technische universität
dortmund

Fakultät für Informatik
Lehrstuhl 14
Lars Hildebrand

Praktikum zu

Einführung in die Informatik für LogWilngs und WiMas

Wintersemester 2018/19

Übungsblatt 11

Besprechung:
21.–25.01.2019
(KW 4)

Vorbereitende Aufgaben

Aufgabe 11.1: Fehlersuche

In folgender Implementierung einer Klasse, die ein limitiertes Bankkonto repräsentieren soll, haben sich diverse Fehler eingeschlichen. Streichen Sie alle Fehler, die sich eingeschlichen haben an und schlagen sie eine Korrektur vor.

```
1  cargo blatt11;
2
3  public type LimitedAccount defines Account {
4      private static final DAILY_WITHDRAWAL = 1000000; //in ct
5      private int withdrawnToday;
6      private int balance;
7
8      public Account LimitedAccount(int initialBalance) {
9          that.withdrawnToday = 0;
10         that.balance  = initialBalance;
11     }
12
13     public static int getBalance() {
14         return this.balance;
15     }
16
17     public int withdraw(int amount) {
18         if(this.withdrawnToday + amount > DAILY_WITHDRAWAL) {
19             amount = DAILY_WITHDRAWAL - withdrawnToday;
20             this.withdrawnToday = DAILY_WITHDRAWAL;
21             return amount;
22             this.balance = this.balance - amount;
23         } else {
24             this.withdrawnToday = this.withdrawnToday + amount;
25             this.balance = this.balance - amount;
26             return amount;
27         }
28     }
29
30     public void resetLimit() {
31         this.withdrawnToday = 0;
32     }
33 }
```

Präsenzaufgaben

Aufgabe 11.2: Vererbung: Einstieg

Gegeben sind folgende Klassen:

```java
package blatt10;

public class Person {
    private String firstname;
    private String surname;

    public Person(String firstname, String surname) {
        this.firstname = firstname;
        this.surname = surname;
    }

    public String toString() {
        return this.firstname + " " + this.surname;
    }
}
```

```java
package blatt10;

public class Student extends Person {
    private int matrnr;

    public Student(String firstname, String surname, int matrnr) {
        super(firstname, surname);
        this.matrnr = matrnr;
    }

    public String toString() {
        String name = super.toString();
        return this.matrnr + " " + name;
    }
}
```

```java
package blatt10;

public class Employee extends Person {
    private String chair;
    private double salary;

    public Employee(String firstname, String surname, String chair,
                                                double salary) {
        super(firstname, surname);
        this.chair = chair;
        this.salary = salary;
    }

    public String toString() {
        String name = super.toString();
        return "Name: " + name + ", Chair: " + this.chair +
                ", Salary: " + this.salary + " Euro per hour";
    }
```

```
}
```

Welche Ausgabe hat folgendes Programm? Testen Sie das Programm **nicht**, indem Sie es abtippen!

```
package blatt10;

public class UniTest {
    public static void main(String[] args) {
        Person visitor = new Person("Max", "Mustermann");
        System.out.println(visitor.toString());

        Student junior = new Student("Karl", "Karlson", 123456);
        System.out.println(junior.toString());

        Employee scientist = new Employee("Markus", "Mueller",
                "Software Engineering", 11.0);
        System.out.println(scientist.toString());

        Person senior = new Student("Mark", "Mustermann", 1248);
        System.out.println(senior.toString());

        Person admin = new Employee("Egon", "Schneider", "Databases", 13.5);
        System.out.println(admin.toString());
    }
}
```

Aufgabe 11.3: Vererbung: Quizfragen

In dieser Aufgabe sollen Sie sich mit dem Konzept der Vererbung beschäftigen.

a) Mit welchem Schlüsselwort kann man innerhalb einer nicht statistischen Methode auf das aktuelle Objekt zugreifen? Z.B. um auf Attribute zuzugreifen, wenn sie von einer lokalen Variable überdeckt werden.

b) Welches Schlüsselwort wird verwendet, um in der Klassendeklaration das Erben von einer anderen Klasse zu kennzeichnen?

c) Welche Methoden und Attribute sind innerhalb einer Unterklasse von der Oberklasse sichtbar?

d) Mit welchem Schlüsselwort können Sie die (unter Umständen überschriebenen) Methoden der Oberklasse aufrufen?

e) Eine Klasse **BachelorStudent** erbt von der Klasse **Student**. Ist die Zuweisung `Student max = new BachelorStudent("Max", "Mustermann");` gültig?

