

---

## Übungsblatt 1

Ausgabe: 22.10.2012

Abgabe: 05.11.2012, 13Uhr

Besprechung: 12.11.2011 - 16.11.2011

---

## Allgemeine Hinweise (bitte aufmerksam lesen)

### Organisation

- Begleitend zur Vorlesung finden wöchentlich Tutorien statt. Diese werden von höhersemestrigen Studierenden geleitet. In diesen Tutorien findet eine **Nachbesprechung der Übungsblätter** und eine **Vertiefung des Vorlesungsstoffes** statt. Zusätzlich ist hier ausreichend Raum für **inhaltliche Fragen** zur Vorlesung und zur Diskussion mit dem Tutor und anderen Studierenden.
- Ihr Tutor wurde Ihnen durch das **WebInScribe-System**<sup>1</sup> zugeteilt und ist in allen Fragen zur Vorlesung Ihr erster **Ansprechpartner** und für die **Bewertung** Ihrer Lösungen der Übungsblätter zuständig.
- Falls Ihnen Ihr Tutor nicht weiterhelfen kann, stehen Ihnen für fachliche und organisatorische Fragen die beiden Übungsleiter Florian Merz (florian.merz@kit.edu, Sprechstunde Di. 13 - 14 Uhr, Gebäude 50.34, Raum 228) und Markus Iser (markus.iser@kit.edu, Sprechstunde Do. 15 - 16 Uhr, Gebäude 50.34, Raum 105) zur Verfügung.

### Übungsschein

- Im Laufe des Semesters werden 6 Übungsblätter zu bearbeiten sein. Auf jedem Übungsblatt können im Durchschnitt 20 Punkte erzielt werden. Insgesamt kann also eine Gesamtpunktzahl von 120 Punkten erreicht werden. Haben Sie am Ende des Semesters mindestens 50% der Gesamtpunktzahl (also insgesamt 60 Punkte) in den Übungsblättern erreicht, dann erhalten Sie den Übungsschein. Der Übungsschein ist unbenotet, er ist jedoch die erste Erfolgskontrolle und Voraussetzung für die Teilnahme an den Abschlussaufgaben, der zweiten Erfolgskontrolle. Zum erfolgreichen bestehen der Lehrveranstaltung sind beide Erfolgskontrollen zu erbringen.
- Sie können sich ab sofort im KIT-Studierendenportal<sup>2</sup> für den Übungsschein anmelden. Der Anmeldezeitraum endet am 31.03.2013. Dennoch empfehlen wir eine **rasche Anmeldung**.
- Neue Übungsblätter werden alle zwei Wochen ausgegeben. In der Regel sind die Aufgaben eines Übungsblatts so gestellt, dass **Teilaufgabe A** mit dem Wissenstand aus der Vorlesung/dem Tutorium in der Woche der Ausgabe und **Teilaufgabe B** mit dem Wissen aus der Vorlesung/dem

---

<sup>1</sup><https://webinscribe.ira.uka.de/>

<sup>2</sup><https://studium.kit.edu>

Tutorium in der Woche nach Ausgabe der Übungsblätter zu lösen sind. In keinem Fall ist für das Lösen der Übungsblätter Wissen notwendig, das vor Abgabetermin der Übungsblätter noch nicht gelehrt wurde.

- Die Übungen müssen **selbständig** und **ohne fremde Hilfe** bearbeitet werden. Sollten wir feststellen, dass es sich bei Ihrer Abgabe um eine nicht selbständig verfasste oder plagierte Lösung handelt, wird der Übungsschein Programmieren **sofort als “nicht bestanden (5,0)” bewertet**. Sie haben dann keine Möglichkeit, in diesem Semester die zweite Leistungskontrolle, nämlich die zwei Abschlussaufgaben, zu absolvieren.
- Bitte beachten Sie unbedingt, dass die Prüfung Programmieren eine **Orientierungsprüfung** ist, für die besondere Prüfungsmodalitäten gelten. Insbesondere müssen die Erfolgskontrollen **vor Ende des dritten Semesters** bestanden werden. Ein erster Versuch muss bereits Ende des zweiten Semesters geschehen.
- Bei **Fragen oder Unklarheiten** wenden Sie sich bitte **vor** der Abgabe des Übungsblattes an Ihren **Tutor** (per E-Mail oder persönlich im Tutorium). Es soll niemand sagen müssen: “Ich habe die Aufgabe nicht verstanden und konnte sie daher nicht bearbeiten.”
- Kommentare in den Abgaben können wahlweise in deutscher oder englischer Sprache verfasst sein, solange die Sprache innerhalb eines Übungsblattes einheitlich ist. Der Java-Code sollte jedoch **immer** englische Begriffe und Bezeichner verwenden.

