Ludwig-Maximilians-Universität München
Institut für Informatik
PD Dr. Peer Kröger
Andreas Züfle, Johannes Niedermayer

WS 2012/13
01.02.2013

Klausur zur Vorlesung Einführung in die Programmierung

Vorname:	
Name:	
Geb.-Datum:	
Matr.-Nr.:	
Platznummer:	

Die Klausur besteht aus 6 Aufgaben. Die Punktzahl ist bei jeder Aufgabe angegeben. Bitte überprüfen Sie, ob Sie ein vollständiges Exemplar erhalten haben.

Tragen Sie die Lösungen in den dafür vorgesehenen Raum im Anschluss an jede Aufgabe ein. Falls der Platz für Ihre Lösung nicht ausreicht, benutzen Sie bitte nur die ausgeteilten Zusatzblätter!
Tragen Sie bitte oben auf jeder ungeraden Seite Ihren Namen und Ihre Matrikelnummer ein.

Verwenden Sie keinen Rot-, Grün- oder Bleistift!

Aufgabe	mögliche Punkte	erreichte Punkte
1. Allgemeine Fragen	14	
2. Algorithmen	12	
3. Vollständige Induktion	6	
4. Modellierung	8	
5. UML und Java	7	
6. Hoare-Kalkül	8	
Summe:	55	
Note:		

Entwertung der Klausur
Meine Klausur soll nicht korrigiert und nicht gewertet werden.

München, den 01.02.2013 **Unterschrift:** _____

Aufgabe 1 Allgemeine Fragen (2+3+3+2+2+2 Punkte)
Allgemeine Fragen

(a) Wandeln Sie die Zahl 1324_5 in 5-adischer Zahlendarstellung in Dezimaldarstellung um. Bei dieser Aufgabe muss der Lösungsweg erkenntlich seine. Ein Ergebnis ohne Lösungsweg gibt keine Punkte.

(b) Gegeben sei die folgende Syntaxdefinition in Backus-Naur-Form für natürliche Zahlen in Binärdarstellung.

```
<Binaerzahl> ::= 0 | <NichtNullDarstellung>
<NichtNullDarstellung> ::= 1 <Binaerzahl>*
```

Welche der folgenden Zeichenreihen sind mit obiger Syntaxdefinition herleitbar? Begründen Sie Ihre Aussage, indem Sie entweder eine korrekte Herleitung angeben (und dabei keine Schritte überspringen) oder indem Sie begründen warum die Zeichenreihe nicht herleitbar ist.

- 1010

- 200

- 0111

(c) Gegeben seien drei (sechsseitige) Würfel. Aufgabe ist es, diese Würfel so auf einem Tisch anzuordnen, dass alle Würfel die selbe Augenzahl zeigen. Dafür werde folgender Algorithmus vorgeschlagen.

```
(1) Wirf alle drei Würfel auf den Tisch
(2) Falls nicht alle Würfel die selbe Augenzahl zeigen,
    nimm die Würfel auf und beginne bei (1)
```

Entscheiden Sie, welche der folgenden Eigenschaften dieser Algorithmus erfüllt. Begründen Sie Ihre Entscheidung kurz.

- Terminierend

- Deterministisch

- Korrekt

- Partiell korrekt

(d) Gegeben ist folgender Programmcode:

```
int i = 1; int j = 5;

System.out.println( ++i ); // Ausgabe:
System.out.println( j-- ); // Ausgabe:

System.out.println( i ); // Ausgabe:
System.out.println( j ); // Ausgabe:
```

Ergänzen Sie die zu erwartenden Ausgabewerte (Eine Begründung ist nicht nötig).

(e) Gegeben sei der folgende Java-Code:

```
public static void swap(int[] a,int[] b){
  int[] tmp = a;
  a = b;
  b = tmp;
  b[0] = 42;
}
```

der Code wird folgendermaßen aufgerufen:

```
public static voic main(String[] args){
  int[] x = new int[]{0,1};
  int[] y = new int[]{2,3};
  swap(x,y);
  //*)
}
```

Was ist der Inhalt der Variablen `x` und `y` nach Ausführung des Codes $swap(a, b)$ an Position `//*)` der `main`-Methode? Warum?

