

Programmierung für Mathematik HS12

Übung 7

3. November 2012

1 Aufgabe: Konstruktoren

1.1 Ziele

- Konstruktoren verstehen

1.2 Aufgabenstellung

In den vergangen Übungen haben Sie vor allem getter und setter Methoden verwendet um den Instanzvariablen Ihrer Klassen Werte zuzuweisen. In dieser Aufgabe sollen Sie Klassen mit Hilfe von Konstruktoren so erweitern, dass Sie Instanzvariablen gleich bei der Instanzierung Werte zuweisen können.

Erweitern Sie die folgenden Klassen der vergangenen Übungen:

- `Polynomials` : In dieser Klasse sollen zwei Konstruktoren definiert werden. Ein Konstruktor soll keine Parameter erwarten und das interne Array mit 0 initialisieren. Der zweite Konstruktor soll wie die `set` Methode funktionieren.
- `Matrix` : Diese Klasse soll ein Konstruktor haben, der ein `int[][]` Array erwartet und wie die `set` Methode funktioniert.
- `MyList` : Die Klasse `MyList` soll zwei Konstruktoren haben. Einer der Konstruktoren soll wie die `set` Methode funktionieren, der zweite Konstruktor soll keinen Parameter erwarten und die `int length` Variable auf 0 setzen. Das interne Array soll aber auf ein Array der Länge 10 gesetzt werden.

Zusatz: Erweitern Sie die `MyList` Klasse um die Variablen `int id` und `int lastId`. `int lastId` soll zu Beginn auf 0 gesetzt sein. Bei jedem Aufruf eines Konstruktors der Klasse soll der Wert dieser Variable um 1 erhöht werden und der entsprechenden `int id` zugewiesen werden. `int lastId` ist also eine statische Variable. Passen Sie beide Konstruktoren der Klasse `MyList` dementsprechend an.

2 Aufgabe: Newton Verfahren

2.1 Ziele

- Mathematische Verfahren implementieren

2.2 Aufgabenstellung

In dieser Aufgabe sollen Sie die schon erstellte Klasse `Polynomials` um eine Methode erweitern die folgende Signatur hat:

```
public float root(float start, float epsilon, int maxSteps)
```

Diese Methode soll anhand des Newton-Verfahrens eine Wurzel(nullstelle) des Polynoms ermitteln. Dabei wird eine Genauigkeit von `epsilon` verlangt, da nicht erwartet werden kann, dass genau 0 gefunden wird. Schliesslich sollen nicht mehr als `maxSteps` erfolgen. Falls die geforderte Genauigkeit `epsilon` nicht vor `maxSteps` erreicht wird soll eine Fehlermeldung auf der Konsole erscheinen.

Das Newton-Verfahren: dieses Verfahren definiert eine Folge wobei das erste Glied x_0 gegeben ist, in diesem Fall `start`. Die weiteren Glieder sind durch folgende Zuordnung definiert

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

wobei f das Polynom ist.

3 Aufgabe: Connect Four

3.1 Ziele

- Arrays vertiefen
- Komplexe Programme schreiben

3.2 Aufgabenstellung

In dieser Übung sollen Sie das Spiel Connect Four programmieren. Dieses Programm soll aus zwei Klassen bestehen. Eine Klasse repräsentiert das Spielbrett und die andere Klasse soll einen Spieler implementieren. Nennen Sie die Spielbrett Klasse `ConnectFour` . Diese Klasse soll einen default Konstruktor haben und folgende Methoden implementieren:

- `public void print()` die den momentanen Spielstand in der Konsole ausgeben soll.
- `public boolean play(int column, Player p)` die in der Spalte `column` eine Marke oben hinzufügt. Dabei entscheidet die Variable `p` welcher Spieler an der Reihe ist. Falls nicht schon zuviele Marken in dieser Spalte sind wird `true` returniert sonst `false` .
- `public int get(int colum, int row)` die eine Zahl zwischen 0 und 2 zurückliefert. 0, falls an dieser Stelle noch keine Marke ist, 1 falls eine Marke des ersten Spielers vorhanden ist und 2 für die Marke des zweiten Spielers.
- `public boolean winner()` die `false` zurückliefert falls noch kein Spieler gewonnen hat und `true` anderenfalls. Falls bereits ein Spieler gewonnen hat soll noch eine Ausgabe in der Konsole den Sieger ermitteln. Im Fall, dass das Spielbrett voll ist und kein Spieler gewonnen hat wird trotzdem `true` returniert und eine entsprechende Meldung ausgegeben.

Schreiben Sie die Klasse `Player` die in ihrem Konstruktor einen `boolean` erwartet. Dieser soll intern gespeichert werden und besagt ob dieser Spieler der erste oder zweite Spieler ist. Diese Klasse soll noch folgende Methode haben:

- `public int play(ConnectFour brett)` die anhand der Methoden der `ConnectFour` Klasse entscheidet in welcher Spalte ihre nächste Marke gelegt werden soll.
- `public boolean isFirst()` dabei soll der Wert `true` für den ersten Spieler und `false` für den zweiten stehen.

Schreiben Sie einen TestDriver der zwei Spieler und ein Spielbrett instanziiert und ein ganzes Spiel simuliert bis ein Spieler gewonnen hat.