## Praktomat

- Die Lösungen der Übungsblätter können **ausschließlich** über das **Praktomat-System**<sup>3</sup> abgegeben werden. Eine Papierabgabe wird nicht akzeptiert, planen Sie also gerade am Anfang genügend Zeit vor der Abgabe (wir empfehlen einen Tag) ein um sich mit dem System vertraut zu machen und Probleme bei der Abgabe zu beheben. Ihre Lösungen werden von Ihrem Tutor im Praktomat bewertet. Sie erhalten dann eine Zusammenfassung der Korrektur per E-Mail an Ihre KIT E-Mailadresse. Überprüfen Sie das Postfach dieses Accounts regelmäßig auf neue E-Mails. Alle Anmerkungen des Tutors können Sie anschließend im Praktomat online einsehen.
- Der Praktomat ist **nur über das Uni-Netz** erreichbar (benutzen Sie ggf. den VPN-Client des KIT<sup>4</sup>). Bei Problemen mit VPN wenden Sie sich bitte an das SCC (z.B. BIT 8000).
- Bei **technischen Problemen** mit Praktomat wenden Sie sich bitte unter Angabe ihrer Matrikelnummer per E-Mail an die Praktomat-E-Mail-Adresse<sup>5</sup> oder fragen Sie im entsprechenden Thema im Forum (s.u.). Im Forum brauchen und sollten Sie ihre Matrikelnummer nicht bekannt geben.
- Wir benötigen von Ihnen eine **Einverständniserklärung** (Disclaimer) zur elektronischen Speicherung und Nutzung Ihrer Daten für Programmieren. Bitte werfen Sie das entsprechende Formular<sup>6</sup> ausgefüllt und unterschrieben bis **spätestens 02.11.2012** in den Briefkasten der Vorlesung Programmieren im Keller des Gebäudes 50.34. Andernfalls wird Ihr Account im Praktomat gesperrt.

Gemeinsam mit dieser Einverständniserklärung müssen Sie unterschreiben, dass Sie die Prüfungsmodalitäten zur Kenntnis genommen haben, insb. die Tatsache, **dass bereits der erste**

<sup>3</sup><https://praktomat.info.uni-karlsruhe.de/>

<sup>4</sup><https://vpn.kit.edu>

<sup>5</sup>[praktomat@ipd.info.uni-karlsruhe.de](mailto:praktomat@ipd.info.uni-karlsruhe.de)

<sup>6</sup>Erhältlich im Studierendenportal (Link auf [http://verialg.iti.kit.edu/prog\\_ws12.php](http://verialg.iti.kit.edu/prog_ws12.php))

**Täuschungsversuch zum Nichtbestehen des Übungsscheins führt und Sie sich damit von der Teilnahme an den Abschlusssaufgaben (Teil 2 der Leistungskontrolle) ausschließen.**

- Mit Ihrem Zugang zu Praktomat haben Sie gleichzeitig auch Zugriff auf das betreute **Forum**<sup>7</sup>. Dort werden Sie u.A. über Klarstellungen und eventuelle Korrekturen der Aufgaben informiert, weshalb das Mitlesen des Forums **obligatorisch** ist. Sie können dort auch mit Ihren Kommilitonen über die Aufgaben in Programmieren diskutieren.

Das Forum ist auch für inhaltliche Fragen die Übungsblätter betreffend gedacht. Stellen Sie deswegen inhaltliche Fragen ausschließlich im Forum und nicht per E-Mail. Nur so können alle Teilnehmer der Vorlesung gleichermaßen davon profitieren.