(f) Nennen Sie die zwei Möglichkeiten zur Übersetzung von generischen Typen und erklären Sie diese.

Aufgabe 2 Algorithmen
Algorithmen (2+3+3+4 Punkte)

(a) Implementieren Sie eine statische Methode `boolean isQuadratzahl(int zahl)` in Java, die den Wahrheitswert `true` zurückgibt, falls `zahl` eine Quadratzahl ist. Die Funktion überprüft also das mathematische Prädikat $\exists i \in \mathbb{N} : i * i = zahl$. Sie können annehmen, dass `zahl` positiv ist.

(b) Implementieren Sie eine statische Methode `void invertieren(int[] werte)` in Java, die ein Integer-Array als Argument erhält und dieses invertiert. Aus dem Array $[0, 1, 2, 3]$ soll also das Array $[3, 2, 1, 0]$ werden. Beachten Sie, dass die Methode kein Array zurückgeben soll. Sie können davon ausgehen, dass das Array eine Länge von mindestens 1 hat.

(c) Implementieren Sie eine statische Methode `boolean isPalindrom(int[] werte)` in Java, die ein Integer-Array als Argument erhält und `true` zurückgibt, wenn es sich bei diesem Array um ein Palindrom handelt. Ansonsten soll `false` zurückgegeben werden. Sie können davon ausgehen, dass das Array eine Länge von mindestens 1 hat.

Bemerkung: Ein Palindrom ist normalerweise eine Zeichenkette, die von vorn und von hinten gelesen gleich bleibt. In unserem Kontext sind die Zeichen die Integer-Werte des Eingabe-Arrays. So ist zum Beispiel das Array $[2, 3, 1, 3, 2]$ und auch das Array $[1, 4, 4, 1]$ ein Palindrom, das Array $[1, 6, 3, 4, 2]$ und das Array $[1, 2, 3, 1]$ nicht.

(d) Implementieren Sie eine statische Methode `int[] interleave(int[] werte1, int[] werte2)` in Java, die zwei Integer-Arrays `werte1` und `werte2` mit der gleichen Länge so in einem neu erstellten Array einsortiert, dass abwechselnd je ein Wert aus `werte1` und `werte2` ins Ergebnisarray einsortiert wird. Fangen Sie den Fall unerlaubter Argumente, also zweier Arrays unterschiedlicher Länge, mit den in der Vorlesung gelernten Techniken ab. Sie können davon ausgehen, dass beide Arrays eine Länge von mindestens 1 haben.

Beispielsweise werden die Arrays $[1, 2, 3, 4]$ und $[5, 6, 7, 8]$ zu dem Array $[1, 5, 2, 6, 3, 7, 4, 8]$ zusammengefügt, das dann als Ergebnisarray zurückgegeben wird.

Aufgabe 3 Beweis durch vollständige Induktion (6 Punkte)

Vollständige Induktion

Zeigen Sie mit Hilfe der vollständigen Induktion, dass folgende Aussage für alle $n \in \mathbb{N}$ gilt:

$$n^2 = \sum_{i=1}^{n} (2 \cdot i - 1)$$

Aufgabe 4 Objektorientierte Modellierung (8 Punkte)

