

Aufgabenblatt 3

Kompetenzstufe 1 & Kompetenzstufe 2

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihr IntelliJ-Projekt bis spätestens **Freitag, 30.11.2018 13:00 Uhr** in TUWEL hoch.
- Zusätzlich müssen Sie in TUWEL ankreuzen, welche Aufgaben Sie gelöst haben und während der Übung präsentieren können.
- Ihre Programme müssen kompilierbar und ausführbar sein.
- Ändern Sie bitte **nicht** die **Dateinamen** und die **vorhandene Ordnerstruktur**.
- Bei manchen Aufgaben finden Sie Zusatzfragen. Diese Zusatzfragen beziehen sich thematisch auf das erstellte Programm. Sie müssen diese Zusatzfragen für gekreuzte Aufgaben in der Übung beantworten können. Sie können die Antworten dazu als Java-Kommentare in die Dateien schreiben.
- Verwenden Sie, falls nicht anders angegeben, für alle Ausgaben `System.out.println()` bzw. `System.out.print()`.
- Verwenden Sie für die Lösung der Aufgaben keine Aufrufe (Klassen) aus der Java-API, außer diese sind ausdrücklich erlaubt, wie z.B. die Klassen `StdDraw` und `Scanner` oder Klassen, die in den Hinweisen zu den einzelnen Aufgaben aufscheinen.

In diesem Aufgabenblatt werden folgende Themen behandelt:

- Umgang mit der Klasse `Scanner`
- Implementieren von Methoden
- Überladen von Methoden
- Methoden rufen Methoden auf
- Spaghetti-Code lesen und in Methoden aufteilen

Aufgabe 1

Aufgabenstellung:

- a) Implementieren Sie eine Methode `calcProduct`, die zwei ganzzahlige Parameter entgegen nimmt, das Produkt berechnet und das Ergebnis retourniert.
- b) Implementieren Sie eine weitere Methode `calcProduct`, die zwei `double`-Parameter entgegen nimmt, das Produkt berechnet und das Ergebnis retourniert.
- c) Gegeben ist eine Methode `mult`, die zwei Parameter entgegen nimmt und das Produkt beider Parameter berechnet. Implementieren Sie nun eine weitere Methode `multNew`, die ebenfalls zwei Parameter entgegen nimmt und das Produkt berechnet (dabei aber nicht `mult` oder eine ähnliche Methode aus der Java-API benutzt, sondern `mult` nachbaut!). Einschränkung ist, dass diese Methode keinerlei Operatoren, bis auf den Zuweisungs-Operator, enthalten darf. Das heißt, dass Sie alle Anweisungen, die Berechnungen (mit Operatoren) beinhalten, in Methoden auslagern müssen. Es dürfen auch die Methoden `calcProduct` aus Punkt a) und b) nicht verwendet werden.

❗ Sie müssen bei keiner Methode auf eine spezielle Überlaufbehandlung achten.

Zusatzfrage(n):

1. Warum können in einem Programm zwei Methoden mit gleichem Namen vorkommen?
2. Was passiert, wenn Sie die Methode `calcProduct` mit einem `int`-Wert und einem `double`-Wert aufrufen?

Aufgabe 2

Aufgabenstellung:

- a) Analysieren Sie den angegebenen Spaghetti-Code und beschreiben Sie mittels Kommentaren (Kommentare im Code sind erlaubt), was die einzelnen Codeabschnitte machen.
- b) Gliedern Sie den vorgegebenen Code durch die Verwendung von Methoden. Schreiben Sie so viele Methoden wie nötig, um die Funktionalität des gegebenen Codes abzubilden. Vermeiden Sie dabei die Duplizierung von Code und überlegen Sie, ob eventuell eine Methode eine andere Methode aufrufen kann.
- c) Rufen Sie in der Methode `main` Ihre Methoden so auf, dass die Ausgabe der des Spaghetti-Codes entspricht.

Zusatzfrage(n):

1. Warum sollte man Spaghetti-Code vermeiden?
2. Warum sollte man versuchen, duplizierten Code zu vermeiden?
3. Wie sollten die Namen von Methoden gewählt werden?

