Prof. Dr. Arnd Poetzsch-Heffter
Dr.-Ing. Jan Schäfer

Technische Universität Kaiserslautern
Fachbereich Informatik
AG Softwaretechnik

Übungsblatt 1: Übersetzer und sprachverarbeitende Werkzeuge (SS 2011)

Hand Out: 20. April 2009
Hand In: 27. April 2009

Aufgabe 1   Scanner Implementation

Suppose you would want to implement a calculator which is able to accept input of the following form:

```
3 5 +
4 5 *
4 5* 42 17+ -
```

We will implement a scanner for this calculator, i.e., a software system which splits the input into a list or stream of tokens. The scanners interface will consist of one `scan` method, which converts the input, given as `String`, into a list of tokens, as `List<Token>`. Where a token is either a natural number or one of several operators. Whitespace, i.e., spaces, tabulators, and newlines, are not represented in the result of the scanner.

The scanner maintains a current position where it expects a token to start. The scanner tries to find the longest character sequence which is a valid token and starts at the current position. If it was successful, it updates the current position to the first character after the newly found token and stores the token into the result list. If no token starting at the current position is found, it returns an error.

It is also the responsibility of the scanner to return tokens in an appropriate representation, e.g. sequences of digits should be converted to numbers.

Use the following Java code to represent tokens.

```java
class Token {
  public enum Type { NUMBER, PLUS, MINUS, MULT, DIV; }

  private final int value;
  private final Type tokenType;

  public Token(Type t) {
    this(t, 0);
  }

  public Token(Type t, int value) {
    this.tokenType = t;
    this.value = value;
  }

  public int getNumValue() {
    return value;
  }

  public Type getType() {
    return tokenType;
  }
}
```

Download the file `Ex1.java` from the web page and use it for your solution.

Aufgabe 2   An Extended Scanner

Now, we will extend the input language of our calculator. In addition to integers, floats should be recognized by the calculator.

- Copy your `Ex1.java` to `Ex2.java` and extend the classes `Token.Type` and `Token` to support floats in addition to integers.

- Extend your scanner to accept floats in the following syntax: `3.141`.

Aufgabe 3   Discussion

Discuss the following questions:

- How would you model tokens in ML/Haskell?

- Which concept of theoretical computer science would you use to define symbol classes?

- Which formal problem does a scanner solve?

- Usually scanners also handle comments. Could you extend your scanner to handle comments? Could you extend your scanner to handle nested comments?

Aufgabe 4   (Optional) Implement the Calculator

Complete the calculator so that it can return the value of a given expression. The operators are applied to the numbers from left to right. For example:

`3 5 +`	$= 3 + 5 = 8$
`4 2 -`	$= 4 - 2 = 2$
`6 2 /`	$= 6/2 = 3$
`4 5* 42 17+ -`	$= (4 * 5) - (42 + 17) = -39$