/eli -c /tmp/CACHE
   ```

   If you are not allowed to create a cache under that name, try a different name.
4. Let Eli check whether it can construct the Wrapper Generator without warnings:

   ```
   Wrapper.fw:exe:warning >
   ```

   If warnings or error messages are given, eliminate the problems.
5. Generate the executable and copy it into your working directory:

```
Wrapper.fw:exe > .
```

6. Create an input file for the Wrapper Generator, which fits to the provided test program (see Wrapper.fw), and call your Wrapper Generator for it:

   ```
   ./Wrapper.fw.exe example
   ```

7. What does the Wrapper Generator produce? Study the results.
8. Compile the generated files together with the main program:

   ```
   g++ -o pair.e example.cc trywrapper.cc
   ```

   and execute it

   ```
   ./pair.e
   ```

9. Extend the example such that further wrapper classes are generated.

Exercise 2 (Structure of Specifications for Eli)

This task will introduce you to the structure of specifications for Eli using the one for the Wrapper Generator.

Let Eli generate the documented specification of the Wrapper Generator:

```
Wrapper.fw:pdf> ./Wrapper.fw.pdf
```

When reading the description, you will recognize that its documentation ends early after a few paragraphs.

It is your task to complete the documentation stepwise, while you learn more about how to develop DSLs using Eli during subsequent lectures. This week you are asked only to find out **which type** every specification fragment in the file has. You can find that information in Eli’s documentation on "Guide for New Eli Users". Write into the wrapper specification for each specification type what kind of specifications are made by texts of that type, and where those specifications are explained in the Eli documentation. Re-generate your extended description.