Hochschule
Bonn-Rhein-Sieg

Fachbereich Informatik
Prof. Dr. Peter Becker

Datenstrukturen und Algorithmen
Sommersemester 2013
27. März 2013

Aufgabenblatt 1

Aufgabe 1 (Abstrakte Klasse, 10 Punkte)

Wir wollen in Java Personen als Objekte mit den folgenden Eigenschaften modellieren:

- Jede Person hat Namen und Alter, die bei der Instanziierung unveränderlich festgelegt werden.
- Jede Person ist entweder Frau oder Mann. Nur Frauen oder Männer sollen instanziiert werden können.
- Eine Frau lacht "hihihihi", ein Mann "hohohoho". Für Personen soll es eine Methode `lachen()` geben, die das Lachen als String liefert.
- Wenn eine Frau schläft, dann macht sie "zsss zsss", ein Mann "schnarch schnarch". Für Personen soll es eine Methode `schlafen()` geben, die die Schlafgeräusche als String liefert.
- Wenn man eine Frau nach ihrem Alter fragt, macht sie sich stets fünf Jahre jünger als sie eigentlich ist. Ein Mann gibt dagegen immer sein richtiges Alter an. Für Personen soll es eine Methode `gibAlter()` geben, die das Alter liefert.
- Für jede Person soll es eine Methode `dasBinIch()` geben, die Name, angebliches Alter (nach Aussage der Person), Lach- und Schlafgeräusche auf die Standardausgabe ausgibt.

(a) Erstellen Sie in Java eine Klassenhierarchie für Personen mit den oben genannten Eigenschaften. Verwenden Sie in angemessener Weise Vererbung und abstrakte Klassen.

(b) Definieren Sie eine Klasse, die eine Klassenmethode zur Erzeugung einer Personenmenge anbietet. Die Personenmenge soll als Feld repräsentiert werden. In solch einer Personenmenge soll es Frauen und Männer geben.

(c) Implementieren Sie eine `main`-Methode, in der Sie die Methode aus (b) nutzen, um eine Personenmenge zu erzeugen. Anschließend soll für jede Person der Menge die Methode `dasBinIch()` ausgeführt werden.

(d) Erstellen Sie ein UML-Klassendiagramm für die Hierarchie der Personenklassen.

Aufgabe 2 (Wiederholung: Klassenvariable und -methode, 4 Punkte)

Erweitern Sie die Klassendefinitionen aus Aufgabe 1 so, dass gezählt wird, wie viele Personen, Frauen und Männer erzeugt werden. Lassen Sie sich diese Zahlen in der `main()`-Methode ausgeben.

Aufgabe 3 (6 Punkte)

Gegeben sei der folgende Quelltext eines Java-Programms:

```
abstract class Hund {
    public abstract void foo();

    public void bellen() {
        this.foo();
        this.bar();
    }

    public void bar() { System.out.print("wau "); }
}

class Schaeferhund extends Hund {
    public void foo() { System.out.print("wuff "); }

    public void bar() { System.out.print("wooh "); }
}

public class Main {
    public static void main(String[] args) {
        Schaeferhund hasso = new Schaeferhund();
        hasso.bellen();
    }
}
```

(a) Der Java-Compiler akzeptiert den obigen Quelltext, obwohl

- in `main()` die Methode `bellen()` der Klasse `Schaeferhund` aufgerufen wird, in `Schaeferhund` aber gar keine Methode `bellen()` definiert ist und
- in `bellen()` der Klasse `Hund` die Methode `foo()` aufgerufen wird, `foo()` in `Hund` aber nicht implementiert ist.

Erläutern Sie, warum die oben genannten Punkte keine Probleme darstellen.

(b) Welche Ausgabe produziert das Java-Programm? Erläutern Sie, wie es zu dieser Ausgabe kommt.

Hinweis: Der Quelltext des Java-Programms ist auf der Homepage der Veranstaltung verfügbar.

Hilfe zur Lösung der Aufgaben erhalten Sie in den Übungen der KW 14. Abgabe der Lösungen bis **Sonntag, 7. April 2013, 24 Uhr**. Über die genauen Modalitäten der Abgabe werden Sie noch via E-Mail informiert.