Übungsblatt 4 (Abgabe bis Dienstag, den 12.11.2024, um 12:00 Uhr)

Prof. Dr. J. Giesl

D. Cloerkes, J.-C. Kassing, N. Lommen, E. Meyer

Allgemeine Hinweise:

- Die **Deadline** zur **Abgabe** der Hausaufgaben ist am **Dienstag, den 12.11.2024, um 12:00 Uhr**.
- Der **Workflow** sieht wie folgt aus. Die Abgabe der Hausaufgaben erfolgt **im Moodle-Lernraum** und kann nur in **Zweiergruppen** stattfinden. Dabei müssen die Abgabepartner*innen **dasselbe Tutorium** besuchen. Nutzen Sie ggf. das entsprechende **Forum** im Moodle-Lernraum, um eine*n Abgabepartner*in zu finden. Es darf **nur ein*e** Abgabepartner*in die Abgabe hochladen. Diese*r muss sowohl die **Lösung** als auch den **Quellcode** der Programmieraufgaben hochladen. Die Bepunktung wird dann von uns für **beide** Abgabepartner*innen **separat** im Lernraum eingetragen. Die Feedbackdatei ist jedoch nur dort sichtbar, wo die Abgabe hochgeladen wurde und muss innerhalb des Abgabepaars **weitergeleitet** werden.
- Die **Lösung** muss als PDF-Datei hochgeladen werden. Damit die Punkte beiden Abgabepartner*innen zugeordnet werden können, müssen **oben** auf der **ersten Seite** Ihrer Lösung die **Namen**, die **Matrikelnummern** sowie die **Nummer des Tutoriums** von **beiden** Abgabepartner*innen angegeben sein.
- Der **Quellcode** der Programmieraufgaben muss als `.zip`-Datei hochgeladen werden und **zusätzlich** in der PDF-Datei mit Ihrer Lösung enthalten sein, sodass unsere Hiwis ihn mit Feedback versehen können.

  Auf diesem Blatt muss Ihre Codeabgabe Ihren vollständigen Java-Code in Form von `.java`-Dateien enthalten. Aus dem Lernraum heruntergeladene Klassen, etwa die Datei `SimpleIO.java`, dürfen nicht mit abgegeben werden.

  Stellen Sie sicher, dass Ihr Programm von `javac` **akzeptiert** wird, wenn die entsprechenden Klassen aus dem Lernraum hinzugefügt werden. Ansonsten werden keine Punkte vergeben.
- Einige Hausaufgaben müssen im Spiel Codescape gelöst werden. Klicken Sie dazu im Lernraum rechts im Block "Codescape" auf den angegebenen Link. Diese Aufgaben werden getrennt von den anderen Hausaufgaben gewertet.

In den Aufgaben 2 und 3 sollen Sie Speicherzustände zeichnen. Angenommen wir haben folgenden Java Code:

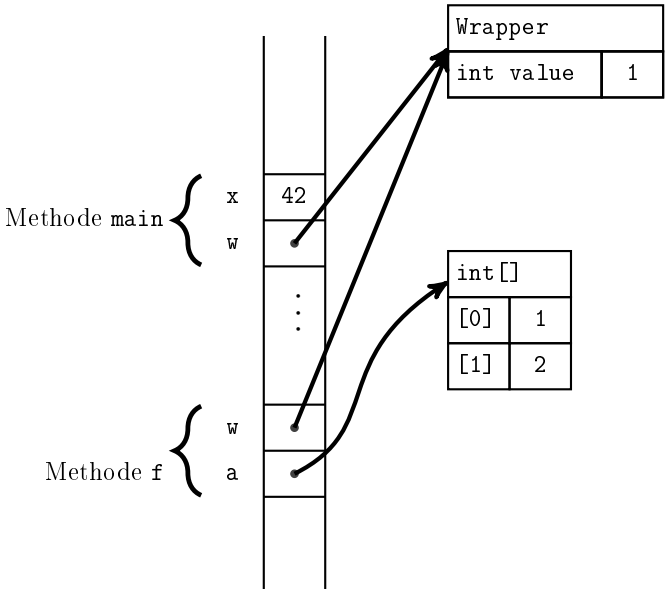
```
public class Wrapper {
      int value;
}
```

```
public class Main {
      public static void main(String[] args) {
            int x = 42;
            Wrapper w = new Wrapper();
            w.value = 0;
            f(w);
      }

      public static void f(Wrapper w) {
            int[] a = {1,2};
            w.value = 1;

            // Speicherzustand hier gezeichnet
      }
}
```

