Third Exercise

Exercise

Develop a syntax-aware editor for programs in the language as developed in exercise 2. The editor shall be written in a statically typed functional language like Haskell and 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 and unterminated strings,
- variables occurring as parts of strings, but not outside of strings (so that they never can become bound),
- variables (not procedure names) occurring outside of strings, but not as parts of strings (so that they are never used).

You can use text attributes like colour, underlining, thickness, inversion and size to highlight the current cursor position, names and other syntactic elements.

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

In this exercise 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 that. Consulting the standard libraries of the used programming system may help to find a good solution.

The Implementation Language

To program in ML we recommend the use of 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 to solve the exercise, don't use the object-oriented language extensions.

Information on Haskell can be found on the Haskell page. Essentially 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 simpler system based solely on an interpreter. Because of the better performance, GHC is preferable over Hugs for the solution of this exercise.