## Anmeldung

Beachten Sie die folgenden Fristen:

System	Zweck	von	bis
Studierendenportal	Anmeldung zum Übungsschein	22.10.2012	31.03.2013
Praktomat	Anmeldung zur Lösungsabgabe	22.10.2012	02.11.2012
Praktomat-Disclaimer	Abgabe des Disclaimers	22.10.2012	02.11.2012

Die Seiten zum Anmelden erreichen Sie über den Vorlesungsarbeitsbereich der Programmieren-Vorlesung. Den Link finden Sie auf unserer Homepage [http://verialg.iti.uka.de/prog\\_ws12.php](http://verialg.iti.uka.de/prog_ws12.php).

## Abgabemodalitäten

Bei Abgabe Ihrer Lösung im Praktomat wird folgendes automatisiert geprüft:

- Eine maximale Zeilenbreite von 120 Zeichen
- Übersetzbarkeit mit dem Java-Compiler `javac`

Zum Einreichen der Lösungen zum **ersten** Übungsblatt ist lediglich das Bestehen der Prüfung “Maximale Zeilenbreite” notwendig. Beachten Sie jedoch, dass das erfolgreiche Übersetzen Ihrer Lösung mit `javac` auf den kommenden Übungsblättern obligatorisch sein wird. Wir empfehlen Ihnen daher, sich ggf. ausführlich mit den Compiler-Fehlermeldungen auseinanderzusetzen um Fehler zu beheben.

---

<sup>7</sup> Erreichbar über den Link im Praktomat-System (nachdem Sie sich dort angemeldet haben)

Die erste Teilaufgabe des ersten Übungsblattes beschäftigt sich mit dem Entwurf von Klassen. Das primäre Ziel ist es sich mit dem Erstellen von Klassen in Java vertraut zu machen. Darüber hinaus sollen Sie das Hinzufügen von Attributen zu Klassen üben und die Auswahl geeigneter Typen für diese Attribute erlernen.

$$\frac{(\overline{(*)})}{\overline{(*)}} \leq \frac{(\overline{(*)})}{\overline{(*)}}$$

Achten Sie beim Schreiben der Klassen auf:

- Bevor Sie mit der Klasse `Bike`, die Fahrräder modellieren soll, loslegen können, sollen Sie jedoch zunächst zwei Hilfsklassen erstellen, die Klassen `Gears` und `Wheels`.

Modellieren sie eine Kettenschaltung. Erstellen Sie dafür eine Klasse `Gears`. Diese Klasse erhält zwei Attribute: Eins für die Anzahl der vorderen Kettenräder (min. 1, max. 3) und eins für die Anzahl der Ritzel hinten (min. 1, max. 10). Wählen Sie für beide Attribute jeweils einen Datentyp mit dem alle benötigten Werte dargestellt werden können. Versuchen Sie gleichzeitig so wenig Speicherplatz wie möglich zu verschwenden.

Erstellen Sie eine weitere Klasse `Wheels`, die die Räder des Fahrrads modellieren soll. Auch diese Klasse erhält zwei Attribute: Felgendurchmesser und Reifenstärke. Der Felgendurchmesser wird üblicherweise ganzzahlig angegeben und beträgt maximal 700mm. Die Reifenstärke beträgt maximal 60mm, kann aber gebrochen sein (z.B 44,5). Berücksichtigen Sie dies bei der Wahl der Datentypen.

### A.3 Klasse Bike

Schreiben Sie jetzt die Klasse `Bike`.

#### A.3.1 Gangschaltung und Bereifung

Im Rahmen dieses Übungsblattes benötigt jedes Fahrrad sowohl Räder als auch eine Gangschaltung. Die Klasse `Bike` erhält also sowohl ein Attribut welches den Typ `Gears` hat, als auch ein Attribut des Typs `Wheels`.

#### A.3.2 Rahmenmaterial

Ein weiteres Attribut gibt an aus welchem der folgenden Materialien der Rahmen des Fahrrads besteht. Dabei handelt es sich je nach Fahrrad entweder um `Alu`, `Stahl` oder `Titan`. Im Rahmen dieses Übungsblatts bestehen Fahrräder niemals aus anderen Materialien als diesen dreien. Implementieren Sie dieses Attribut indem Sie einen passenden Datentyp hierfür finden und definieren.