Aufgabe 3

Aufgabenstellung:

- a) Implementieren Sie die Methode `drawHorizontalRect`:

```
void drawHorizontalRect(int mx, int my, int length, int width)
```

Diese Methode hat vier Parameter und zeichnet ein liegendes Rechteck (Abbildung 1a). Die Parameter `mx` und `my` beschreiben den Mittelpunkt, der Parameter `length` die Länge und der Parameter `width` die Breite des Rechtecks. Für das Zeichnen der Rechtecke dürfen die Befehle `StdDraw.rectangle(...)` und `StdDraw.filledRectangle(...)` nicht verwendet werden.

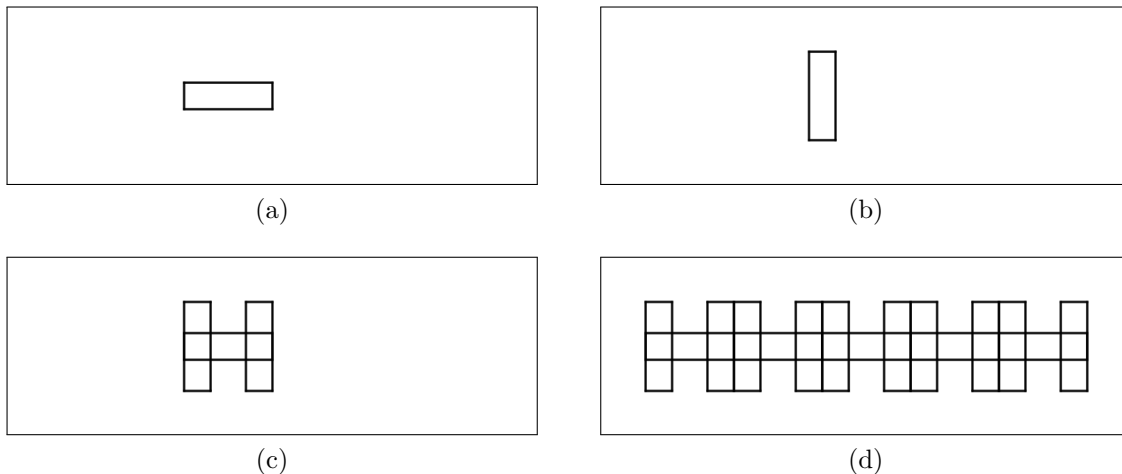


Abbildung 1: (a) Liegendes Rechteck. (b) Stehendes Rechteck. (c) **H** bestehend aus drei Rechtecken. (d) Fünf **H**-Elemente aneinander gereiht.

- b) Implementieren Sie die Methode `drawVerticalRect`:

```
void drawVerticalRect(int mx, int my, int length, int width)
```

Diese Methode zeichnet ein stehendes Rechteck (Abbildung 1b). Fassen Sie die Parameter wie bei einem liegenden Rechteck auf, zeichnen Sie aber ein stehendes Rechteck. Vermeiden Sie die Duplizierung von Code durch Aufrufen von `drawHorizontalRect`.

- c) Implementieren Sie die Methode `drawBigH`:

```
void drawBigH(int mx, int my, int length, int width)
```

Diese Methode hat vier Parameter und zeichnet ein **H**, bestehend aus drei Rechtecken (Abbildung 1c). Die Parameter `mx` und `my` beschreiben den Mittelpunkt des liegenden Rechtecks. Die Parameter `length` und `width` beschreiben die Seitenlängen des liegenden Rechtecks. Die beiden stehenden Rechtecke (hier sind die Seitenlängen vertauscht) sind so einzuzeichnen, wie in Abbildung 1c gezeigt. Verwenden Sie für die Implementierung die zuvor implementierten Methoden.

d) Implementieren Sie die Methode `drawLineOfH`:

```
void drawLineOfH(int sx, int sy, int length, int width, int numCopies)
```

Diese Methode hat fünf Parameter und zeichnet eine Reihe an **H**-Elementen (Abbildung 1d). Die Parameter `sx` und `sy` beschreiben den Mittelpunkt des ersten, ganz links liegenden Rechtecks. Die Parameter `length` und `width` beschreiben die Seitenlängen eines Rechtecks. Der Parameter `numCopies` gibt an, wie viele dieser **H**-Elemente aneinander gereiht werden. Verwenden Sie für die Implementierung die zuvor implementierten Methoden.

