

Aufgabe 6 Bearbeitung bis Samstag, 27.11.2010, 18:00 Uhr
Abgabe bis zum Beginn der Übung, die auf das Bearbeitungsdatum folgt

Verwenden einfacher Klassen (III) , Verwenden von Graphics

Aufgabe 6.a: Untersuchung von Klassen der Java Standardbibliothek

Betrachten Sie für die folgende Untersuchung die Dokumentation der folgenden Klassen:

java.lang.Math
java.lang.String
java.util.Collections
java.util.ArrayList
java.util.Arrays
java.awt.Color
java.awt.Dimension
java.awt.Graphics
java.util.Scanner

Geben Sie jeweils an, wie viele Konstruktoren und wieviele Methoden es in der Klasse gibt. Notieren Sie das Ergebnis in Form einer Tabelle.

Geben Sie jeweils an, wie hoch der Anteil von Klassenmethoden an den Methoden in der Klasse insgesamt ist (falls der Anteil nicht 0% oder 100% ist, reicht ein Schätzwert).

Ordnen Sie die Klassen der Klassenart MVK (Methodenverbundklasse) oder BK (Bauplanklasse) zu.

Geben Sie auch für alle Klassen die Anzahl der gefundenen Konstruktoren an. Gibt es Klassen, bei denen die Einordnung schlecht möglich ist?

Versuchen Sie für zwei der Methodenverbundklassen jeweils ein Objekt zu erzeugen (nach dem Schema `XKlasse o = new XKlasse();`). Geht das? Falls nein: Erhalten Sie eine Fehlermeldung? Falls ja: Welche? Notieren Sie die Fehlermeldung.

Schreiben Sie die Klasse `pr1.a06.SimpleClassInspection`. Erzeugen Sie in `main()`, soweit es möglich und sinnvoll ist, von jeder der o.g. Klassen ein Objekt und weisen sie es einer Variablen zu. Geben für jede Klasse ein Beispiel für den Aufruf einer typischen Methode aus der Klasse an.

Aufgabe 6.b: Die Klasse java.awt.Graphics

Stöbern Sie ein wenig in der Dokumentation zur Klasse Graphics. Schauen Sie nach, wie man eine Linie zeichnet, einen Text ausgibt, die Zeichenfarbe festlegt und einen gefüllten Kreis zeichnet (Hinweis: Graphics kennt keine Kreise, sondern nur Ovale: Kreise sind Ovale mit gleich großer Breite und Höhe des umfassenden Rechtecks).

Hinweis: Das Koordinatensystem, das Graphics verwendet, hat seinen Ursprung in der linken oberen Ecke des Zeichenbereichs. Die x-Achse ist nach rechts gerichtet. Die y-Achse ist nach unten (nicht wie gewohnt nach oben) gerichtet.

Aufgabe 6.c: Schreiben Sie ihre ersten Bauplanklassen

Um etwas zeichnen zu können, benötigt man ein Graphics-Objekt. Weil es nicht einfach ist, ein Graphics-Objekt selbst zu erzeugen, benutzen wir ein Graphics-Objekt, das in der Klasse DirtyPainter erzeugt wird.

Installieren Sie zunächst die Bibliothek lernhilfe-dirtypainter2010ws.jar in ihrem Eclipse-Projekt.

Testen Sie die den Bibliotheksimport mit:

```
public class Test {  
    public static void main(String[] args) {  
        DirtyPainter.main(null);  
    }  
}
```

Schreiben Sie im Package pr1.a06.draw drei Klassen, die jeweils eine Instanzmethode public void draw(Graphics g) enthalten (*Beachte: kein static, also keine Klassenmethode!*). In der draw-Methode soll etwas beliebiges Buntes (Rechtecke, Kreise, etc.) gezeichnet werden.

Hinweis (Einfachstbeispiel):

```
public class DrawDemo {  
    public void draw(Graphics g) {  
        g.drawLine(11, 22, 333, 444);  
    }  
}
```

Erzeugen Sie in einer Testklasse jeweils ein Objekt von pr1.qad.DirtyPainter und von jeder der selbst geschriebenen Klassen (die nur im Beispiel unten Drawable1, FederMasseSystem und Hochhaus heissen – verwenden Sie eigene Klassen, die die Methode draw implementieren) und lassen Sie sich die Objekte von Painter zeichnen!

Ein Beispiel mit den fiktiven Klassen Drawable1, FederMasseSystem und Hochhaus:

```
public class Test {  
    public static void main(String[] args) {  
        // Zeichnen eigener Objekte  
        DirtyPainter painter = new DirtyPainter();  
        painter.add(new Drawable1());  
        painter.add(new FederMasseSystem());  
        painter.add(new Hochhaus());  
        painter.showDrawing();  
    }  
}
```