



Programmierung für Mathematik HS10

Übung 7

1 Aufgabe: Integer Arrays

1.1 Lernziele

1. Arrays repetieren.
2. Verzweigungen und Schleifen repetieren.

1.2 Aufgabenstellung

Schreiben Sie eine Klasse `IntegerArray`, die als Instanzvariable ein Array vom Basistyp `int` besitzt, welches direkt bei der Initialisierung gesetzt werden kann. Schreiben Sie folgende Methoden, wobei die Instanzvariable jeweils nicht verändert werden soll. (Verwenden Sie keine Methoden der Klassen aus der Java-API.):

1. Eine Methode, welche das Array zurückgibt. Es soll aber nicht direkt eine Referenz auf die Instanzvariable übergeben werden, sondern nur eine Referenz auf eine Kopie der Instanzvariable.
2. Eine Methode, welche das sortierte Array zurückgibt
3. Eine Methode

```
public int[] replace(int oldNumber, int newNumber)
```

Diese Methode soll `oldNumber` mit `newNumber` ersetzen, sofern `newNumber` im Array vorkommt.

4. Eine Methode, welche die Absolutbeträge aller Zahlen im Array berechnet und wieder als Array zurückgibt.
5. Eine Methode, welche das Array in umgekehrter Reihenfolge zurückgibt.
6. Eine Methode, welche die i -te Position des Arrays löscht (i soll vom Benutzer eingegeben werden können), das heisst es muss ein Array zurückgegeben werden, welches um eine Position kürzer ist. Falls der Benutzer eine ungültige Position eingibt, soll das ganze Array zurückgegeben werden.

2 Aufgabe: Rekursion

2.1 Lernziele

1. Mathematische Rekursion verstehen.
2. Rekursionen in Java implementieren können.

2.2 Aufgabenstellung

a) Fibonacci

Schreiben Sie eine Methode, welche das n -te Glied der Fibonacci Folge berechnet.

Hinweis: die Fibonacci Folge ist durch folgende Rekursion gegeben:

$$a_0 := 1, \quad a_1 := 1, \quad a_n := a_{n-1} + a_{n-2} \quad \forall n \in \mathbb{N}_{\geq 2}$$

b) Intervallschachtelung

Betrachten Sie folgende Intervallschachtelung und überlegen Sie sich, was damit berechnet wird (dabei soll $x \in \mathbb{R}_{>1}$ beliebig sein).

$[a_n, b_n] \subset \mathbb{R}$ sei Intervall, wobei:

$$[a_0, b_0] := [1, x] \quad [a_{n+1}, b_{n+1}] := \begin{cases} [\frac{a_n+b_n}{2}, b_n] & \text{falls } (\frac{a_n+b_n}{2})^2 \leq x \\ [a_n, \frac{a_n+b_n}{2}] & \text{falls } (\frac{a_n+b_n}{2})^2 > x \end{cases}$$

Implementieren Sie diese Rekursion, wobei x vom Benutzer eingegeben werden soll. Überlegen Sie sich eine geeignete Abbruchbedingung. Der Rückgabewert soll der Mittelpunkt des zuletzt berechneten Intervalls sein.

3 Aufgabe: Polynom Interpolation

3.1 Lernziele

1. Rekursive Methoden schreiben können.
2. Vorhandene Programme sinnvoll verwenden können.

3.2 Aufgabenstellung

1. Schreiben Sie eine rekursive Methode um die Potenz x^n zu berechnen, wobei $x \in \mathbb{R}$ und $n \in \mathbb{N}$.
2. Schreiben Sie eine Klasse `Point`, welche einen Punkt im \mathbb{R}^2 darstellen soll.
3. Schreiben Sie ein Programm, mit welchem n gegebene Punkte durch ein Polynom vom Grad $n-1$ interpoliert werden. Das heisst, der Benutzer soll n Punkte $(x_1, y_1), \dots, (x_n, y_n)$ mit $x_i \neq x_j$ für $i \neq j$ eingeben und dann soll von Ihrer Methode ein Polynom $p(x)$ zurückgegeben werden, sodass $p(x_i) = y_i \forall i \in \{1, \dots, n\}$.

(Verwenden Sie die Klasse `Polynomial` von Übung 4 um das Polynom abzuspeichern.)

Hinweis: Es müssen n Koeffizienten berechnet werden:

$$p(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$$

dies führt zu folgendem linearen Gleichungssystem (welches eine eindeutige Lösung besitzt):

$$\begin{aligned} a_0 + a_1 \cdot x_1 + \dots + a_{n-1} \cdot (x_1)^{n-1} &= y_1 \\ a_0 + a_1 \cdot x_2 + \dots + a_{n-1} \cdot (x_2)^{n-1} &= y_2 \\ &\vdots \\ a_0 + a_1 \cdot x_n + \dots + a_{n-1} \cdot (x_n)^{n-1} &= y_n \end{aligned}$$

4. Schreiben Sie einen `TestDriver` um Ihre Methoden zu testen.

4 Aufgabe: Interface

4.1 Lernziele

1. Sie können eine gemeinsame Schnittstelle schreiben und andere Klassen diese implementieren lassen.
2. Sie verstehen anhand einem ersten Beispiel die Vorteile einer gemeinsamen Schnittstelle.

4.2 Aufgabenstellung

a) Geometrische Figuren

Modellieren Sie die folgenden geometrischen Figuren jeweils in einer eigenen Klasse. Die Implementierung soll dabei nur so detailliert sein, dass für jede Figur das Volumen abgefragt werden kann.

- Quader
- Reguläre Pyramide
- Gerader Kreiszylinder
- Kugel

Die Eingaben des Benutzer sollen dabei jeweils überprüft werden und allenfalls eine Exception geworfen werden (zum Beispiel wenn negative Seitenlängen eingegeben werden).

Falls Ihnen die einzelnen Voluminaberechnungen nicht geläufig sind, hilft [diese Auflistung](#) weiter.

b) Volumina

Erweitern Sie die Klassen aus a) dahingehend, dass es möglich ist, unterschiedliche Instanzen in einem geeigneten Array abzulegen und in einer Schleife alle Volumina auszugeben. Schreiben Sie einen `TestDriver` welche die Funktionalität Ihres Programmes testet, wobei die verschiedenen Instanzen in einer `ArrayList` abgespeichert werden sollen.