

## Aufgabe 14.1 (P) Lambdas und GUI

In der Vorlesung haben Sie das Konzept der Listener-Klassen kennengelernt, um Aktionen bei Interaktion mit der Benutzeroberfläche auslösen zu können. Zum Beispiel benötigt es einen ActionListener, welcher die `actionPerformed()`-Methode implementiert, um bei einem Klick auf einen Knopf reagieren zu können:

```
1 java.awt.Button button = new java.awt.Button();
2 button.addActionListener(new ActionListener() {
3     @Override
4     public void actionPerformed(ActionEvent actionEvent) {
5         // nach Klick auf Button wird dieser Code ausgeführt
6     }
7 });
```

In den Musterlösungen der GUI-Aufgaben der vorangegangenen Blätter wurde eine alternative Implementierungsform mittels Lambda-Ausdrücken gezeigt, welche weniger Codezeilen bedarf und somit als *syntaktischer Zucker* angesehen werden kann. Dies soll in dieser Aufgabe anhand eines Beispiels genauer erläutert werden.

Im Wesentlichen besteht die Vereinfachung gegenüber der bekannten Listener-Implementierungsweise darin, dass die zu implementierende Listener-Methode als Lambda-Ausdruck formuliert wird. Dieser wird dann als Listener-Ersatz dem entsprechenden GUI-Element hinzugefügt, welches intern einen Listener mit dieser anonymen Funktion als Listener-Methode anlegt:

```
1 java.awt.Button button = new java.awt.Button();
2 button.addActionListener(actionEvent -> {
3     // nach Klick auf Button wird dieser Code ausgeführt
4 });
```

Um dieses Konzept zu verstehen, verwenden Sie die Musterlösung der Chataufgabe 12.2 (P) und ersetzen Sie alle Lambda-Ausdrücke falls möglich mit entsprechenden Listener-Objekten. Überlegen Sie sich, ob die in der Musterlösung vorhandenen Listener-Objekte durch Lambda-Ausdrücke ersetzt werden können, und wenn bzw. wenn nicht, warum.

## Aufgabe 14.2 (P) Syntaxbaum

Zeichnen Sie für das folgende MiniJava-Programm den Syntaxbaum. Dazu steht Ihnen die vereinfachte Grammatik von MiniJava in Abbildung 1 zur Verfügung.

```

1  int prod, x, n;
2  x = read();
3  if (0 < x) {
4      prod = 1;
5      n = 0;
6      while (prod <= x) {
7          n = n + 1;
8          prod = prod * (-n);
9      }
10     write(prod);
11 } else {
12     write(n);
13 }

```

### Aufgabe 14.3 (P) Altklausur vom letzten Jahr

Auf Moodle finden Sie die Wiederholungsklausur des letzten Jahres. Lösen Sie diese Klausur!

Die Hausaufgabenabgabe erfolgt über Moodle. Geben Sie Ihre Lösungen in einer oder zwei, korrekt orientierten und gut lesbaren PDF-Datei(en) ab. Hausaufgaben, die sich nicht im vorgegebenen Format befinden, werden nur mit Punktabzug oder gar nicht bewertet.

### Aufgabe 14.4 (H) Bäumchen

[5 Punkte]

Zeichnen Sie für das folgende MiniJava-Programm den Syntaxbaum. Dazu steht Ihnen die vereinfachte Grammatik von MiniJava in Abbildung 1 zur Verfügung.

```

1  int a, b;
2  a = 1;
3  while (a == 1) {
4      a = read();
5  }
6  b = read();
7  while (b > a) {
8      if (b > a)
9          b = b - a;
10     else
11         a = b + a;
12 }
13 write(b);

```

```

<program> ::= <decl>* <stmt>*
<decl>    ::= <type> <name> ( , <name> )* ;
<type>    ::= int
           | int[]
<stmt>    ::= <expr>? ;
           | { <stmt>* }
           | <name> = <expr>;
           | <expr> [ <expr> ] = <expr>;
           | if ( <cond> ) <stmt> ( else <stmt> )?
           | while( <cond> ) <stmt>
           | do <stmt> while( <cond> );
           | return <expr>;
<expr>    ::= <number>
           | <name>
           | new int [ <expr> ]
           | <expr> [ <expr> ]
           | length ( <expr> )
           | read ( )
           | write ( <expr> )
           | <name> ( ( ε | <expr>( , <expr>)* ) )
           | ( <expr> )
           | <unop> <expr>
           | <expr> <binop> <expr>
<unop>    ::= -
<binop>   ::= - | + | * | / | %
<cond>    ::= true | false
           | ( <cond> )
           | <expr> <comp> <expr>
           | <bunop> ( <cond> )
           | <cond> <bbinop> <cond>
<comp>    ::= == | != | <= | < | >= | >
<bunop>    ::= !
<bbinop>   ::= && | ||
<name>    ::= [A-Za-z_] [A-Za-z_]*
<number>  ::= 0 | ([1-9] [0-9]*)

```

Abbildung 1: Vereinfachte Grammatik der Sprache MiniJava

### Aufgabe 14.5 (H) Ende gut, alles Pinguin gut!

[2 Bonuspunkte]

Uns ist aufgefallen, dass wir bisher im Praktikum noch keine machbare Aufgabe gestellt haben. In dieser Aufgabe geht es nun darum, einen Pinguin möglichst natürlich und farbenfroh auszumalen. Gehen Sie bei Ihrer Malung wie folgt vor:

1. Beginnen Sie mit dem Kopf des Pinguins.
2. Setzen Sie Ihre Arbeit mit dem Rumpf des Tierchens fort.
3. Runden Sie Ihr Werk durch die Gestaltung der Extremitäten ab.