Dann sieht der Speicher an der markierten Stelle wie folgt aus:



Aufgabe 3 (Seiteneffekte): (20 Punkte)

Betrachten Sie das folgende Programm:

```
public class HSeiteneffekte {
    public static void main(String[] args) {
        Wrapper[] ws = new Wrapper[3];
        ws[0] = new Wrapper();
        ws[0].value = 42;
        ws[1] = new Wrapper();
        ws[1].value = 43;
        ws[2] = new Wrapper();
        ws[2].value = 44;
        Wrapper w = ws[1];

        int[] is = { 45, 46, 47 };
        int i = is[1];

        bar(is, i, ws, w);

        // Speicherzustand hier zeichnen
    }

    public static void bar(int[] is, int i, Wrapper[] ws, Wrapper w) {
        // Speicherzustand hier zeichnen

        i = 48;
        i = is[0];
        i = 49;
        is[1] = 50;
        is[1] = is[2];
        is[1] = 51;

        w.value = 52;
        w = ws[0];
        w.value = 53;
        ws[1].value = 54;
        ws[1] = ws[2];
        ws[1].value = 55;

        // Speicherzustand hier zeichnen
    }
}

public class Wrapper {
    int value;
}
```

Es wird nun die Methode `main` ausgeführt. Stellen Sie den Speicher an allen drei markierten Programmzuständen graphisch dar. Achten Sie darauf, dass Sie alle (implizit) im Programm vorkommenden Arrays (außer `args`) und alle Objekte sowie die zu dem Zeitpunkt existierenden Programmvariablen darstellen.

Aufgabe 6 (Programmierung mit Datenabstraktion): (1 + 3 + 2 + 7 + 3 + 4 + 10 = 30 Punkte)

In dieser Aufgabe soll eine Java-Klasse erstellt werden, mit der sich Rechtecke repräsentieren lassen. Ein solches Rechteck lässt sich mit den Koordinaten x und y für die linke obere Ecke, der Breite *width* und der Höhe *height* beschreiben, wobei x, y, *width* und *height* ganze Zahlen sind. Die Breite und die Höhe eines Rechtecks können nicht negativ sein.
Ihre Implementierung sollte mindestens die folgenden Methoden beinhalten. Sie sollten dabei die *Prinzipien der Datenkapselung* berücksichtigen. Hilfsmethoden müssen als `private` deklariert werden. In dieser Aufgabe dürfen Sie die in der Klasse `Utils` zur Verfügung gestellten Hilfsfunktionen, aber keine Bibliotheksfunktionen verwenden. Sie finden die Klasse im Lernraum.
Um Ihre Implementierung selbst zu testen, können sie in der Klasse `Rectangle` eine `main`-Methode schreiben, um verschiedene Szenarien zu überprüfen. Vergessen Sie nicht, Randfälle zu betrachten.

a) Erstellen Sie eine Klasse `Rectangle` mit den Attributen `x`, `y`, `width` und `height`.

b) Erstellen Sie die folgenden öffentlichen Methoden, um Objekte des Typs `Rectangle` erzeugen zu können. Entscheiden Sie dabei selbst, welche Methoden Sie als statisch deklarieren:

```
Rectangle(int xInput, int yInput, int widthInput, int heightInput)
Rectangle(int xInput, int yInput, int sidelengthInput)
Rectangle copy(Rectangle toCopy)
```

Beachten Sie dabei folgende Punkte:

- Der Konstruktor `Rectangle` mit vier Argumenten soll ein Rechteck erzeugen, dessen Attribute jeweils die von den entsprechenden Parametern angegebenen Werte haben.
- Der Konstruktor `Rectangle` mit drei Argumenten soll ein Quadrat erzeugen, dessen Höhe und Breite den Wert des Parameters `sidelengthInput` annehmen und dessen übrige Attribute jeweils die von den entsprechenden Parametern angegebenen Werte haben.
- Falls bei den ersten beiden Methoden eines der Argumente einen unzulässigen Wert hat, muss eine geeignete Fehlermeldung ausgegeben werden. Im Anschluss darf sich der Konstruktor beliebig verhalten. Zur Ausgabe einer Fehlermeldung kann die Methode `Utils.error(String msg)` genutzt werden.
- Die Methode `copy` soll ein Rechteck kopieren, d.h., es soll ein neues Rechteck zurückgeliefert werden, das die gleichen Attribut-Werte wie das Rechteck `toCopy` hat.

c) Erstellen Sie Selektoren, um die Koordinaten, die Breite und die Höhe eines Rechtecks setzen und auslesen zu können. Entscheiden Sie dabei selbst, welche Methoden Sie als statisch deklarieren. Falls ein Attribut auf einen unzulässigen Wert gesetzt werden soll, darf der Wert des Attributes nicht verändert werden. Stattdessen muss eine geeignete Fehlermeldung ausgegeben werden.