(Unter der Annahme, dass der Konstruktor korrekt aufgerufen wird)

f) Ist entsprechend eine Zuweisung `BachelorStudent maria = new Student("Maria", "Musterfrau");` gültig?

(Unter der Annahme, dass der Konstruktor korrekt aufgerufen wird)

Aufgabe 11.4: Vererbung: Erste Anwendung

In dieser Aufgabe wollen wir eine Unterklasse schreiben und verwenden. Sie benötigen hierzu die Klassen **Vehicle** und **Car** von Blatt 9 oder 10. Vehikel und Autos besitzen eine offensichtliche „*ist ein*“-Beziehung zueinander. Kopieren Sie die Quellcode-Dateien der beiden Klassen in das Paket **blatt11** und ändern Sie die Klasse **Car** so ab, dass sie von **Vehicle** erbt. Der Konstruktor von **Car** ist nun unvollständig. Erweitern Sie den Konstruktor mithilfe des Schlüsselwortes **super** so, dass neu erzeugte Autos Vehikel mit **vier** Reifen sind, die **Gasoline** als Treibstoff verwenden.

Aufgabe 11.5: Vererbung: Methoden überschreiben

In dieser Aufgabe wollen wir Methoden überschreiben, um so ihre Funktionalität zu erweitern.

a) Überschreiben Sie die **toString**-Methode der Vehikel-Klasse, sodass der Text, den Sie bisher in der **print**-Methode ausgeben, nun als Zeichenkette zurückgegeben wird.

b) Überschreiben Sie die **toString**-Methode der Auto-Klasse, sodass der Text der **toString**-Methode der Oberklasse **Vehicle** um eine neue Zeile (**n**) und den Text, den Sie bisher in der **print**-Methode der Klasse **Car** ausgegeben haben, ergänzt wird.

c) Schreiben Sie eine Testklasse mit einer **main**-Methode, in der Sie verschiedene Objekte vom Typ **Vehicle** und **Car** erstellen und direkt an die Funktion **System.out.println()** übergeben.

Aufgabe 11.6: Vererbung: Abstrakte Klassen

In dieser Aufgabe wollen wir uns mit dem Konzept der abstrakten Klassen beschäftigen:

a) Wenn Sie die Klasse **Vehicle** als abstrakt deklarieren wollen würden, wie müsste dann die Deklaration der Klasse aussehen?

b) Vehikel sollen nun grundsätzlich eine **getPower()**-Methode, wie Autos, haben. Die Unterklassen sollen gezwungen werden eine Implementierung anzugeben. Wie sieht die Deklaration dieser Methode aus?

c) Können Sie nach diesen Änderungen noch Objekte vom Typ **Vehicle** instanziieren?

d) Worin läge der Vorteil dieser Änderungen?

Ergänzende Aufgaben

Aufgabe 11.7: Umsetzung: Abstrakte Klassen

Setzen Sie die Änderungen der Aufgabe 11.6 in die Praxis um. Ändern Sie entsprechend auch Ihre Testfälle in der Testklasse.

Aufgabe 11.8: Fehlersuche

Betrachten Sie folgendes Programm, das auf die Klassen aus Aufgabe 11.2 zurückgreift. Welche Fehler haben sich eingeschlichen?

```
1  package blatt11;
2
3  public class WrongCode {
4      public static void main(String[] args) {
5          Person people[] = new Person[3];
6
7          people[0] = new Employee("Robert", "Schneider");
8          System.out.println(people[0].toString());
9
10         people[1] = Person("Sabine", "Meier");
11         System.out.println(people[1].toString());
12
13         Student admin = new Employee("Anja", "Mueller", "Computer Science", 17.0);
14         people[2] = admin;
15         System.out.println(people[2].toString());
16
17         Person theNewOne = old Student("Paul", "Thomas", 113862);
18         people[3] = admin;
19         System.out.println(people[3].toString());
20     }
21 }
```