SS2021, Klausur Programmierung 1, Mittwoch, 07.07.2021

Es sind keine Hilfsmittel außer dem Schreibgerät und einer lokalen Installation der JavaDoc-Dokumentation zum JDK zugelassen.

Es sind keine Unterlagen zugelassen.

Es sind ausschliesslich die Antwortbögen als Extrapapier zugelassen.
Auf jedem Antwortbogen sind Name, Matrikelnummer, Seitenzahl und Aufgabennummer anzugeben

Es ist anzugeben, wenn ein letzter Prüfungsversuch vorliegt.

Bearbeitungszeit: 80 min

Leistungsbewertung:

Maximal sind 95 Punkte zuzüglich 5 Bonuspunkte erreichbar.

Die Klausur ist ab 47 Punkten bestanden.

Aufgabe	F	1	2	3	4	5	6	Übung	Gesamt	LV-Note
Punkte max:	5	17	6	11	11	22	18	10	100	XXXXXX
Punkte										

Weitere Hinweise:
1) Kommunikation jeder Art ist nur mit dem Dozenten gestattet.
2) Lassen Sie auf allen Blätter links, rechts, oben und unten mindestens je 2 cm Rand
3) Lassen Sie auf allen Blätter die Rückseite frei.
4) Die Lösungen sind programmiertechnisch sinnvoll, verständlich und den jeweiligen Konventionen entsprechend auszuführen.
5) Es sind die in der Lehrveranstaltung eingeführten Fachbegriffe zu verwenden.
6) Falls Sie Teile Ihrer Lösung auf Extrablätter schreiben:
   * Verwenden Sie für jede Aufgabe ein eigenes Extrablatt.
   * Beschriften Sie alle Extrablätter nur einseitig -> die Rückseite ist freizulassen.
   * Auf allen Extrablätter lassen Sie links, rechts, oben und unten mindestens je 2 cm Rand und schreiben Sie in die obere rechte Ecke Ihren Namen und Ihre Matrikelnummer.

Aufgabe F (AbgabeForm: 5 Bonuspunkte):

Abgabe der Lösungen nur handgeschreiben auf den bereitgestellten Klausurbögen, ausgedruckt auf weißen Papier, mit ausgefülltem Formularfeld. Die Klausurbögen nur einseitig beschreiben. Rückseite freilassen.

Für jede Aufgabe einen eigenen Klausurbogen verwenden – Teilaufgaben können auf den gleichen Klausurbogen.

Vor der Abgabe bitte darauf achten, daß der Scan:

verzerrungsfrei ist,

gutem Kontrast hat (nicht dunkelgrau auf hellgrau, keine Schatten, keine Ränder durch Schlagschatten)

angemessene Auflösung hat, so daß Details gut lesbar, aber die Datei nicht zu groß ist (Richtwert: Auflösung von 150 dpi). Die eingescannte Klausur ist im Format pdf abzugeben.

Aufgabe 1 (17 Punkte):

Gegeben sei: Ein Sortierkasten hat einzelne Kästchen mit den Bezeichnungen Z14 bis T07, die folgendermaßen angeordnet sind:

```
Z14  Y14  X14  W14  V14  U14  T14
Z13  Y13  X13  W13  V13  U13  T13
Z12  Y12  X12  W12  V12  U12  T12
Z11  Y11  X11  W11  V11  U11  T11
Z10  Y10  X10  W10  V10  U10  T10
Z09  Y09  X09  W09  V09  U09  T09
Z08  Y08  X08  W08  V08  U08  T08
Z07  Y07  X07  W07  V07  U07  T07
```

Schreiben die folgenden Methoden, die alle Teil der selben Klasse sind.
Vermeiden Sie unnötige Codewiederholungen, beachten Sie DRY.

1.a) [4 Punkte]
```
/** Gibt die Bezeichnungen der Kästchen eines Sortierkastens (Z14 – T07) auf der Konsole aus: */
public static void printSortierkasten()
```

1.b) [4 Punkte]
```
/** Gibt die Bezeichnungen der Kästchen eines Sortierkastens (Z14 – T07) aus: */
public static void printSortierkasten(PrintWriter out)
```

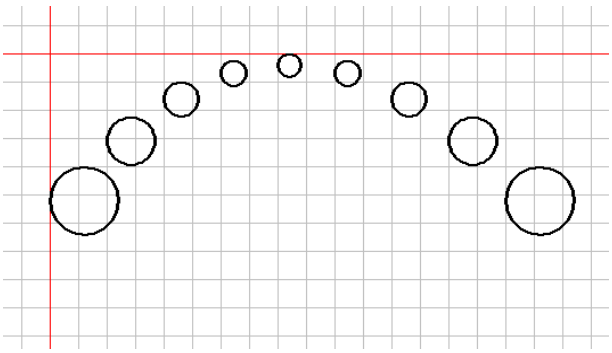
1.c) [9 Punkte]
```
/** Erzeugt einen String, der die Bezeichnungen der Kästchen eines Sortierkastens (Z14 – T07) enthält (wie in der Abbildung oben). */
public static String sortierkastenAsString()
```

Aufgabe 2 (6 Punkte): Geben Sie die vollständige Deklaration für die folgende Methode an:

```
/**  Der Rückgabewert gibt an, ob drei aufeinanderfolgende Zahlen a, b, c der Größe nach
*    absteigend angeordnet sind. Bsp.:
* b = isOrderedDescending(5,  3, -8); // b wird true zugewiesen
* m = isOrderedDescending(3,  3,  1); // m wird false zugewiesen
* k = isOrderedDescending(2, -1,  0); // k wird false zugewiesen
*/
public static boolean isOrderedDescending(int a, int b, int c)
```

Aufgabe 3 (12 Punkte):
Schreiben Sie die Methode `public void drawCircles(Graphics g)`. Die Methode soll mithilfe des Graphics-Objekts mehrere **blaue** Kreise zeichnen.

