Übung 1: Umrechnungstabelle (Wiederholung)

Als Wiederholungsaufgabe der Themen letzter Woche sollen Sie eine **Umrechnungstabelle für Celsius- und Fahrenheitwerte** programmieren! Die Schwierigkeit liegt darin, eine Methode zu schreiben, in der die Umrechnung von Celsius nach Fahrenheit erfolgt und eine korrekte Skalierung für die Anzeige der Werte zu bestimmen.

Ein Programmiercode für die Klasse `CelsiusFahrenheit` ist vorgegeben und muss an verschiedenen Stellen *vervollständigt* und *angepasst* werden.

```
1 import static java.lang.Math.round;
2 public class CelsiusFahrenheit {
3
4     // Fehlende Methode hier hinzufügen
5
6     public static void main(String[] args) {
7         System.out.println("Celsius Fahrenheit Converter");
8         System.out.println("=====");
9         System.out.println("Celsius \t Fahrenheit");
10
11         for (int c = 5; c < 20; c++) {
12
13             System.out.println(c + "\t" + celsiusFahrenheit(c));
14         }
15
16     }
17 }
```

Als Ausgabe wird Folgendes erwartet:

```
1 Celsius-Fahrenheit-Konverter
2 =====
3 Celsius      Fahrenheit
4 5            41
5 6            43
6 7            45
7 8            46
8 9            48
9 10           50
```

Übung 2: Racing Car Game

Wir werden ein einfaches **Autorennspiel** erstellen. Zuerst erstellen wir eine Klasse `RacingCar` mit den folgenden Attributen:

- `racer` speichert den Namen des Fahrers. Dieses Attribut muss ein nicht leerer `String` sein und sollte bei der Instanziierung des Objekts initialisiert werden.
- `speed` speichert die Geschwindigkeit des Autos. Dieses Attribut kann nur nicht-negative Integer-Werte enthalten und muss **kleiner als oder gleich** einer Höchstgeschwindigkeit sein.
- `pos` ist eine Ganzzahl, die die Position des Fahrzeugs angibt und nur nicht-negative Werte annehmen kann.

Jedes Auto hat außerdem die folgenden weiteren Attribute:

- `maxSpeed`, das die maximale Geschwindigkeit angibt, die das Auto haben kann. Dieses Attribut sollte bei der Instanziierung des Objekts initialisiert werden.
- `finish`, das die Zielstrecke speichert, die das Auto durchfahren muss. Bei der Initialisierung sollte es auf -1 gesetzt werden.

Die Klasse hat die folgenden Methoden:

- `start(initSpeed, finishDistance)`: stellt die Geschwindigkeit des Autos auf einen Anfangswert, setzt die Zielentfernung auf den übergebenen Wert und setzt auch die Position des Autos auf 0.
- `race(acceleration)`: nimmt einen ganzzahligen Wert für die Beschleunigung; passt zuerst die Geschwindigkeit des Autos an und aktualisiert danach die Position des Autos.
- `isFinished()`: berechnet einen Booleschen Wert (**true** oder **false**) und gibt an, ob das Auto die Ziellinie erreicht hat.

Nun Ihre Aufgaben:

1. Erstellen Sie ein **Klassendiagramm** passend zur textuellen Beschreibung. Diskutieren Sie mit Ihrem Sitznachbarn.
2. **Implementieren** Sie die Klasse! Nehmen Sie den bereitgestellten Programmcode als Vorlage. Die `main`-Methode muss ohne Fehler ausführbar sein.

Übung 3: Sparbuch

Die Aufgabe besteht darin, die Transaktionen *Geld abheben* und *Geld einzahlen* sowie *Kontostand anzeigen* in einem Programm zu implementieren. Wenn der Kontostand unter 0 Euro liegt, soll eine Warnung ausgegeben werden.

1. **Entwerfen** und **implementieren** Sie eine Klasse `Sparbuch`, die entsprechende Methoden für diese Transaktionen zur Verfügung stellt. Achten Sie darauf, dass Ihre Klasse `Sparbuch` in die schon bereitgestellte Klasse `Bankautomat` eingebunden werden kann. Die Klasse `Bankautomat` dient als Hauptprogramm und bietet einen Infobildschirm für die Interaktion mit dem Benutzer.
2. Die Klasse `Bankautomat` fragt aktuell nur nach einer Option und beendet dann das Programm. Können Sie das Programm erweitern, damit es so lange im Menü bleibt, bis der Benutzer die Option "Beenden" auswählt?
3. (Bonus) Aktuell kann noch so einiges schiefgehen, wenn der Benutzer nicht die erwarteten Eingabenformate einhält. Testen Sie das Programm mit verschiedenen Eingaben und versuchen Sie, die Klasse `Bankautomat` entsprechend robuster zu implementieren.
4. (Bonus) Wenn das Programm beendet wird, geht aktuell der Kontostand verloren. Wie könnte man das lösen?