

Aufgabe 8: Scannen von Namen

Implementieren Sie die Methode `readName(t)` gemäß ihrer Spezifikation in Kapitel 2. Die Methode wird von `next` aufgerufen, um einen Namen oder ein Schlüsselwort zu erkennen. Die Tokenart soll in `t.kind` gespeichert werden und der Text des Namens in `t.val`.

Lösung

Beim Aufruf von `readName` enthält `ch` das erste Zeichen des Namens (einen Buchstaben). `readName` muss nun weitere Zeichen, Ziffern und Unterstreichungszeichen lesen und daraus einen Namen aufbauen, der in `t.val` gespeichert wird.

Namen und Schlüsselwörter haben die gleiche lexikalische Struktur. Nachdem ein Name erkannt wurde, muss daher in einer Tabelle nachgesehen werden, ob es sich um ein Schlüsselwort handelt. Wir benutzen dazu binäres Suchen, obwohl man auch eine Hashtabelle benutzen könnte. Dazu deklarieren wir ein sortiertes Array `key` mit den Schlüsselwörtern von MicroJava und parallel dazu ein Array `keyVal` mit den entsprechenden Tokencodes:

```
private static final String key[] =
    { "break", "class", "else", "final", "if", "new", "print", "program", "read", "return", "void", "while" };
private static final int keyVal[] =
    { break_, class_, else_, final_, if_, new_, print_, program_, read_, return_, void_, while_ };
```

Außerdem deklarieren wir ein `char`-Array `lex`, in dem wir die Zeichen des Namens sammeln:

```
private static char[] lex = new char[128];
```

Die Methode `readName` sieht dann wie folgt aus:

```
private static void readName (Token t) {
    int i = 0;
    do {
        lex[i++] = ch; nextCh();
    } while ('a' <= ch && ch <= 'z' || 'A' <= ch && ch <= 'Z' || '0' <= ch && ch <= '9' || ch == '_');
    t.val = new String(lex, 0, i);
    t.kind = checkIfKeyword(t.val);
}
```

Die Methode `checkIfKeyword` sucht den Namen mittels binärem Suchen in der Schlüsselworttabelle `key`. Wenn der Name gefunden wird, wird der entsprechende Schlüsselwortcode zurückgegeben, ansonsten `ident`.

```
private static int checkIfKeyword (String s) {
    int i = 0, j = key.length-1;
    while (i <= j) {
        int k = (i + j) / 2;
        int d = s.compareTo(key[k]);
        if (d < 0) j = k - 1;
        else if (d > 0) i = k + 1;
        else return keyVal[k];
    }
    return ident;
}
```