Ein von drawCircles(..) gezeichneter Bildschirmbereich sieht *[mit zusätzlichem 25-Pixel-Hilfsraster und Koordinatenachsen – das nicht zur Aufgbe gehört]* so aus:



Für die Koordinaten (xe, ye) der linken oberen Ecken der umfassenden Rechtecke der Kreise gilt:

xe hat für die aufeinanderfolgenden Kreise die Werte 0, 50, 100, 150, etc.
im Bereich 0 <= xe <= 400

ye(xe) = $(200 – xe)^2 / 400$.

Die Radius r eines Kreises hängt von seiner Position xe nach der Formel
r(xe) = ½ * (20 + 0,001 * $(xe -200)^2$) ab.

Aufgabe 4 (11 Punkte):
Beantworten Sie die Fragen **<u>auf einem Extablatt unter Angabe der Teilaufgabe</u>**

Hinweise:

- Bei der Lösung dieser Aufgabe kann Ihnen die JavaDoc-Dokumentation zum JDK helfen.

- Angeben ist im Aufgabentext jeweils nur der einfache Klassenname (ohne Packagename)

a) [2] Geben Sie die Anzahl der Objektmethoden der Klasse Math an.

b.1) [1] Geben Sie an, ob es zulässig ist, ein Objekt des Typs Segment an den Konstruktor von StringBuilder zu übergeben.

(mögliche Antworten: ja / nein)

b.2) [2] Geben Sie eine kurze Begründung an.

c.1) [1] Kann der Standardkonstruktor von Graphics innerhalb der main-Methode von pr1.k01.GTest erfolgreich aufgerufen werden?

(mögliche Antworten: ja / nein)

c.2) [2] Falls f.1 ja: Geben Sie eine korrekte Java-Anweisung als Beispiel an.
Falls f.1 nein: Geben Sie eine kurze Begründung an

d.1) [1] Hat die Klasse java.awt.Color (einen oder mehrere) statische Fatorymethoden?

(mögliche Antworten: ja / nein)

d.2) [2] Falls f.1 ja: Geben Sie einen vollständigen Methodenheader an
Falls f.1 nein: Geben Sie an, welche Eigenschaft einer statischen Factorymethode nicht zu finden ist.

Aufgabe 5 (22 Punkte):

Vermeiden Sie unnötige Codewiederholungen, beachten Sie DRY.

Schreiben Sie eine **Klasse Baum** mit den Attributen höhe und radius (der Krone) sowie den folgenden Eigenschaften:

a) Die Klasse soll einen vollständigen Konstruktor und einen Kopierkonstruktor haben.

b) Für alle Attribute soll es Getter-Methoden (aber keine Setter-Methoden) geben.

c) Als Service für Forstbetriebe hat Baum die folgende Methode:
*/** Gibt die Menge des Holzes in Festmetern (FM) zurück, die nach der Formel*

$holzmenge = \frac{\pi}{5} \cdot d^2 \cdot h$ *berechnet wird. Dabei ist die Dicke d des Stamms* $d = \frac{(h + 2{,}3 \cdot r)}{20}$

Das Formelzeichen h ist die höhe, dasFormelzeichen r der radius des Baums.
**/*

```
public double holzmenge()
```

d) Die Methode **`public String toSizeString()`** für die Textdarstellung der Größe des Baums gibt einen formatierten String zurück, dessen Format dem Beispiel `17.3 m,  3.2 m` entspricht *(im Beispiel: ein Baum mit Höhe 17,34 und Radius 3,22)*.

e) Die Methode für die standardmäßige Textdarstellung gibt einen formatierten String zurück, dessen Format dem Beispiel `17.3 m,  3.2 m, 19.25 FM` entspricht *(im Beispiel: ein Baum mit Höhe 17,34 und Radius 3,22, der eine holzmenge von 19,25 Festmetern hat)*.

Aufgabe 6 (18 Punkte):

Vermeiden Sie unnötige Codewiederholungen, beachten Sie DRY.

Schreiben Sie die **Klasse Obstbaum** als Subklasse der Klasse Baum (vgl.: Aufgabe 5).

a) Die Sorte des Obstes wird als Zeichenkette im Attribut sorte gespeichert. Für die Sorte gibt es eine Gettermethode.

b) Die Klasse Obstbaum hat einen vollständigen Konstruktor.

c) Der Obstbaum wird nicht zur Holzverarbeitung genutzt, seine Holzmenge wird deshalb als 0 angegeben.

d) Die Klasse Obstbaum hat eine Methode, die den Ernteertrag an Obst in kg angibt:
*/** Gibt den* Ernteertrag an Obst in kg zurück. Der Ernteertrag ist:
$ertrag = der\ größere\ der\ beiden\ Werte\ 0\ und\ (9{,}76 \cdot r^3 - 8{,}7)$ *. */*

```
public double ertrag()
```

e) Die standardmäßige Textdarstellung soll dem Format des Beispiels (Apfel-Obstbaum mit Höhe 4,32 und Radius 2,24) `4.3 m,  2.3 m,   123 kg Apfel` entsprechen.