Aufgabenblatt 15

Lesen Sie Kapitel 19 (Ausnahmebehandlung, außer Stack-Beispiel) und 23.2 (Datenströme, in älterer Auflage 22.2) des Lehrbuchs. Die Klassen In und Out dürfen bei der Lösung der folgenden Aufgaben nicht verwendet werden.

a) Beantworten Sie zum folgenden Programm die darunter angegebenen Fragen.

```
( 1) class TrafficExample {
( 2)   public static void main(String[] args) {
( 3)     // Train train = new Vehicle();
( 4)     // Out.println(((Vehicle) (new Train())).nCoaches);
( 5)     Vehicle vehicle = new Train(5);
( 6)     vehicle.set(10);
( 7)     ((Train) vehicle).set(10);
( 8)     vehicle.setDefault();
( 9)     ((Vehicle) vehicle).setDefault();
(10)   }
(11) }
(12)
(13) class Vehicle {
(14)   static final double MAX_VELOCITY = 200.0;
(15)   double velocity = 0.0;
(16)   Vehicle() { Out.println("Aufruf des parameterlosen Konstruktors"); }
(17)   Vehicle(double velocity) { this.velocity = velocity; }
(18)   void setDefault() { velocity = 0.0; }
(19)   void set(double velocity) { this.velocity = velocity; }
(20)   boolean checkVelocity() { return velocity < MAX_VELOCITY; }
(21) }
(22)
(23) class Train extends Vehicle {
(24)   static final int  N_SEATS_PER_COACH = 72;
(25)   int nCoaches = 0;  // Anzahl Waggons
(26)   Train() { }
(27)   Train(int nCoaches) { this.nCoaches = nCoaches; }
(28)   Train(double velocity) { super(velocity); }
(29)   Train(double velocity, int nCoaches) {
(30)     super(velocity); this.nCoaches = nCoaches;
(31)   }
(32)   void setDefault() {
(33)     super.setDefault(); nCoaches = 0;
(34)   }
(35)   void set(int nCoaches) { this.nCoaches = nCoaches; }
(36)   int nSeats() { return nCoaches * N_SEATS_PER_COACH; }
(37) }
```

1) Nennen Sie, wenn möglich, je eine Methode (keinen Konstruktor!), die überschrieben, überladen bzw. "einfach nur" vererbt wird.

(3 Punkte)

2) Warum sind die Anweisungen in den Zeilen (3) und (4) unzulässig?

(2 Punkte)

3) An welcher Stelle der Programmausführung (nicht im Kommentar) wird der parameterlose Konstruktor `Vehicle()` aufgerufen?

(1 Punkt)

4) Lässt sich der Konstruktor in den Zeilen (29)–(31) unter Verwendung von `this()` kürzer programmieren? Begründen Sie.

(2 Punkte)

5) Begründen Sie, warum sowohl in Zeile (8) als auch in Zeile (9) die Methode `setDefault()` der Klasse `Train` aufgerufen wird.

(2 Punkte)

(6) Welches Attribut wird durch den Methodenaufruf in Zeile (6) verändert?

(1 Punkt)

(7) Welches Attribut wird durch den Methodenaufruf in Zeile (7) verändert?

(1 Punkt)

b) Die Klasse `MyFile` (siehe Moodleseite) erwartet als Kommandozeilenparameter einen Dateiname. Sie erzeugt dann die Datei und trägt eine Zeichenreihe ein.

Beim Versuch, die Klasse zu übersetzen, erhält man eine Fehlermeldung. Korrigieren Sie das Programm, so dass es fehlerfrei übersetzt und ausgeführt werden kann. Leiten Sie dabei die Ausnahme weiter. Sehen Sie sich die erzeugte Datei in einem Editor an. Wenn Sie Aufgabe c) erfolgreich gelöst haben, genügt die Abgabe der Lösung zu c).

(2 Punkte)

c) `MyFile.java` Wenn man beim Aufruf des Programms aus Aufgabe b) vergisst, einen Kommandozeilenparameter anzugeben, bricht es mit der Fehlermeldung

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
at MyFile.main(MyFile.java:6)
```

ab. Korrigieren Sie das Programm, so dass es eine verständliche Fehlermeldung (und nur diese!) ausgibt. Beispiel:

```
%java MyFile
Aufruf: MyFile dateiname
```

Die Aufgabe *muss* durch Abfangen der `ArrayIndexOutOfBoundsException` gelöst werden.

(5 Punkte)

d) `MyNewFile.java` Schreiben Sie ein Java-Programm, das vom Nutzer einen Dateinamen abfragt und dann eine Zeichenreihe in diese Datei schreibt. Im Unterschied zu den Aufgaben b) und c) darf noch keine Datei dieses Namens existieren. Falls der Nutzer den Name einer bereits existierenden Datei eingibt, soll er so lange nach einem neuen Dateiname gefragt werden, bis er diesen eingibt.

Hinweis: Die Aufgabe kann mit Hilfe der Klassen `java.io.File` und `java.util.Scanner` gelöst werden.

Beispiele:

```
Bitte Dateiname eingeben: f.txt
Habe soeben in die Datei f.txt geschrieben

Bitte Dateiname eingeben: f.txt
Bitte Dateiname eingeben: f.txt
Bitte Dateiname eingeben: g.txt
Habe soeben in die Datei g.txt geschrieben
```

(9 Punkte)

e) `Midnight.java` Schreiben Sie eine Java-Methode

```
static int minutesToMidnight(int currentHour, int currentMinute)
            throws InvalidTimeException
```

Die Methode bekommt die gegenwärtige Uhrzeit übergeben und rechnet aus, wie lange es noch bis Mitternacht dauert. Falls die übergebenen `int`-Werte keiner sinnvollen Uhrzeit entsprechen, wird eine Ausnahme des Typs `InvalidTimeException` ausgelöst. Dabei sollen im Ausnahmeobjekt Informationen zum konkreten Fehler abgespeichert werden (ist Stunde oder Minute falsch und wie lautet der falsche Wert). Programmieren Sie diese Ausnahmeklasse selbst.

Schreiben Sie dann ein Rahmenprogramm zum Test der Methode. Das Programm soll sich so verhalten, wie die folgenden Beispiele demonstrieren (inkl. formatierte Ausgabe der gelesenen Uhrzeit). Die Informationen zum Fehler (beispielsweise `333 ist ungueltige Minute`) sollen mittels `System.out.println(e);` ausgegeben werden. Dabei bezeichnet `e` das Ausnahmeobjekt. Wahrscheinlich erhalten Sie zunächst eine Ausschrift der Form `InvalidTimeException: 333 ist ungueltige Minute`. Überlegen Sie sich, wie Sie die Ausschrift verkürzen und trotzdem `System.out.println(e);` verwenden können.

Beispiele:

```
Aktuelle Stunde: 2
Aktuelle Minute: 3
Gelesen: 2:03
Noch 1317 Minuten bis Mitternacht

Aktuelle Stunde: 3
Aktuelle Minute: 333
Gelesen: 3:333
333 ist ungueltige Minute

Aktuelle Stunde: -1
Aktuelle Minute: 100
Gelesen: -1:100
-1 ist ungueltige Stunde
```

(12 Punkte)