

1. Übung zur Vorlesung Einführung in die Programmierung

Übungen: Das Übungsblatt ist jeweils ab Montag per UniWorX (uniworx.ifi.lmu.de) erhältlich. Wir gehen davon aus, dass während der Übung je 2–3 Teilnehmer gemeinsam einen Laptop mit Java und einen Editor (oder Entwicklungsumgebung) zur Verfügung haben. Die mit A gekennzeichneten Aufgaben sind explizit zur Diskussion innerhalb von kleinen Gruppen gedacht. Rufen Sie den Tutor nur, wenn Ihre Gruppe wirklich stecken bleibt!

Eigener Rechner oder CIP-Pool: Hinweise zur Installation von Java finden Sie auf der Vorlesungshomepage (www.tcs.ifi.lmu.de/lehre/ws-2017-18/eip). Bitte haben Sie Verständnis, dass wir Ihnen bei der Installation auf Ihrer eigenen Hardware keine weitere Unterstützung leisten können.

Teilnehmergruppen ohne eigenen Rechner nutzen bitte die Übung am Montag 10–12 Uhr im Rechnerraum “Gobi”; wie für alle Übungen ist eine Anmeldung bei UniWorX notwendig.

Hausübungen: Bearbeiten Sie die mit H gekennzeichneten Aufgaben alleine im Stillen. Benutzen Sie dazu die Rechner im CIP-Pools der Informatik, falls Sie keinen eigenen Rechner nutzen können. Dort sind Java, Editor und IDE bereits installiert.

Die Frist zur Abgabe Ihrer Lösungen per UniWorX finden Sie immer am Ende des Übungsblattes. Bitte beachten Sie jeweils den in der Titelzeile der Aufgabe spezifizierten **Dateinamen und das Dateiformat!** Abgaben mit anderen Dateinamen oder in anderen Formaten werden ignoriert (z.B. bitte keinen Java-Code in .pdf, .txt oder .doc).

Erzielte Hausaufgabenpunkte können Ihre Endnote verbessern, haben aber keinen Einfluss darauf, ob Sie bestehen. Aufgrund der Hausaufgabenpunkte sind Hausaufgaben ausschließlich alleine zu bearbeiten! Falls unser System Abschreiben entdeckt (dies kann auch mit großer Zeitverzögerung geschehen) kann dies zum Ausschluss von der Prüfung zu diesem Modul in diesem Semester führen! Weitere Hinweise dazu finden Sie auf der Vorlesungshomepage.

A1-1 Hallo Verfassen Sie in Ihren Lieblingseditor folgendes kleines Java-Programm:

```
public class Anfang
{
    public static void main(String[] args){
        System.out.print("Aller Anfang ");
        System.out.println("ist einfach!");
    }
}
```

Speichern Sie die Datei unter dem Dateinamen **Anfang.java**. Öffnen Sie eine Kommandozeile, und rufen Sie den Compiler `javac Anfang.java` auf, um das Programm zu übersetzen. Starten Sie das Programm mit Hilfe des Interpreters `java Anfang`. Was passiert jeweils?

A1-2 Fehler finden Finden Sie in folgendem Code, welches der Inhalt der Datei namens Fehlerprogramm.java sein soll, mindestens 5 der 7 Fehler! Beschreiben Sie jeden Fehler!

```
10    public class programm_fehlerhaft
20    {    public static main(String args)
30        // Hier findet die
40        system.out.    // Ausgabe statt
50        print(Hello World!)
60    /* Hier ist die Ausgabe schon abgeschlossen */
70    }
```

Hinweise: Die Zeilennummern gehören nicht zum Inhalt der Datei, sondern sollen Ihnen lediglich bei der Fehlerbeschreibung behilflich sein.

Die Verwendung einer Entwicklungsumgebung (Eclipse, NetBeans, IntelliJ, ...) kann diese Aufgabe für Sie weitestgehend lösen; beachten Sie jedoch, dass Ihnen in der Klausur leider keine solche IDE zur Verfügung steht, sondern nur Papier & Stift.

A1-3 Objekte & Verweise Betrachten Sie folgenden, recht häßlichen, Java-Code:

```
10    int x1 = 10; int y1 = 15;
12    int h1 = 30; int b1 = 20;
14    Rectangle packung1 = new Rectangle(x1,y1,b1,h1);
16    Rectangle packung2 = packung1;
18    System.out.println(packung2); /* 0 */
20    int x2 = x1 + 9;
22    int y2 = y1;
24    b1 = b1 * 2;
26    h1 = 40;
28    Rectangle packung3 = packung2;
30    packung2 = new Rectangle(x2,y2,b1,h1);
32    packung3.translate(1,1);
34    x1 = x1 + 14;
36    y1 = y1 - 2;
38    packung1.translate(66,70);
40    packung2.translate(-20,10);
42    packung3.translate(x1,y1);
44    System.out.println(packung1); /* 1 */
46    System.out.println(packung2); /* 2 */
48    System.out.println(packung3); /* 3 */
```

