

Übungsblatt 4 zur Vorlesung Programmieren

Abgabe: Do 02.11.2023 – Abgabe: Do 16.11.2023

Hinweise (diese gelten auch für zukünftige Übungen):

- Bitte beachten Sie, dass auch weiterhin neben der Abgabe in ILIAS eine Präsentation im Tutorium erforderlich ist. Dabei sollten alle Teammitglieder anwesend sein.
 - Sie sollten, wenn möglich, die Aufgaben bereits vor dem Tutorium durchgelesen und sich mit der Aufgabenstellung vertraut gemacht haben.
 - Fragen zu den Aufgaben können Sie gerne auch über MatterMost stellen (<https://mattermost.hska-iwi.de/i-prg1>).
-

Aufgabe 1 (Komplexe Zahlen):

Die *komplexen Zahlen* sind eine Erweiterung der reellen Zahlen, so dass auch die Gleichung $x^2 = -1$ eine Lösung besitzt. Die Lösung dieser Gleichung wird mit dem Symbol i bezeichnet, d.h. es gilt $i^2 = -1$ für die komplexe Zahl i , die auch *imaginäre Einheit* genannt wird.

Komplexe Zahlen z haben die Form $z = a + bi$, wobei $a, b \in \mathbb{R}$. D.h. jede komplexe Zahl lässt sich durch ein Paar (a, b) von reellen Zahlen repräsentieren, wobei a auch der Realteil und b der Imaginärteil der Zahl z genannt wird.

Auf den reellen Zahlen lassen sich die üblichen arithmetischen Operationen wie Addition, Subtraktion, Multiplikation und Division definieren. Dies geschieht wie folgt, wobei wir annehmen, dass $z_1 = a + bi$ und $z_2 = c + di$.

- $z_1 + z_2 = (a + bi) + (c + di) = (a + c) + (b + d)i$, d.h. die Summe von z_1 und z_2 ist die komplexe Zahl mit Realteil $a + c$ und Imaginärteil $b + d$.
- $z_1 - z_2 = (a - c) + (b - d)i$
- $z_1 \cdot z_2 = (ac - bd) + (ad + bc)i$
- $\frac{z_1}{z_2} = \frac{ac+bd}{c^2+d^2} + \frac{bc-ad}{c^2+d^2}i$

Erstellen Sie eine Klasse `Complex` zur Repräsentation von komplexen Zahlen in einer Datei `Complex.java` mit folgender Funktionalität:

- a) Realteil und Imaginärteil sollen als `double`-Werte abgespeichert werden.
- b) Die Klasse soll zwei Konstruktoren besitzen, einen mit zwei `double`-Argumenten und einen mit einer leeren Argumentliste.
- c) Die Klasse soll Methoden für die vier Grundrechenarten auf komplexen Zahlen enthalten:

```
Complex add(Complex other) { ... }  
Complex sub(Complex other) { ... }  
Complex mul(Complex other) { ... }  
Complex div(Complex other) { ... }
```

Jede Methode soll die entsprechende Verknüpfung von 'this' und 'other' durchführen und das Ergebnis als neue komplexe Zahl zurückgeben. 'this' bleibt dabei unverändert.

- d) Die Klasse soll eine Methode `public String toString()` enthalten, die die textuelle Darstellung einer komplexen Zahl berechnet. Wenn der Imaginärteil 0 ist, soll nur der Realteil ausgegeben werden.
- e) Die Klasse soll eine Methode `public boolean equals(Complex other)` enthalten, um zwei komplexe Zahlen auf Gleichheit zu prüfen.

Erstellen Sie des Weiteren eine Klasse `ComplexTest` (u.a. mit einer `main`-Methode), mit der Sie Ihre Implementierung testen.

- f) Erstellen Sie vier Testfälle, je einen für `add`, `sub`, `mul` und `div`. Überlegen Sie sich jeweils passende Operanden, die zu nicht-trivialen Berechnungen führen (z.B. wäre $0 + 0$ trivial).

- g) Erstellen Sie einen weiteren Testfall für die `mul`-Operation, der nachweist, dass $i^2 = -1$ gilt.

Die Ausgabe eines jeden Testfalls soll dabei einerseits angeben, welche Operation mit welchen Operanden durchgeführt wurde, und andererseits, ob das Ergebnis mit dem erwarteten Wert übereinstimmt. Ist dies der Fall, so soll "Ok" ausgegeben werden, ansonsten "FAILED: expected X, got Y", wobei X und Y das erwartete und das berechnete Ergebnis sind. Der Wert für das erwartete Ergebnis soll fest in Ihrem Programm hinterlegt sein.

Aufgabe 2 (Tic-Tac-Toe):

Tic-Tac-Toe ist ein einfaches Strategiespiel für zwei Personen (siehe z.B. <https://de.wikipedia.org/wiki/Tic-Tac-Toe>).

In dieser Aufgabe soll das Spiel Tic-Tac-Toe implementiert werden, so dass zwei menschliche Gegner am Computer gegeneinander spielen können. Die Eingabe der Züge sowie die Ausgabe des aktuellen Spielstandes soll auf der Konsole erfolgen. Gehen Sie wie folgt vor:

- a) Überlegen Sie sich Datenstrukturen zur Darstellung des Spielfeldes und ggf. der Spieler.
- b) Entwickeln Sie eine Methode zur Ausgabe des aktuellen Stands des Spiels auf der Konsole.
- c) Schreiben Sie eine Methode, die es erlaubt, einen Spielstein für einen Spieler auf ein leeres Feld zu setzen. Die Methode soll den Rückgabetypp `boolean` besitzen und `true` genau dann zurückgeben, wenn das Feld frei war und der Stein gesetzt werden konnte.
- d) Erstellen Sie eine Methode, die feststellt, ob das Spiel für einen Spieler gewonnen ist.
- e) Fügen Sie eine Spiellogik hinzu, die Spielzüge der beiden Spieler abwechselnd entgegennimmt so lange, bis das Spiel für einen Spieler gewonnen wurde. Das Ergebnis des Spiels soll anschließend ausgegeben werden.