#### A.3.3 Modellname

Jedes Fahrrad soll eine alphanumerische Modellbezeichnung erhalten. Modellbezeichnungen folgen dabei keinem festen Schema, sie können also aus beliebigen Kombinationen von Buchstaben und Zahlen bestehen.

#### A.3.4 Zubehör

Fügen Sie für die folgenden Ausstattungsmerkmale *je eine* weitere Eigenschaft hinzu, die angibt ob dieses vorhanden ist oder nicht: `Klingel`, `Beleuchtungseinrichtung`

### A.4 Preis

Fügen Sie den drei Klassen jeweils ein weiteres Attribut hinzu, das einen Preis in der Einheit `Cent` modelliert. Machen Sie sich Gedanken darüber, warum wir hier eine Modellierung in `Cent` gewählt haben. Schreiben Sie Ihre Gedanken hierzu als Kommentar in den `Java`-Code über das Attribut.

### A.5 Datentypen

Wie bereits erwähnt wird in `Java` für Ganzzahlen meist der Datentyp `int` benutzt und für Fließkommazahlen der Datentyp `double`. In diesem wie in kommenden Übungsblättern werden Sie daher auch meist diese beiden elementaren Typen verwenden. In dieser Teilaufgabe sollen Sie sich jedoch dennoch Gedanken über andere Ganz- und Fließkommazahl Datentypen in `Java` machen.

Geben Sie in dieser Datei für jedes numerische Attribut der Klassen `Bike`, `Gears` und `Wheels` an welcher der in `Java` verfügbaren Datentypen jeweils geeignet ist, um sämtliche sinnvollen Werte darzustellen und gleichzeitig ein Minimum an Speicherplatz benötigt. Legen Sie hierfür eine Textdatei `Datentypen.txt` an und halten Sie in dieser Datei sämtliche numerischen Attribute und die von Ihnen hierfür ermittelten Datentypen sowie eine kurze Erklärung Ihres Lösungsweges fest. Vergessen Sie nicht diese Datei zusammen mit ihren `Java`-Dateien im Praktomaten hochzuladen.

## B Methoden und Konstruktoren (10 Punkte)

Im folgenden sollen die in Teilaufgabe A erstellten Klassen durch Methoden ergänzt werden.

### B.1 Methode isStreetLegal

Die Straßenverkehrsordnung schreibt vor, dass ein Fahrrad über eine Klingel und Beleuchtung verfügen muss. Schreiben Sie für die Klasse `Bike` eine Methode `isStreetLegal`, die **true** zurückgibt, wenn das Fahrrad dieser Norm genügt, und **false**, wenn es dies nicht tut.

## B.2 Methode getNumberOfGears

Die Anzahl der möglichen Gänge beim Fahrrad hängt ab von der Anzahl der Kettenräder vorne und hinten. Schreiben Sie für die Klasse `Gears` eine Methode `getNumberOfGears`, die die Anzahl der Gänge (i.e. Kombinationsmöglichkeiten) zurückgibt.

### B.3 Methode getPrice

Sie haben in A.4 Ihren drei Klassen je ein Attribut für den Preis hinzugefügt. Dieses erhält den Preis der Bereifung (`wheels`), den Preis der Gangschaltung (`gears`) und den Basispreis des Fahrrads (`bike`). Schreiben Sie jetzt für `Wheels`, `Gears` und `Bike` je eine Methode `getPrice`. Diese Methoden sollen im Fall der Klassen `Wheels` und `Gears` einfach den Wert des eigenen Attributs zurückgeben. Die Methode `Bike.getPrice()` hingegen soll die Summe der Preise von Bereifung, Gangschaltung und des eigenen Basispreises zurückgeben. Benutzen Sie dabei in `Bike.getPrice()` nicht die Attribute von `Gears` und `Wheels`, sondern die jeweiligen `getPrice`-Methoden.

## B.4 Konstruktoren

Schreiben Sie für jede der drei Klassen `Bike`, `Wheels` und `Gears` einen Konstruktor. Für jedes Attribut einer Klasse soll dieser Konstruktor ein Argument des gleichen Typs übergeben bekommen. Im Rumpf der Konstruktormethode soll jedem Attribut der Wert des entsprechenden Arguments zugewiesen werden. Achten Sie auf eine aussagekräftige Benennung der Argumente.