Hinweise:

- Um einen Fehler auszugeben, kann die Methode `Utils.error(String msg)` genutzt werden.

d) Erstellen Sie die folgenden öffentlichen Methoden. Entscheiden Sie dabei selbst, welche Methoden Sie als statisch deklarieren und begründen Sie Ihre Entscheidung kurz:

```
boolean areSquares(Rectangle... rectangles)
int area()
Rectangle intersection(Rectangle... rectangles)
```

Bei den in diesem Aufgabenteil geforderten Methoden muss der Aufrufer (und nicht Sie als Implementierer der Klasse `Rectangle`) sicherstellen, dass die Parameter, mit denen die Methoden aufgerufen werden, nicht den Wert `null` haben bzw. enthalten.

Beachten Sie dabei folgende Punkte:

- Die Methode `boolean areSquares(Rectangle... rectangles)` soll `true` zurückgegeben, falls jedes Rechteck in `rectangles` ein Quadrat ist.
- Die Methode `int area()` soll die Fläche des Rechtecks zurückgeben, auf dem sie aufgerufen wird.

- Die Methode `intersection(Rectangle... rectangles)` gibt das größte Rechteck zurück, das *vollständig* in *allen* als Argument übergebenen Rechtecken enthalten ist. Wenn `rectangles` leer ist, soll `null` zurückgegeben werden. Falls der Schnitt der Rechtecke leer ist, soll ebenfalls `null` zurückgegeben werden.
  Hinweis: Es empfiehlt sich, eine Hilfsmethode zu implementieren, die den Schnitt *zweier* Rechtecke berechnet.
  Beispiel: Wir betrachten die beiden Rechtecke, die mit den Aufrufen
  `new Rectangle(1,4,2,3)` und `new Rectangle(2,5,3,3)` erzeugt werden. Das ausgegebene Rechteck beim Aufruf von `intersection` soll ein Rechteck mit den Werten $2, 4, 1, 2$ zurückgegeben werden. Die Werte stehen jeweils in der Reihenfolge $x, y, width, height$.

  Hinweise:
  - Sie finden in der Klasse `Utils` zwei Methoden `min` und `max`, die das Minimum bzw. Maximum von beliebig vielen Werten des Typs `int` berechnen.

e) Erstellen Sie ebenfalls eine Implementierung für die öffentliche Methode

```
String toString()
```

Entscheiden Sie dabei selbst, ob Sie die Methode als statisch deklarieren. Die Methode `toString()` erstellt eine textuelle Repräsentation des aktuellen Rechtecks. Dies geschieht über die Eckpunkte des Rechtecks, die, beginnend bei der oberen linken Ecke, gegen den Uhrzeigersinn ausgegeben werden sollen.

Zum Beispiel stellt der String `(3|5),(3|1),(6|1),(6|5)` das Rechteck mit den Koordinaten 3 und 5, der Breite 3 und der Höhe 4 dar.

f) Dokumentieren Sie alle Methoden, die als `public` markiert sind, indem Sie die Implementierung mit `javadoc`-Kommentaren ergänzen. Diese Kommentare sollten eine allgemeine Erklärung der Methode sowie weitere Erklärungen jedes Parameters und des `return`-Wertes enthalten. Verwenden Sie innerhalb des Kommentars dafür die `javadoc`-Anweisungen `@param` und `@return`.

Benutzen Sie das Programm `javadoc`, um Ihre `javadoc`-Kommentare in das HTML-Format zu übersetzen. Überprüfen Sie mit einem Browser, ob das gewünschte Ergebnis generiert wurde. Falls `javadoc` Ihre Abgabe nicht kompiliert, werden **keine** Punkte vergeben.

g) Überführen Sie die Klasse `Rectangle` nun in eine `record`-Klasse. Dabei soll es in der `record`-Klasse nur den ersten Konstruktor aus Teilaufgabe b) geben.

Welche Probleme treten hierbei auf? Welche Methoden benötigen Sie nicht mehr?

Aufgabe 7 (Deck 4): (Codescape)

Lösen Sie die Missionen von Deck 4 des Codescape Spiels. Ihre Lösung für die Codescape Missionen wird nur dann für die Zulassung gezählt, wenn Sie Ihre Lösung vor der einheitlichen Codescape Deadline am Freitag, den 24.01.2025, um 23:59 Uhr abschicken.