CS321 Languages and Compiler Design I, Winter 2015 1/12/15

Prof. Jingke Li (FAB120-06, lij@pdx.edu), Mon 8:15-10:05 @FAB 171, Lab: Wed 8:15-10:05 @FAB 88-10

The miniJava Token Definitions

The miniJava language's token definitions follow Java's lexical rules in most cases. But for simplification purpose, they may not support the full set of Java features.

- miniJava is case sensitive — upper and lower-case letters are *not* considered equivalent.
- The following are miniJava's *reserved* words — they must be written in the exact form as given:

  ```
  class extends static public void int double
  boolean new this if else while return
  main true false String System out println
  ```

  Note that the words in the third row are not reserved in Java. They are made reserved in miniJava to simplify syntax compatibility with Java. (For example, you can use `System.out.println` to print in miniJava even though miniJava does not support packages.)
- An *identifier* starts with a letter, followed by an optional sequence of letters and/or digits. If such a sequence matches a reserved word, then the sequence is considered a reserved word, not an identifier. There is no limit on the length of the sequence.
- There are three forms of *integer* literals:
  - a *decimal* constant consists of a non-empty sequence of digits, with a non-zero digit at the beginning;
  - an *octal* constant consists of a digit `0` followed by a non-empty sequence of digits `0-7`;
  - a *hexadecimal* constant consists of `0x` or `0X` followed by a non-empty sequence of hexadecimal digits, *i.e.* digits plus letters `a` through `f` (both upper and lower cases allowed).

  An integer literal's value must be in the range 0 to $2^{31}-1$. (Note that the value is always non-negative. A negative integer constant, such as `-3`, is constructed from an unary minus operator and an integer literal.)
- A *floating-point* literal contains a non-empty sequence of digits and a decimal point. The decimal point may appear anywhere in the sequence, *e.g.* `.123`, `12.3`, and `123.` are all valid floating-point literals.
- A *String* literal contains a sequence of ASCII characters (except double quotes (`"`), carriage returns (`\r`), and newlines (`\n`)) delimited between a pair of double quotes (`"`). A string literal can be of arbitrary length, including zero. Note that the beginning and ending double quotes of a string literal are not part of the literal, even though they are included in the lexeme.
- *Comments* can be in two forms: a single-line comment starts with `//` and ends with an (invisible) newline character (`\n`); multi-line comments are enclosed in the pair `/*`, `*/`; they cannot be nested. All ASCII characters are legal in a comment.
- The following are miniJava's *operators* and remaining *delimiters*:

  ```
  operator  = "+"|"-"|"*"|"/"|"&&"|"||"|"!"|"=="|"!="|"<"|"<="|">"|">="
  delimiter = "="|";"|","|"."|"("|")"|"["|"]"|"{"|"}"
  ```