

---

**Programmierstarthilfe WS 2008/09**  
**Fakultät für Ingenieurwissenschaften und Informatik**

**10. Blatt**

Für die Weihnachtszeit vom 22.12. bis zum 9.1.2009

---

## Organisatorisches

Die Webseiten zur Veranstaltung sind unter <http://www.uni-ulm.de/in/mi/lehre/2008ws/programmierstarthilfe.html> zu finden.

Dieses Mal behandeln wir keinen neuen Stoff sondern bieten dir ein Bonusblatt mit einem kleinen abgeschlossenen Projekt zum Üben des bisherigen Stoffs an.  
Wir wünschen dir ein frohes Weihnachtsfest und einen guten Rutsch ins neue Jahr!

## 1 Bonusaufgabe

### 1.1 Wettbüro

Schreibe ein Programm für ein Online-Wettbüro, das Fußballwetten annimmt. Dabei gibt es pro Spiel verschiedene Wettmodi: Man kann auf den Sieg einer Mannschaft (bzw. ein Unentschieden) setzen oder zusätzlich das Ergebnis tippen. Nach dem Spiel werden die gesetzten Beträge nach einer vorher gesetzten Quote (jeweils für Sieg, Niederlage, Unentschieden, richtig getipptes Ergebnis und richtiger Tordifferenz) ausgezahlt.

Z.B. kann für das Spiel Litauen gegen Deutschland die Quote für den Sieg der Deutschen bei 1.2 liegen, für den Sieg der Litauer bei 20.3, für ein Unentschieden bei 15.0. Ein Tipper, der 10 Euro auf Litauen gesetzt hat, bekommt bei einem Sieg der Litauer  $20.3 * 10 = 203$  Euro ausgezahlt. Für ein richtiges getipptes Ergebnis kann es eine Quote von 10.0 geben, für die richtige Tordifferenz von 5.0. Der entsprechende Betrag wird zusätzlich ausgezahlt.

- a) Lege zunächst eine Klasse `Wette` an. In dieser Klasse wird der Wettmodus z.B. als Integer abgespeichert, außerdem wird der Name des Tippenden, der gesetzte Betrag und der Tipp abgespeichert. Der Tipp sieht folgendermaßen aus: Im ersten Wettmodus ist es ein Integer  $\in \{0, 1, 2\}$  für Unentschieden oder den Sieg der Mannschaft 1 oder 2, im zweiten Wettmodus zwei Integer mit den entsprechenden Toranzahlen. Stelle zwei entsprechende Konstruktoren zur Verfügung. Wird nicht das Ergebnis getippt, sollen die entsprechenden Variablen auf  $-1$  gesetzt werden.
- b) Lege eine Klasse `Spiel` an, deren Objekte jeweils für ein Spiel die Wetten entgegennehmen. Im Konstruktor sollen die Quoten und die Namen der beiden Mannschaften übergeben werden. Außerdem soll es ein Array für Objekte der Klasse `Wette` geben.
- c) Erweitere nun die Klasse `Spiel` um zwei Methoden `tipp`. Einer wird entsprechend dem ersten Wettmodus nur ein Integer übergeben, der anderen zwei. Außerdem werden der Name des Tippenden und der gesetzte Betrag übergeben. In der Methode soll ein Objekt der Klasse `Wette` angelegt und im Array abgespeichert werden.

- d) Implementieren in der Klasse `Spiel` nun noch die Methode `auszahlung`, der das Endergebnis des Spiels übergeben wird. In der Methode soll das `Wettarray` durchlaufen und entsprechend der Quoten für jeden Tippenden der gewonnene Betrag ausgegeben werden.
- e) Teste dein Programm, indem du ein oder zwei Objekte der Klasse `Wette` anlegst, jeweils einige Tipps eingibst und dann die Methode `auszahlung` aufrufst.

## 1.2 Waffenfabrik

Du sollst für ein Strategiespiel eine Waffenfabrik programmieren.