## B.5 Fahrradfabrik

In der Realität fallen neue Fahrräder nicht vom Himmel, sondern sie werden in Fahrradfabriken und Werkstätten hergestellt. Schreiben Sie deshalb nun eine Klasse `BikeFactory`, die eine Fahrradfabrik modelliert und zum Erstellen neuer `Bike` Objekte dient. Die Klasse `BikeFactory` soll dabei über keine Attribute verfügen, aber über die folgende Methoden:

$$\begin{array}{ccccccc} \cdot & \circ & \circ & (\bar{\phantom{x}}) & (\bar{\phantom{x}}) & & \\ | & & & & & & \\ | & & & & & & \\ | & / & / & / & / & / & / \\ | & & & & & & \\ | & & & & & & \\ | & \text{---} & & \text{---} & & & \\ | & | & | & | & | & & \\ | & \text{---} & & \text{---} & & & \\ & & & & & \text{---}^{\text{a}} & \\ & & & & & \backslash < , & \\ & & & & & (\star) / & (\star) \end{array}$$

### B.5.1 Stadtrad Konfiguration

Schreiben Sie die Methode `createCityBike()`. Sie soll ein neues Objekt der Klasse `Bike` zurückgeben. Die Attribute sollen folgendermaßen gesetzt sein:

**Bereifung** Felgendurchmesser: 559mm, Reifendicke: 50mm, Preis: 100 €

**Gangschaltung** Zahnräder: 3, Ritzel: 1, Preis: 50 €

**Rahmen** Stahl

**Zubehör** Klingel, Beleuchtung

**Modellname** CB105

**Basispreis** 300 €

### B.5.2 Rennrad Konfiguration

Schreiben Sie die Methode `createRoadBike()`. Sie soll ein neues Objekt der Klasse `Bike` zurückgeben. Die Attribute sollen folgendermaßen gesetzt sein:

**Bereifung** Felgendurchmesser: 622mm, Reifendicke: 23mm, Preis: 100 €

**Gangschaltung** Zahnräder: 3, Ritzel: 9, Preis: 150 €

**Rahmen** Titan

**Zubehör** keine Extras

**Modellname** RB87

**Basispreis** 400 €

### B.5.3 Mountain-Bike Konfiguration

Schreiben Sie die Methode `createMountainBike()`. Sie soll ein neues Objekt der Klasse `Bike` zurückgeben. Die Attribute sollen folgendermaßen gesetzt sein:

**Bereifung** Felgendurchmesser: 559mm, Reifendicke: 50mm, Preis: 100 €

**Gangschaltung** Zahnräder: 3, Ritzel: 9, Preis: 150 €

**Rahmen** Alu

**Zubehör** keine Extras

**Modellname** MTBX13b

**Basispreis** 200 €

## B.6 Abgabe im Praktomat

Damit Ihr Tutor Ihre Abgabe korrigieren kann ist es nun notwendig diese in das Praktomat-System hochzuladen. Wenn Sie versuchen Ihre Lösung in das Praktomat-System hochzuladen wird dieses mit folgender Fehlermeldung reagieren:

```
The class containing the main method('public static void main(String[] args)') could not be found.
```

Um diese Fehlermeldung zu vermeiden fügen Sie der Klasse BikeFactory folgende Methode hinzu:

```
public static void main(String[] args) {}
```

Da dieses erste Übungsblatt sich nur mit Datenmodellierung beschäftigt müssen Sie die Bedeutung dieser Methode zu diesem Zeitpunkt noch nicht kennen. Sie werden sie jedoch schon bald kennenlernen.

Nachdem Sie die Methode hinzugefügt haben, loggen Sie sich in das Praktomat-System ein und laden Sie die Dateien über das hierfür vorgesehen Formular hoch. Vergessen Sie dabei nicht `Datentypen.txt` mit hochzuladen. Sollten sich Schwierigkeiten hierbei ergeben fragen Sie Ihren Tutor oder besuchen Sie die Poolraumbetreuung.