- a) Die mit 0 markierte Ausgabe druckt: "java.awt.Rectangle[x=10,y=15,width=20,height=30]"
Überlegen Sie sich ohne Hilfe eines Computers, also z.B. mit Papier und Bleistift, was die anderen drei println Aufrufe am Ende ausgeben!
- b) Zeichnen Sie analog zu Vorlesungsfolie 2-6 ein Diagramm, das den Zustand kurz vor den Textausgaben am Ende darstellt.
 - Welche Rechteck-Objekte existieren?
 - Welche Verweise gibt es, und auf welches Rechteck zeigen diese jeweils?

H1-1 Fehler finden II (3 Punkte; Abgabe: H1-1.txt oder H1-1.pdf)

Finden Sie in folgendem Code, welches der Inhalt der Datei namens `Rechnen.java` sein soll, alle echten Fehler! Beschreiben Sie jeden Fehler eindeutig und geben Sie einen Vorschlag an, wie es richtig heißen müsste. Schlagen Sie unbekannte Methoden in der Java Dokumentation nach!

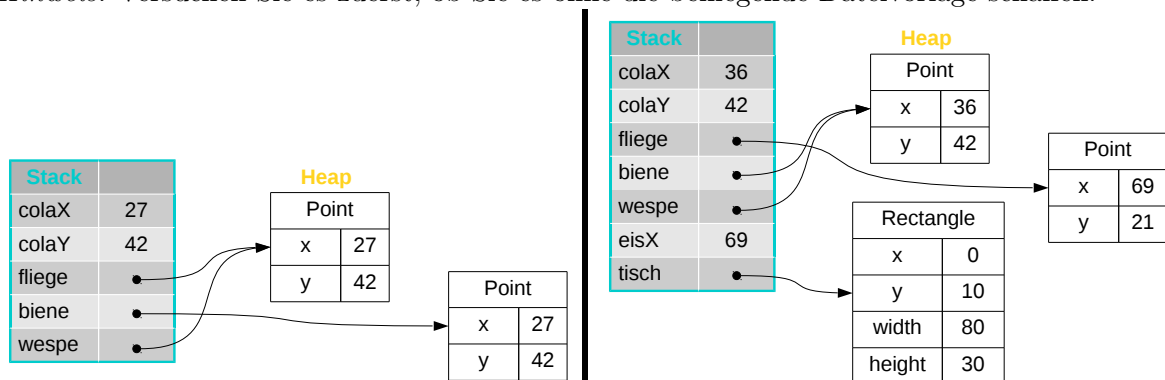
```
10  public class Rechnen {
20  public static void main( String [ ] args    ){
30      final int RADIUS    = 3 ;
40      final int HÖHE      = 10 ;
50      double kreisFläche = Math.PI * (double) RADIUS;
60      double kreisFläche = kreisFläche * RADIUS;
70      double kegelInhalt = 1/3 * kreisFläche * HÖHE;
80      double mantellinie = Math.sqrt(RADIUS * RADIUS + Math.pow(double HÖHE,2));
90      double winkel = Math.asin(RADIUS/mantellinie);
100     System.out.println("Kegelinhalt= " + kegelInhalt);
110     System.out.println("Winkel Kegelspitze= "+ winkel);
120  } }
```

Hinweise: Die Zeilennummern gehören nicht zum Inhalt der Datei, sondern sollen Ihnen lediglich bei der Fehlerbeschreibung behilflich sein.

H1-2 Objekte & Verweise II (5 Punkte; Speicherbild.java)

Dies ist die Umkehrung von Aufgabe A1-3b): Gegeben sind zwei Speicherbilder; schreiben Sie ein passendes Java Programm dazu! Ihr Programm muss zwei Kommentarzeilen `// 1. LINKES Speicherbild` und `// 2. RECHTES Speicherbild` enthalten, so dass bei Erreichen dieser Kommentarzeilen während der Ausführung das jeweilige Speicherbild passt.

Hinweis: Versuchen Sie es zuerst, ob Sie es ohne die beiliegende Dateivorlage schaffen!



Abgabe: Lösungen zu den Hausaufgaben können bis Sonntag, den 29.10.17, mit UniWorX nur als `.zip` abgegeben werden. Aufgrund des Klausurbonus müssen die Hausaufgaben von Ihnen alleine gelöst werden. Abschreiben bei den Hausaufgaben gilt als Betrug und kann zum Ausschluss von der Klausur zur Vorlesung führen. Bitte beachten Sie auch die Hinweise zum Übungsbetrieb auf der Vorlesungshomepage (www.tcs.ifi.lmu.de/lehre/ws-2017-18/eip).