

Aufgabe 13

Gegeben ist ein Kartenspiel $K = (A, k)$ bestehend aus einer Menge A von Augenzahlen und einer natürlichen Zahl k , die angibt, wie oft jede der Augenzahlen im Spiel vorkommt. Zum Beispiel könnte man das oft benutzte Blatt mit vier Farben zu je 8 Karten beschreiben durch das Paar:

$$(\{2, 3, 4, 7, 8, 9, 10, 11\}, 4)$$

Außerdem ist noch eine natürliche Zahl z gegeben: Der Zielwert, dem man möglichst nahe kommen soll, den man aber nicht überschreiten darf.

In Anlehnung an das Kartenspiel “17 und 4” soll nach folgenden Regeln gespielt werden.

- Es treten immer zwei Spieler gegen einander an.
- Die Spieler sind immer wechselseitig dran.
- Wenn ein Spieler dran ist, bekommt er eine Karte, die zufällig aus den noch verbliebenen ausgewählt wurde.
- Jeder Spieler ist mindestens ein Mal dran.
- Nachdem ein Spieler dran war, muss dieser sofort entscheiden, ob er noch ein weiteres Mal dran kommen möchte, also noch eine weitere Karte bekommen möchte.
- Hat ein Spieler einmal die nächste Karte abgelehnt, dann bekommt er keine weiteren Karten. Der Gegner kann dann aber noch weitere Karten ziehen, bis auch er das erste Mal ablehnt.
- Das Spiel wird abgebrochen, sobald bei mindestens einem der Spieler die Summe der gezogenen Augenzahlen echt größer als z ist.
- Ein Spieler kann nur dadurch gewinnen, dass die Summe seiner gezogenen Augenzahlen kleiner oder gleich z ist und echt größer als die Summe der gezogenen Augenzahlen des Gegners ist.

Im beiliegenden Material ist schon eine Vaterklasse `Player.java` enthalten. Aus dieser sollen Sie eine eigene Klasse `PlayerTollerGruppenname.java` ableiten und die Methode

```
public boolean oneMore(int value)
```

überschreiben, welche beim Spielen aufgerufen wird. Ein origineller und einzigartiger Gruppenname ist unerlässlich. Die übergebene ganze Zahl `value` ist die gezogene Augenzahl. Mit dem Rückgabewert vom Typ `boolean` können Sie dann angeben, ob Sie noch eine weitere Karte haben wollen.

Außerdem sollten Sie den Konstruktor

```
public Player(int[] multiset, int target)
```

anpassen. Das Array `multiset` beschreibt dabei die im Kartenspiel auftretenden Augenzahlen. Der Einfachheit halber enthält dieses Array jede Augenzahl so oft, wie diese vorkommt. In unserem Beispiel oben gibt es also vier Arraypositionen, an denen der Wert 2 steht, ebenso vier Arraypositionen, an denen der Wert 3 steht, usw. Die ganze Zahl `target` ist der Zielwert z , der nicht überschritten werden darf.

Wir werden insgesamt drei Turniere veranstalten, eins für jedes der folgenden Kartenspiele und Zielwerte:

- $(\{2, 3, 4, 7, 8, 9, 10, 11\}, 4)$ mit Zielwert 22
- $(\{1, 2, 3, 4, 5\}, 8)$ mit Zielwert 20
- $(\{1, 2, 3, 4, 17, 18, 19, 20\}, 5)$ mit Zielwert 40

Beim Turnier für ein Kartenspiel tritt jede Gruppe gegen jede andere Gruppe einmal an. Dabei werden tausend Spiele zwischen den beiden Gruppen gespielt. Jede der beiden Gruppen ist dabei abwechselnd zuerst am Zug. Am Ende wird zusammengezählt, wie oft jede Gruppe insgesamt in dem Turnier den Sieg davon getragen hat.

Jede Gruppe, die am Turnier teilnehmen möchte, schickt die Datei mit der aus `Player` abgeleiteten Klasse per e-Mail an den Übungsleiter. Bitte vermerken Sie in Ihrer Mail auch genau, wer alles zu Ihrer Gruppe gehört. Jedes Gruppenmitglied einer Gruppe, die einen lauffähigen Spieler ins Rennen schickt, bekommt dafür

$$\lceil f(n) + h(n, t) \rceil$$

Punkte, wobei n die Anzahl der Mitglieder der Gruppe ist, $f(n) = 4 \frac{n+1}{n}$ die Punktzahl für die Abgabe der Aufgabe ist und $h(n, t) = t \frac{n+1}{n}$ von der Anzahl der gewonnenen Turniere t abhängt.