



Übungsblatt 4

Programmierung und Softwareentwicklung (WiSe 2019/2020)

Abgabe: Fr. 15.11.2019, 23:59 Uhr — Besprechung: KW 47

- Bitte lösen Sie die Übungsaufgabe in **Gruppen von 2 Studenten**.
- Dieses Übungsblatt besteht aus zwei Teilen (A+B). Teil A ist in der Präsenzübung zu lösen. Teil B ist in Heimarbeit (Gruppe von 2 Studenten) zu lösen und rechtzeitig abzugeben.
- Für Ihre Abgabe erstellen Sie bitte eine Java-Datei (Endung .java), die Ihre Lösung aus Aufgabe 2 Teil B beinhaltet. Erstellen Sie zudem EINE weitere PDF-Datei (Endung .pdf) mit Ihrer Lösung aus Aufgabe 1 aus Teil B.
- Geben Sie zu Beginn der Dateien Ihre Namen (Vor- und Nachname), die Matrikelnummern und die E-Mail-Adressen an.
- Benennen Sie die Dateien nach dem folgenden Schema:
PSE[ÜB-Nr]-[Aufgaben-Nr]-[Nachnamen der Teammitglieder]-[Nachname des Tutors].[java/pdf].

Lernziel: Dieses Übungsblatt dient dazu, Sie mit den grundlegenden Prinzipien von Objekten, Klassen und Schnittstellen vertraut zu machen. Sie sollen lernen, was Objekte sind und wie man mit ihnen anhand ihrer Schnittstellen umgeht.

Vorbereitung: Stellen Sie sicher, dass Sie Übungsblatt 1 - 3 absolviert haben, alle Software installiert und funktionsfähig ist (BlueJ und Java 1.11), und Ihr W-Lan richtig konfiguriert ist.

Punkte: Dieses Übungsblatt enthält zwei Teile. Teil A mit 2 Aufgaben und Teil B mit 2 Aufgaben. Im Teil B können Sie bis zu 30 Punkte erzielen.

1 Teil A - Präsenzaufgaben

Aufgabe 1 Klassen, Objekte und Schnittstellen

In dieser Aufgabe geht es darum, das Verständnis über den Zusammenhang zwischen Klassen, Objekten und deren Schnittstellen zu üben und zu vertiefen.

Hierzu betrachten wir als Beispiel ein Programm zur Simulation von Fußballspielen. Konkret soll es hier um die Fußballspieler gehen, die in einem Spiel simuliert werden.

- Erläutern Sie den Zusammenhang zwischen Klassen, Objekten und deren Schnittstellen am Beispiel eines Fußballspielers in der Simulation und geben Sie geeignete Beispiele. Was wäre ein möglicher Name für die Klasse? Was wären mögliche Namen der Objekte?
- Skizzieren Sie eine mögliche Schnittstelle der Klasse für Fußballspieler. Geben Sie hierzu die Signaturen möglicher Abfragen und Kommandos (ohne deren Implementierung) in Java-ähnlicher Syntax an und verwenden Sie Javadoc zur Dokumentation. Verwenden Sie bereits bekannte und neue Klassen in den Rückgabe- und Parametertypen. Beachten Sie unsere Stilregeln für Operationen und Javadoc und dessen Vollständigkeit. Vor- und Nachbedingungen dürfen in natürlicher Sprache angegeben werden. Klassifizieren Sie Ihre Operationen in Kommandos und Abfragen.
- Geben Sie eine beispielhafte Verwendung von Objekten der Klasse in einem Java-Programm — unter Benutzung der zuvor definierten Schnittstelle — an. Gehen Sie hierzu davon aus, dass eine oder mehrere (fest, aber beliebig) benannte Objektinstanzen der Klasse zur Verfügung stehen. Orientieren Sie sich an einer typischen Spielsituation wie: Thomas Müller macht einen Anstoß; er läuft mit dem Ball nach vorne; er umläuft einen gegnerischen Spieler und schießt.

Aufgabe 2 Verträge

Ziel dieser Aufgabe ist es, die Spezifikation von Verträgen (ohne und als Herausforderungen mit JML) zu üben. Hierzu betrachten wir in dieser Aufgabe die Klasse `Hamster.java`, die Ihnen unter <https://github.com/RSS-PSE-WS1920/exercise-sheet-4> zur Verfügung steht. Hinweis: Wenn Sie fortgeschrittener sind und JML nutzen wollen, sollten Sie zuerst weitere Informationen zu JML recherchieren. Ein Startpunkt könnte folgendes IBM Tutorial sein: <https://www.ibm.com/developerworks/library/j-jml>

(a) Definieren Sie nun für die folgende Abfrage Vor- und Nachbedingungen:

- `mouthEmpty`

(b) Definieren Sie für die folgenden Kommandos Vor- und Nachbedingungen:

- `turnLeft`
- `pickGrain`

Dies kann zunächst mittels eines beliebigen (Pseudo-)Formalismus oder mittels natürlicher Sprache erfolgen.

(c) Definieren Sie geeignete Klasseninvarianten.

(d) **Herausforderung 1:** In der Vorlesung wurden in JML definierte Vor- und Nachbedingungen für das Kommando `move` vorgestellt. Gibt es Fälle, die noch nicht behandelt wurden? Wenn ja, überarbeiten Sie die Definitionen. Informieren Sie sich hierzu u. a. über das `assignable`-Schlüsselwort.