Modellierung

Gegeben sei die folgende Klasse `Punkt2D`, die einen zweidimensionalen Punkt modelliert. Die Klasse stellt die Funktion `void verschiebe(double x, double y)`, die Funktion `void skaliere(double fac)` und eine Funktion `double euklidischeDistanz(Punkt2D p2)` zur Berechnung der euklidischen Distanz zwischen zwei Punkten zur Verfügung.

```
/**Die Klasse Punkt2D dient zur Modellierung von
  *Punkten im zweidimensionalen kartesischen Raum*/
public class Punkt2D {
  private double x;
  private double y;


  /**Erzeugt einen zweidimensionalen Punkt
   * @param x Die x-Koordinate des erzeugten Punktes
   * @param y Die y-Koordinate des erzeugten Punktes*/
  public Punkt2D(double x, double y){
          this.x = x;
          this.y = y;
  }

  /**Verschiebt den Punkt entlang der Koordinatenachsen.
   * @param x Distanz, um die der Punkt entlang der
   *          x-Koordinate verschoben werden soll
   * @param y Distanz, um die der Punkt entlang der
   *          y-Koordinate verschoben werden soll*/
public void verschiebe(double x, double y){
        this.x += x;
        this.y += y;
}

/**Berechnet die euklidische Distanz zwischen
 * diesem Punkt und dem Punkt p. Die euklidische Distanz
 * d(p1,p2) ist definiert als d(p1,p2)= sqrt((p1.x-p2.x)^2+(p1.y-p2.y)^2).
 * @param p Punkt, zu dem die Distanz zu diesem
 *          Punkt berechnet werden soll.*/
public double euklidischeDistanz(Punkt2D p){
        double dx = p.x-x;
        double dy = p.y-y;
        return Math.sqrt(dx*dx+dy*dy);
}

  /**Skaliert den Punkt um den Faktor fac. Durch die Skalierung
   * wird der Punkt näher zum Koordinatenursprung (falls -1 < fac < 1) hin
   * bzw weiter vom Ursprung weg bewegt (fac > 1 oder fac < -1). Bei fac = 1
   * verändert sich der Punkt nicht. Ist fac < 0 wird der Punkt
   * am Koordinatenursprung gespiegelt.*/
  public void skaliere(double fac){
          this.x *= fac;
          this.y *= fac;
  }
}
```

Implementieren Sie eine Klasse `Kreis` in Java, die einen Kreis über einen Mittelpunkt und einen nichtnegativen Radius modelliert. Zwischen der Klasse `Kreis` und `Punkt2D` soll eine Verwendungsbeziehung bestehen.

keine `isa`-Beziehung. Die Klasse `Kreis` soll die folgenden Methoden zur Verfügung stellen:

- Einen Konstruktor, der einen `Punk2D` als Mittelpunkt sowie eine Zahl vom Typ `double` als Radius erhält. Stellen Sie sicher, dass `radius` nur nichtnegative Werte annimmt. Sollte ein ungültiger Parameter `radius` übergeben werden, werfen Sie eine `IllegalArgumentException` (diese erbt von der Klasse `RuntimeException`).

