

Programmierung A/B (in Java) – Übung 10

Thema: Anonyme Klassen und Lambda-Ausdrücke

Lernziele: In dieser Übung sollen Sie lernen, mit anonymen Klassen und Lambda-Ausdrücken umzugehen

Hinweis: Dieses Übungsblatt ist nur für Studierende relevant, die an Teil B der Veranstaltung teilnehmen. Nachfolgende Übungsblätter werden allerdings wieder Aufgaben enthalten, die auch für Studierende von Teil A von Interesse sein werden.

Aufgabe 1: Verständnis

Beschreiben Sie zunächst mit eigenen Worten, was anonyme Klassen sind und in welchem Kontext sie verwendet werden. Anschließend erklären Sie, wozu Lambda-Ausdrücke verwendet werden und was die notwendige Voraussetzung darstellt, die erst deren kompakte Schreibweise ermöglicht.

Gegeben seien die folgenden beiden Interfaces *IFoo* und *IBar*, sowie die Lambda-Ausdrücke *f1* bis *f5*. Erklären Sie deren Funktion:

```
public interface IFoo {  
    public String foo(String s);  
}
```

```
public interface IBar {  
    public int bar(int x, int y);  
}
```

```
IFoo f1 = x -> x.substring(0, 1).toUpperCase() +  
                x.substring(1).toLowerCase();
```

```
IFoo f2 = x -> x==null ? "null" : x.isEmpty() ? "<empty>" :  
                x.length() + " chars";
```

```
IFoo f3 = x -> {  
    int len = x.length();  
    String y = "";  
    for(int i=0; i<len; i++) y = x.charAt(i) + y;  
    return y;  
};
```

```
IBar f4 = (a,b) -> a < b ? a : b;
```

```
IBar f5 = (f,g) -> {  
    for(f=f4.bar(f,g), g=f-1; g>0; f*=g--);  
    return f;  
};
```

Aufgabe 2: Umwandlung anonyme Klasse <--> Lambda-Ausdruck

Wandeln Sie die Lambda-Ausdrücke der Variablen mit den Bezeichnern f1, f2 und f4 aus Aufgabe 1 in äquivalente Ausdrücke einer anonymen Klasse um. Wandeln Sie anschließend die folgenden Definitionen von anonymen Klassen in äquivalente Lambda-Ausdrücke um:

```
IOps f6 = new IOps() {  
    public double ops(int a, int b, int c) {  
        return Math.acos((a*a + b*b - c*c) / (2.0*a*b));  
    }  
};  
  
IOps f7 = new IOps() {  
    public double ops(int a, int b, int c) {  
        double m = (a+b+c)/3.0;  
        double v = ((m-a)*(m-a)+(m-b)*(m-b)+(m-c)*(m-c))/3.0;  
        return v;  
    }  
};
```

Aufgabe 3: Anwendung

Gegeben sei das von Ihnen in Übungsblatt 6, Aufgabe 2 geschriebene Interface *IArrayOperation* zur Implementierung diverser Arrayoperationen. Warum können für die Initialisierung von Variablen vom Typ *IArrayOperation* zwar anonyme Klassen, jedoch keine Lambda-Ausdrücke verwendet werden?

Ändern Sie das Interface so, dass auch Lambda-Ausdrücke verwendet werden können (Tipp: Für uns von Interesse ist an dieser Stelle nur die Methode *arrayOperation*) und schreiben Sie folgende Operationen als Lambda-Ausdruck:

- Jeder Wert des Arrays in der Diagonalen, beginnend bei Index 0/0, soll auf 0 gesetzt werden.
- Der Wert an jeder Position des Arrays soll auf die Summe der Werte der direkt angrenzenden Felder gesetzt werden (nur horizontal/vertikal, nicht diagonal). Der Wert angrenzender Felder, die über das Array hinausgehen, soll dabei als 0 angenommen werden. Verwenden Sie dazu ein Hilfsarray und nehmen Sie an, dass die Dimensionen des Eingabearrays für alle Zeilen bzw. Spalten gleich ist („rechteckige“ Form).