Lehrstuhl für Angewandte Informatik I
Prof. Dr. Bernhard Westfechtel
Sandra Greiner

WS 2017/18
03. November 2017

Konzepte der Programmierung

Übung 3 – Abgabe: 16.11.2017

Aufgabe 3.1 (2D-Arrays)

Erweitern Sie Ihre Klasse `DoubleArrays` aus Blatt 2 um folgende **Klassenmethoden**:

(a) `erzeugeMatrix2D()`
Die Methode bekommt die Anzahl der Zeilen und die Anzahl der Spalten übergeben und erzeugt ein 2D-Array entsprechender Größe.
Anschließend soll der Benutzer die Einträge der Matrix eingeben können. Diese werden in der Matrix gespeichert und die gefüllte Matrix zurückgegeben.

(b) `male2DMatrix()`
Die Methode bekommt eine 2D-Matrix übergeben und gibt deren Werte **formatiert** auf der Standardausgabe aus (, d.h. Sie sollten beispielsweise bedenken, dass einzelne Einträge verschieden viele Stellen besitzen können). Die Ausgabe für eine 2x3-Matrix könnte, z.B. so aussehen:

```
13.0  4.0  7.0
2.0   5.0  80.0
```

(c) Ändern Sie die main-Methode in dieser Klasse ab. Fragen Sie den Benutzer nach der Größe des Arrays. Erzeugen Sie dann eine entsprechende 2D-Matrix. Lassen Sie diese zuletzt auf der Konsole ausgeben. Dabei sollen die zuvor implementierten Methoden verwendet werden.

(**6 + 3 + 3 = 12 Punkte**)

Aufgabe 3.2 (Mobiltelefon)

Zum Betrieb eines Mobiltelefons wird eine SIM-Karte benötigt, die gewöhnlich genau einem Mobiltelefon zugeordnet ist. Die SIM-Karte authentifiziert das Mobiltelefon gegenüber dem *Netzanbieter* (z. B. T-Mobile, Vodafone, etc.). Zusätzlich besitzt die SIM-Karte Speicherfunktionen, z. B. für die zuletzt gewählten Telefonnummern.

(a) Erstellen Sie eine Klasse `SimKarte` mit den **privaten Objektfeldern** *Netzanbieter* und *zuletzt gewählte Rufnummer*, die jeweils als `String` gespeichert werden.
Warum ist es sinnvoll, das Objektfeld als privat zu deklarieren?
Warum wird ein Objektfeld und kein Klassenfeld verwendet?

Die Antworten zu den Fragen können auch im Quelltext gegeben werden.

(b) Schreiben Sie einen **Konstruktor** für die Klasse `SimKarte`. Er soll den Namen des Netzanbieters, der ihm übergeben wird, setzen. Schreiben Sie auch eine Methode `gibNetzAnbieter()`, die den Namen des Netzanbieter zurückgibt.
Schreiben Sie zwei weitere Methoden: eine (`setzeRufnummer()`) bekommt die zuletzt gewählte Telefonnummer als `String`-Wert übergeben und speichert diese auf der Karte. Die andere bietet die Möglichkeit, die gespeicherte Rufnummer abzurufen (`gibRufnummer()`).

(c) Erstellen Sie eine Klasse `Mobiltelefon`, welche ihre zugehörige *SIM-Karte* in einem privaten Objektfeld speichert. Dem Konstruktor muss ein entsprechendes SIM-Karten-Objekt übergeben werden.

(d) Das Mobiltelefon besitzt eine Methode `void einschalten()`, wodurch der Name des Netzbetreibers auf dem Display (hier: Konsole) ausgegeben wird.

Die Methode `waehlen()` des Mobiltelefons gibt die zu wählende Rufnummer auf dem Display aus und speichert die Rufnummer auf der SIM-Karte ab.
Durch die Methode `wahlWiederholung()` wird die zuletzt gewählte Nummer von der SIM-Karte gelesen und noch einmal gewählt.

Überprüfen Sie die Richtigkeit Ihrer Implementierung, indem Sie in BlueJ mehrere Objekte von den jeweiligen Klassen erzeugen. Sie können ein Objekt erzeugen, indem Sie einen Rechtsklick auf die Klasse in Ihrem Projekt machen und `new <Klassenname>(...)` wählen. Dabei müssen Sie entsprechende Werte, der Konstruktor erwartet übergeben. Sie können den Objekten Namen geben, die Sie wiederum benutzen können um Sie einem weiteren Objekt zu übergeben. Die Eigenschaften des Objekts können Sie sehen, indem Sie auf das erstellte Objekt rechtsklicken und `Inspizieren` wählen.

(3 +5 +2 + 6 = 16 Punkte)