Aufgabe 4

Aufgabenstellung:

- Implementieren Sie das *Guessing Game* aus der Vorlesung (Vorlage auf TUWEL darf verwendet werden), modularisieren Sie es in Methoden und erweitern Sie es um grafische Komponenten.
- Es sollen folgende vier Methoden implementiert werden:
 1. Eine Methode für die Generierung der Zufallszahl. Die generierte Zufallszahl liegt zwischen -100 und 100 (jeweils inklusive).
 2. Eine Methode für das Einlesen der Spielereingaben. Verwenden Sie, wie in der Vorlesung vorgestellt, den Scanner, um die Daten von der Konsole einzulesen. Geben Sie den eingelesenen Wert zurück. Falsche Eingaben werden innerhalb der Methode behandelt.
 3. Eine Methode, die eine Nachricht (**String**) an den Spieler auf der Konsole ausgeben kann.
 4. Eine Methode, die in jeder Spielrunde die grafischen Komponenten zeichnet und in einem StdDraw-Fenster ausgibt.

Sie können weitere Methoden implementieren, falls dadurch das Programm übersichtlicher wird.

- Nachfolgend wird beschrieben, wie der Spielablauf ist, wie mit Sonderfällen umgegangen werden muss und wie die grafischen Komponenten des Spiels aussehen sollen.
 - Spielablauf: Ein Spieler gibt eine Zahl zwischen -100 und 100 auf der Konsole ein (Rateversuch). Danach wird auf der Konsole ausgegeben, ob die gesuchte Zahl größer oder kleiner ist als der Rateversuch. Hat der Spieler die gesuchte Zahl erraten, ist das Spiel beendet und der Spieler hat gewonnen. Hat der Spieler die gesuchte Zahl nach 8 Rateversuchen nicht erraten, ist das Spiel beendet und der Spieler hat verloren.
 - Sonderfälle: Falsche Eingaben wie Fließkommazahlen oder Zeichen werden ignoriert und es geht kein Rateversuch verloren. Auch die Eingabe einer Zahl kleiner -100 oder größer 100 wird ignoriert. Der Spieler wird immer mit einer Meldung auf der Konsole informiert.
 - Grafische Ausgabe: Für die grafische Ausgabe legen Sie ein Fenster der Größe 300×200 Pixel an. Wie in Abbildung 2a gezeigt, werden die verbleibenden Versuche in Form von Autoteilen dargestellt. Für jeden falschen Versuch wird ein Autoteil entfernt (Abbildung 2b). Die grafische Realisierung und Formatierung (inkl. Position) der Autoteile bleibt Ihnen überlassen. Zusätzlich soll die Information, ob man gewonnen hat, mit **You WON!!!**, wie in Abbildung 2c gezeigt, realisiert werden. Andernfalls soll der Schriftzug **You LOST!!!** (Abbildung 2d) angezeigt werden, wenn man verloren hat.

Zusatzfrage(n):

1. Wie kann man die eingegebenen Daten (bzw. deren Datentypen) unterscheiden?
2. Muss eine ungültige Eingabe aus dem Input-Stream des Scanners entfernt werden?

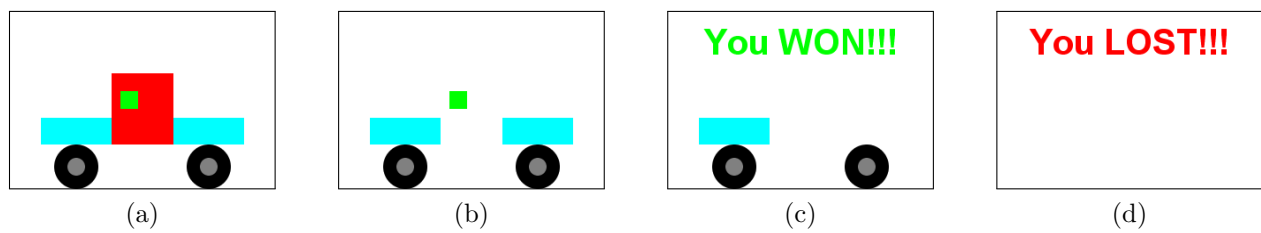


Abbildung 2: Bildausschnitte der Spielzustände.