- a) Erstmal benötigen wir Einheiten, die von der Waffenfabrik produziert werden können. Der Einfachheit halber soll es nur eine Einheit geben. Diese soll 100 Lebenspunkte und 25 Waffenstärke haben, wenn sie von einer Waffenfabrik der Ausbaustufe 1 produziert wird. Bei Ausbaustufe 2 soll die Einheit doppelt so viel Lebenspunkte und Waffenstärke haben, bei Stufe 3 dreimal so viel.
- b) Ändere die Einheitenklasse so ab, dass das Produkt aus Lebenspunkte und Waffenstärke gleich bleibt, das Verhältnis aber beim Anlegen der Objekts bestimmt werden kann. So können zum Beispiel auf Ausbaustufe 1  $100 * 25 = 2500$  Punkte verteilt werden. Das Verhältnis von Waffenstärke zu Lebenspunkte bestimmt dann die Aufteilung, zum Beispiel  $1:4 \Rightarrow 25$  Waffenstärke, 100 Lebenspunkte.
- c) Zum Schluss muss die Einheit noch getroffen werden können. überlege dir wie sich das realisieren lässt. Stelle deine Idee kurz einem Tutor vor, bevor du dich an die Implementierung machst. überlege auch, was passieren soll, wenn die Einheit stirbt (alle Lebenspunkte verbraucht).
- d) Nun können wir die Waffenfabrik erstellen. Diese soll 40.000 Lebenspunkte und eine Ausbaustufe zwischen 1 und 3 haben. Außerdem benötigt sie Methoden zur Produktion von Einheiten, um die Ausbaustufe zu erhöhen und um getroffen werden zu können. Die Methode zu Einheitenproduktion soll ein Einheiten-Objekt erstellen und dies an den Aufrufer zurückgeben.
- e) Kommentiere dein Programm so, dass ein Dritter es versteht. Insbesondere solltest du darauf achten, dass verständlich ist, wie Methoden benutzt werden sollen, und was sie zurück geben. Optional: Schau dir an, wie Javadoc funktioniert und kommentiere deinen Code entsprechend

## 1.3 Strategiespiel

Du sollst ein einfaches Strategiespiel programmieren. Das Spiel soll für zwei Spieler sein und in Runden gespielt werden. Jede Runde kann jeder Spieler entscheiden, wieviele Einheiten er bauen will oder ob er die Ausbaustufe seiner Waffenfabrik erhöhen will. Jeder Spieler hat von Anfang an eine Waffenfabrik, und kann keine weitere bauen. Das Spiel ist gewonnen, wenn die Waffenfabrik des Gegner zerstört ist.

Kommentiere dein Programm so, dass ein Dritter versteht was du machst.

- a) Schreibe eine Klasse `Spieler` die für jeden Spieler eine Waffenfabrik und ein Array für maximal 256 Einheiten anlegt. Benutze dafür die Klassen der Aufgabe "Waffenfabrik". Außerdem soll beiden Spielern ein Konto mit 1000 Geldeinheiten angelegt werden. Da keiner

der Spieler ins Minus darf braucht deine Klasse eine Methode `boolean abheben(int betrag)`, die den Betrag vom Konto abzieht und `true` zurückgibt, wenn dies möglich ist, ansonsten wird nichts abgezogen und `false` zurückgegeben. Es wird sich später als nützlich erweisen, wenn der Spieler noch eine Methode anbietet, die zurückgibt ob er noch lebt (also ob seine Waffenfabrik noch Lebenspunkte hat).

- b) Füge `Spieler` eine Methode hinzu, die ausgibt, wieviel Geld und wieviele Einheiten jeder Spieler noch hat und wieviele Lebenspunkte die jeweilige Waffenfabrik noch hat. Anschließend sollen die Spieler eingeben können, wieviele Einheiten sie mit welchem Lebenspunkte/Waffenstärke-Verhältnis bauen wollen, oder ob sie ihre Waffenfabrik ausbauen wollen. In einer Runde können nur entweder Einheiten produziert, oder die Waffenfabrik ausgebaut werden, nicht beides. Die Produktion einer Einheit kostet  $100 \cdot \text{Ausbaustufe}$  Geldeinheiten. Der Ausbau der Waffenfabrik kostet 1000 Geldeinheiten.
- c) Erstelle eine Klasse `Game`. Damit die Einheiten gegeneinander kämpfen können braucht `Game` noch eine Methode `static void fight(Einheit angreifer, Spieler gegner)`. Multipliziere die Waffenstärke des Angreifers mit einem Zufallswert zwischen 0 und 1. Dies ist der Schaden, den die Einheit in dieser Runde anrichtet. Solange noch Einheiten des Gegners existieren sollen diese mit dem Schaden belegt werden. Sind keine Einheiten mehr da wird der Schaden auf die Waffenfabrik gerechnet.
- d) Schließlich brauchen wir noch die `main`-Methode. Diese soll zu Beginn jeder Runde den Spielern 100 Geldeinheiten gutschreiben. Anschließend sollen die Spieler ihre Eingaben machen können. Wähle nun abwechselnd die angreifenden Einheiten der Spieler aus (Jede Einheit sollte nur einmal pro Runde ausgewählt werden) und lasse sie gegen den Gegner kämpfen. Dann beginnt die nächste Runde, sofern beide Waffenfabriken noch stehen. Ansonsten soll natürlich ausgegeben werden, welcher Spieler gewonnen hat.

## 2 Für das nächste Blatt

Nach Weihnachten werden wir Listen behandeln. Lies dazu aus dem Kapitel *Einfache abstrakte Datentypen* den Teil über Listen im Skript.