

Übungsblatt 14: Software-Entwicklung 1 (WS 2010/11)

Ausgabe: in der Woche vom 07.02. bis zum 11.02.11

Abgabe: keine

Abnahme: keine

Zur Vorbereitung auf die Klausur bieten wir einige Zusatzübungen und Fragestunden an (Details auf unserer Webseite). Insbesondere findet am Montag dem **07.03.** um **15.00 Uhr** eine offizielle Fragestunde in **24-102** statt.

Aufgabe 1 Attribute und Methoden

Betrachten Sie das folgende Codestück und beantworten Sie dazu die nachfolgenden Fragen:

```
1 class A {
2     int i = 1;
3     boolean typtest() { return ( this instanceof A ); }
4     void f()          { g(); h(); }
5
6     public void g() { System.out.println("g in A"); }
7     private void h() { System.out.println("h in A"); }
8 }
9
10 class B extends A {
11     int i = 2;
12     boolean typtest() { return super.typtest(); }
13
14     public void g() { System.out.println("g in B"); }
15     protected void h() { System.out.println("h in B"); }
16 }
17
18 class C extends B {
19     int i = 3;
20
21     public void g() { System.out.println("g in C"); }
22     public void h() { System.out.println("h in C"); }
23 }
```

- Wie viele objektlokale Variablen des Namens `i` besitzt ein Objekt der Klasse `C`?
- Implementieren Sie eine Methode `sum_i` in `C`, die die Summe der in diesen objektlokalen Variablen gespeicherten Werte zurückliefert.
- Kann auch in einer Methode, die in der Klasse `B` deklariert wurde, auf jede dieser objektlokalen Variablen zugegriffen werden? Wenn ja, wie, und wenn nicht, warum nicht?
- Liefert die Methode `typtest()` der Klasse `A` stets `true`? Welchen *speziellsten* dynamischen Typ kann der implizite Parameter dieser Methode besitzen? Von welcher Klasse ist der implizite Parameter der Methode in Zeile 3 bei Ausführung von `boolean b = new B().typtest();`? Begründen Sie alle Antworten!
- Was wird jeweils ausgegeben, wenn die Methode `f` auf je einem Objekt des Typs `A`, `B` und `C` aufgerufen wird? Begründen Sie das Resultat!

Aufgabe 2 Java Collection Klassen

Neben den eigentlichen Sprachfeatures zeichnet sich Java durch eine gut ausgestattete Standardbibliothek aus, welche durch eine Vielzahl von Klassen und Schnittstellen Implementierungen vereinfacht, indem auf die zur Verfügung gestellten Bausteine mittels Vererbung, Subtyping usw. zurückgegriffen werden kann.

Die im Paket `java.util` angesiedelten *Collection*-Klassen stellen unter anderem Implementierungen für im Programmieralltag häufig auftretende Datenstrukturen bereit. Die Dokumentation zu diesem und den anderen Bestandteilen des aktuellen JDKs ist unter <http://java.sun.com/javase/6/docs/api> verfügbar.

- Betrachten Sie die Klasse `LinkedList`. Wo ist die Implementierung der Methode `containsAll` zu finden?
- Gruppieren Sie die verschiedenen Implementierungen von Datenstrukturen nach typischen Grundcharakteristika wie beispielsweise *mengenartig* oder *listenartig*. Mit welchen Java-Sprachmitteln werden diese Grundcharakteristika für den Software-Entwickler abgebildet?
- Betrachten Sie die Klasse `HashSet`. Mittels der generischen Methode `toArray` lässt sich ein Array erzeugen, welches die von der `HashSet`-Instanz verwalteten Elemente enthält. Wieso sollte die Methode nicht wie im folgenden Fragment

```
...
Set<String> set = new HashSet<String>();

set.add("A");
set.add("B");
set.add("C");
set.add("E");
set.add("X");

String[] array = set.toArray(new String[3]);
...
```

verwendet werden? Wie sieht ein sinnvoller Aufruf der Methode `toArray` aus?

- Beschreiben Sie die eventuell vorhandenen Beziehungen zwischen `Collection`, `HashSet`, `List`, `Vector` und `Stack` durch Anfertigung eines Klassendiagramms.
- Erläutern Sie, weshalb die Beziehung zwischen der Klasse `Stack` und der Klasse `Vector` zumindest sehr unglücklich gewählt ist.

Aufgabe 3 Grafikeditor 2

In dieser Aufgabe werden wir den Grafikeditor von Übungsblatt 12 weiterentwickeln. Der Editor soll jetzt nicht mehr nur Polygone sondern verschiedene Figuren verwalten und darstellen können. Folgende Figuren sollen möglich sein: Rechteck, Quadrat, Kreis und Beschriftungen und natürlich weiterhin Polygone.

- Entwerfen Sie eine Klassenhierarchie für verschiedene Arten von Figuren. Verwenden Sie eine abstrakte Klasse `Figur` als gemeinsame Superklasse. Verändern Sie ihr Klassendiagramm von Übungsblatt 12 Aufgabe 4, so dass es nun die neuen Klassen und ihre Beziehungen enthält.
- Implementieren Sie die neuen und ändern Sie Ihre bestehenden Klassen, um die neuen Figuren eingeben und darstellen zu können.
- Erweitern Sie den Editor um die Funktion *Figur duplizieren*, die die aktuell im Fokus stehende Figur dupliziert. Beide Figuren (das Original und das Duplikat) sollen anschließend unabhängig voneinander verschoben werden können.