Third Task

Programming Languages

LVA 185.208
2025 S
TU Wien

What to do

Develop a syntax-aware editor for programs in the language developed in Task 2. The editor shall be written in a statically typed functional language like Haskell or ML. In addition to the usual functionality of an editor (loading, showing, modifying and storing text), the editor shall highlight

- obvious syntactic errors like unbalanced braces,
- other occurrences of a name pointed to by the cursor,
- the matching brace when the cursor points to a brace,
- and similar useful things depending on the language.

For highlighting you can use text attributes like color, underlining, thickness, inversion and size.

It is not necessary (although possible and useful) that the editor runs in its own window. As a simple solution you can use a text terminal and rewrite all of its contents on each change.

Ideally (not mandatory) it shall be possible to start a process executing the edited program just by pressing a button.

In this task it is important to manage data. At a first glance, that seems to be difficult to do in modern functional languages. However, there are several simple ways to do it. Consulting the standard libraries of the used programming system may be helpful to find a good solution.

The Implementation Language

To program in ML we recommend to use OCaml. This system extends ML with object-oriented features. They support a programming style similar to that of Java. Please use only the functional concepts of the language, do not use the object-oriented language extensions.

For programming in Haskell you can select between GHC and Hugs. GHC is a rather large and efficient system. You can download an entire development platform including all tools that you will probably need. Hugs is a simple system based solely on an interpreter. Because of the better performance, GHC is preferable over Hugs for this task.