University of Zurich[UZH]

s.e.a.l. software evolution & architecture lab

Programmierung für Mathematik HS11

Übung 3

1 Aufgabe: Online Modul

1.1 Lernziele

1. Die verschiedenen Kontrollstrukturen verstehen.

1.2 Aufgabenstellung

Arbeiten Sie das *Self-Study Module 3* im OLAT durch.

2 Aufgabe: Berechnungen zu Polynomen 2. Grades

2.1 Lernziele

1. Selbstständig Lösungen mathematischer Problemstellungen finden und in Java implementieren können.
2. Verzweigungen (`if-else`-Statements) programmieren können.

2.2 Aufgabenstellung

a) Polynom zweiten Grades

Erstellen Sie eine Klasse `Poly2`, welche ein Polynom vom Grad 2 darstellen soll. Ein Polynom zweiten Grades ist durch die Koeffizienten a, b und c eindeutig bestimmt (wobei $p(x) = ax^2 + bx + c$). Die Klasse `Poly2` muss also drei Instanzvariabeln vom Typ `double` enthalten. Schreiben Sie dazu noch passende `getter`- und `setter`-Methoden.

Tipp: Betrachten Sie die Klasse `ComplexNumber` der letzten Übung, `Poly2` sieht sehr ähnlich aus.

b) Nullstellen des Polynoms

Ergänzen Sie `Poly2` mit der Methode `printRoots`, welche die reellen Nullstellen des Polynoms berechnen und auf die Kommandozeile ausgeben soll. (Beachten Sie, dass es Polynome zweiten Grades gibt, die nur eine bzw. keine reellen Nullstellen besitzen.)

Tipp: Benutzen Sie wieder die Methode `sqrt` aus der Klasse `Math`, um die Quadratwurzel einer Zahl zu berechnen. (Vgl. Übung 2)

3 Aufgabe: Fakultät

3.1 Lernziele

1. Die `for`-Schleife implementieren können.
2. Das `switch`-Statement implementieren können.

3.2 Aufgabenstellung

a) Fakultät berechnen

Schreiben Sie eine Klasse `Factorial`, welche eine Instanzvariable `number` vom Typ `int` besitzt und schreiben Sie eine geeignete `setter`-Methode. Fügen Sie der Klasse danach eine Methode `calculateFactorial` hinzu, welche mithilfe einer `for`-Schleife die Fakultät einer natürlichen Zahl (ohne 0) berechnet und diese dann als Rückgabewert vom Typ `long` zurück gibt. Die Methode soll nur innerhalb der Klasse nutzbar sein, weshalb man davon ausgehen kann, dass ihr nur natürliche Zahlen als Parameter übergeben werden. Schreiben Sie als letztes eine Methode `getFactorial`. Diese Methode soll als Parameter eine ganze Zahl erhalten und danach mithilfe eines `switch`-Statements folgende Fälle gesondert abhandeln:

- Falls die als Parameter übergebene Zahl grösser als 0 ist, so soll mithilfe der Methode `calculateFactorial` die Fakultät der Zahl berechnet werden.
- Für den Fall, dass eine 0 übergeben wird, so soll direkt die Fakultät von 0, also 1 zurückgegeben werden.
- Bei einer negativen Zahl soll die Methode einen Warnhinweis auf der Konsole ausgeben, dass die Fakultät für negative Zahlen nicht definiert ist und den Wert 0 zurück geben.

Benutzen Sie für die Fallunterscheidung die bereits aus Übung 1 bekannte `signum`-Methode aus der Klasse `Math`.

b) TestDriver

Testen Sie ihre `Factorial`-Klasse mittels eines geeigneten TestDrivers. Beachten Sie, dass trotz des verwendeten Datentyps `long` für das Ergebnis, die Fakultät einer Zahl sehr schnell einen Overflow erzeugen kann.