(e) **Herausforderung 2:** Definieren Sie die oben ermittelten Vor- und Nachbedingungen sowie Klasseninvarianten nun mittels (Open)JML direkt in der Java-Klasse `Hamster.java`. Ggf. können Sie zusätzliche JML-Modellvariablen anlegen. Betrachten Sie hierzu das Vorgehen zur JML-Definition für `move` oben in der Klasse.

2 Teil B - Heimarbeit

Aufgabe 1 Abstract Syntax Tree (AST) und Programmanalyse

Gegeben ist folgende, einfache Java Klasse. Die Objekte `myLog` und `database` sind in `DatabaseIO` definiert.

- (a) (10 Punkte) Zeichnen Sie den abstrakten Syntaxbaum von Listing 7 analog zur selbigen Aufgabe aus Übungsblatt 3.

```

1  class DatabaseAdapter extends DatabaseIO {
2
3      String getFirstName() {
4          return database.getFirstName();
5      }
6
7      Integer getAge() {
8          myLog.debug("Get_Age_called");
9          return database.getAge();
10     }
11 }
12 }
```

Listing 7: Java AST example code

- (b) (10 Punkte) Finden und nennen Sie alle syntaktischen und statisch semantischen Fehler in Listing 8. Klassifizieren Sie jeden Bug in syntaktisch und statisch semantisch, begründen Sie die Klassifizierung und schlagen Sie jeweils einen Bugfix vor.

```

1  classe Calculator extends 101 Calculator {
2      Sting getResult() {
3          return 2 * 2;
4      }
5  }
```

Listing 8: Buggy Java code

Aufgabe 2 Javadoc

Jemand hat damit begonnen, eine Taschenrechner-API für einen Taschenrechner mit ganzen Zahlen zu programmieren.

```

1  class Calculator {
2
3      Integer getSum(Integer a, Integer b) {
4          return a + b;
5      }
6
7  }
```

Listing 9: Integer Calculator

- (a) (10 Punkte) Ergänzen Sie Operationen für die folgenden mathematischen Operationen (die alle ausschließlich Ganzzahlen verarbeiten und zurückgeben sollen): Multiplizieren, Dividieren und die Berechnung der Fläche eines Rechtecks, die Ihre anderen Operationen nutzt. Schreiben Sie zur Klasse und allen Operationen eine passende, möglichst gute natürlichsprachliche Javadoc-Dokumentation. Denken Sie auch daran, Fälle zu beschreiben, die zu Problemen führen könnten (Pre- und Postconditions). Beachten Sie auch, dass Ganzzahlen in Java nur von $-2.147.483.648$ (abgekürzt `MIN_INT`) bis $2.147.483.647$ (abgekürzt `MAX_INT`) reichen und überlegen Sie sich, wo das zu Problemen führen könnte.

3 Teil L - Lehramtsteil

Dieser Teil enthält eine Präsenzaufgabe, die speziell für Lehramtsstudierende konzipiert ist. Das bedeutet allerdings nicht, dass “Nicht-Lehramtsstudierende” diese nicht auch bearbeiten dürfen. Sie bezieht sich auf den Vorlesungsstoff und die Übungsaufgaben, betrachtet diese aber aus einer didaktischen Perspektive.

Aufgabe 1 Andere Miniwelten

Es gibt noch andere Miniwelten, die speziell für den Einsatz im Unterricht und zum Erlernen der ersten Programmiersprache erstellt wurden. Eine davon ist die Welt des Käfers “Kara” (“JavaKara”: <https://www.swisseduc.ch/informatik/karatojava/javakara/>), welche vom SwissEduc-Team entwickelt wurde.

- (a) Laden Sie sich die `javakara.jar` von der Webseite oben herunter. Starten Sie die Jar anschließend mit einem Doppelklick.
- (b) Machen Sie sich etwas mit der Nutzeroberfläche vertraut und versuchen Sie sich an einer der einfachen Aufgaben (bspw. “Kara und die Blätter”).
 - Über die *Programmieren* Schaltfläche oben links kann man das Quellcodefenster öffnen. Dort finden sich oben im Kommentar auch alle Operationen, die Kara durchführen kann. Auf Kara selbst kann man über das Objekt `kara` zugreifen.
 - Mit den blauen Pfeilen auf der linken Seite kann man Kara Befehle erteilen, sofern er sich auf dem Spielfeld befindet.
 - Mit den Elementen rechts kann man diese auf dem Spielfeld verteilen. Elemente können auch nachträglich noch bewegt werden.
 - JavaKara hat ein paar integrierte Aufgaben, die über die *Aufgaben* Schaltfläche oben rechts erreichbar sind.
 - Im Aufgabenfenster finden sich drei Reiter:
 - **Aufgabe:** Die Aufgabenstellung.
 - **Welten:** Beispielwelten, die speziell für diese Aufgabe gedacht sind.
 - **Lösung:** Eine mögliche Lösung.
- (c) Erläutern Sie mit Ihrem Team Gemeinsamkeiten und Unterschiede zwischen der Karawelt und unserer Hamsterwelt (inkl. BlueJ).
- (d) Diskutieren Sie im Team, welche Vor- und Nachteile Kara im Unterricht im Vergleich zu Paule hätte.

Hinweise zu den Übungen:

- Die Abgabe erfolgt im ILIAS.
- Durch die Teilnahme am Übungsbetrieb können Sie sich für die Teilnahme an der Klausur qualifizieren.
 - Bestehen von min. 80% aller Übungsblätter.
 - Ein Übungsblatt gilt als bestanden, wenn 50% der Punkte erreicht wurden.
 - Teilnahme an min. 80% der Übungen.
 - Bestehen der Scheinklausur.

Viel Erfolg!