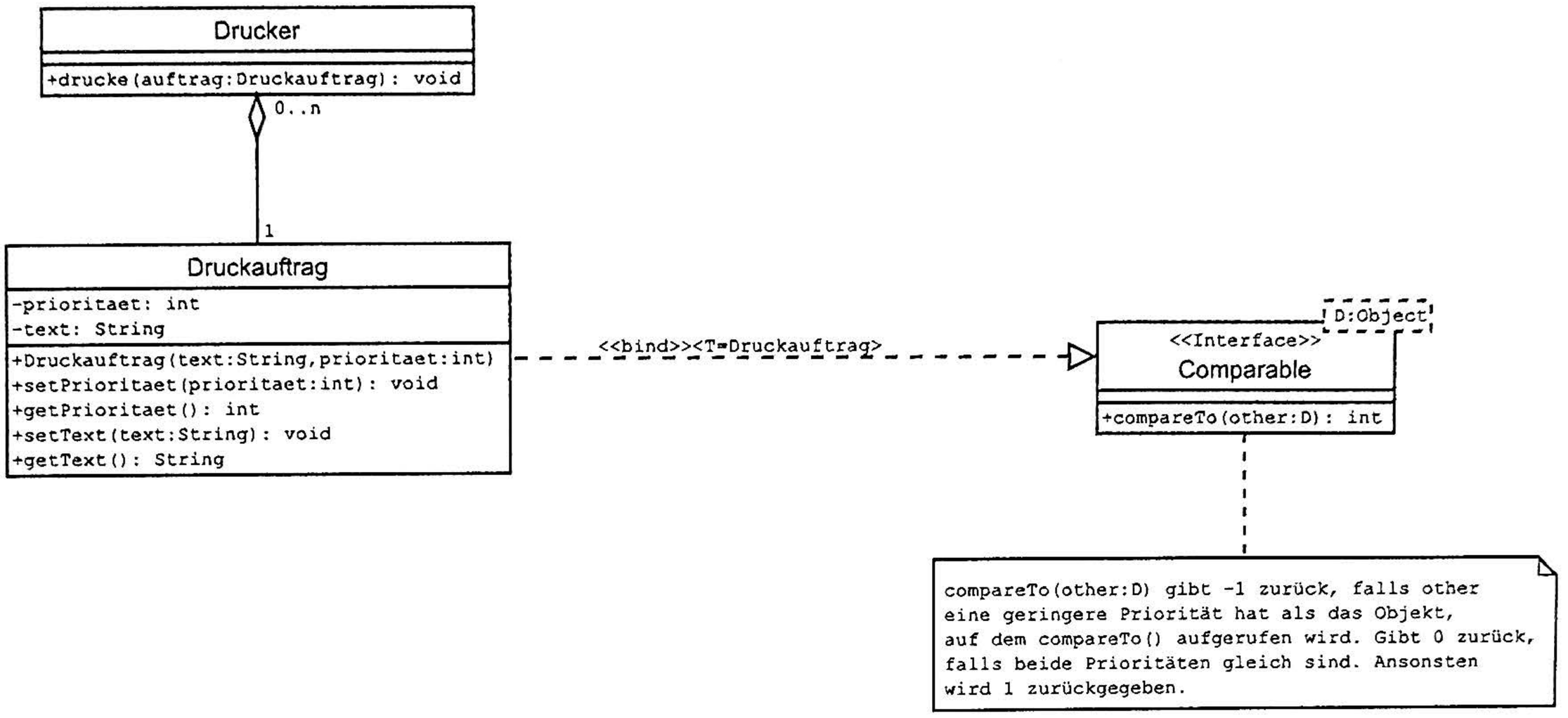
- Eine Methode `void verschiebe(double x, double y)`, die den Kreis um den Wert x entlang der x-Achse sowie um den Wert y entlang der y-Achse verschiebt.

- Eine Methode `void skalieren(double fac)`, die den Kreis um den Faktor fac um den Ursprung skaliert. Durch die Skalierung wird der Kreis näher zum Koordinatenursprung (falls $-1 < fac < 1$) bzw weiter vom Ursprung wegbewegt ($fac > 1$ oder $fac < -1$). Bei $fac = 1$ verändert sich der Kreis nicht. Ist $fac < 0$, wird der Kreis am Mittelpunkt gespiegelt. Beachten Sie auch, dass die Größe des Kreises bei einer Skalierung angepasst werden muss.

- Eine Methode `boolean schneidet(Kreis k)`, die `true` zurückgibt wenn sich die beiden Kreise schneiden, wenn also die beiden Kreise mindestens einen Punkt gemeinsam haben.

Aufgabe 5 Objektorientierter Entwurf *(7 Punkte)*
UML und Java

In dieser Aufgabe soll ein Drucker zum drucken von `Strings` modelliert werden. Ein Drucker enthält 0 bis n Druckaufträge, die er anhand ihrer Priorität sortiert und in der entsprechenden Reihenfolge druckt. Druckaufträge enthalten neben der entsprechenden Priorität auch den zu druckenden Text in Form eines `Strings`. Die Zusammenhänge wurden im folgenden UML-Klassendiagramm zusammengefasst.



Implementieren Sie `Druckauftrag.java` und `Comparable.java` anhand der aus dem Klassendiagramm ersichtlichen Vorgaben. Beachten Sie auch eine sinnvolle Implementierung der Methoden!

Aufgabe 6 Korrektheitsbeweis **Hoare-Kalkül** *(8 Punkte)*

Beweisen Sie mit den Mitteln des Hoare-Kalkülls die partielle Korrektheit des folgenden Programmstücks. Dabei seien `x`, `k`, `result` Variablen vom Typ `int`.

```
// Vorbedingung: x >= 1
result = x;
k = 1;
while(k < x) {
  result = result+k;
  k = k+1;
}
```

// Nachbedingung: result $= \sum_{i=1}^{x} i$

Hinweis: Ein Teil der Invariante lautet: $result = x + \sum_{i=1}^{k-1} i$