

Aufgabe 9.1

Schreiben Sie eine Klasse `MyComplexNumber.java`, die eine komplexe Zahl modellieren soll. Der Realteil der Zahl soll den Namen `rValue`, der Imaginärteil den Namen `iValue` erhalten. Implementieren Sie folgende Methoden:

Addition `MyComplexNumber add(MyComplexNumber a, MyComplexNumber b)`
Subtraktion `MyComplexNumber sub(MyComplexNumber a, MyComplexNumber b)`
Multiplikation `MyComplexNumber mult(MyComplexNumber a, MyComplexNumber b)`

Es seien $c_1 = a + bi$ und $c_2 = c + di$ zwei komplexe Zahlen, dann gilt:

$$\begin{aligned}c_1 + c_2 &= (a + c) + (b + d)i \\c_1 - c_2 &= (a - c) + (b - d)i \\c_1 \cdot c_2 &= (ac - bd) + (ad + bc)i\end{aligned}$$

(8 Punkte)

Aufgabe 9.2

Erstellen Sie eine abstrakte Klasse `Function.java`, die eine Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ modellieren soll. Die Klasse soll die folgenden drei abstrakten Methoden enthalten:

- `public abstract Function ableitung();`
Diese berechnet die Ableitung der Funktion und gibt diese als Objekt vom Typ `Function` zurück.
- `public abstract double integrate(double a, double b);`
Diese berechnet das Integral über die Funktion in den Grenzen von a bis b . Das Ergebnis ist eine reelle Zahl, die zurückgegeben wird. Dabei wird $a \leq b$ vorausgesetzt.
- `public abstract String toString();`
Diese gibt eine Beschreibung der Funktion als String zurück.

Als konkrete Typen von Funktionen wollen wir Polynome und trigonometrische Funktionen betrachten. Leiten Sie dazu eine Klasse `Polynom.java` aus der Klasse `Function.java` ab. Diese Klasse soll die drei obigen abstrakten Methoden in einer für Polynome geeigneten Art und Weise implementieren. Wir gehen davon aus, dass das Polynom als Summe von Termen der Form ax^k vorliegt, dabei ist x das reellwertige Argument der Funktion, a eine reellwertige Konstante und k eine natürliche Zahl. Ein Beispiel für ein Polynom wäre etwa:

$$2x^2 + 6x^3 - 12x^1 + 7x^5 + 4x^0$$

Überlegen Sie sich zuerst, wie Sie ein Polynom intern repräsentieren wollen. Der Konstruktor der Klasse `Polynom.java` bekommt die Beschreibung des Polynoms als String:

```
public Polynom(String p)
```

Diese muss dann erst einmal in die von Ihnen gewählte interne Repräsentation übertragen werden. Sie können die Methode `replace` der Klasse `String` nutzen, um den String zu vereinfachen. Nutzen Sie die Methode `split`, um den String anhand aller `+`-Zeichen aufzuteilen. Zur Nutzung der Methoden lesen Sie in der Online Dokumentation zu Klasse `String`. Trennen Sie danach Koeffizient, Variable und Exponent.

Schreiben Sie ein Programm, welches als Kommandozeilenparameter folgende Informationen erhält:

- Den Namen einer Datei, welche in jeder Zeile die Beschreibung einer Funktion als String enthält. Ein Beispiel für eine solche Datei ist im Material zu diesem Aufgabenblatt enthalten.
- Die Anweisung **ableiten**. Dann soll das Programm für jede Funktion in der Eingabedatei die Ableitung berechnen und diese dann zeilenweise in eine Ausgabedatei schreiben. Der Name der Ausgabedatei wird vom Nutzer als letzter Kommandozeilenparameter angegeben.
- Die Anweisung **integrieren**. Dann soll das Programm für jede Funktion in der Eingabedatei das Integral berechnen. Die Grenzen a und b werden vom Nutzer auch als Kommandozeilenparameter vorgegeben und sind für alle Funktionen in der Eingabedatei dieselben. In die Ausgabedatei soll dann zeilenweise immer die Funktion und der Wert des Integrals geschrieben werden.

(8 Punkte)