# Richtlinien und Hinweise für die OOVS-Übungsaufgaben

- Jeder Student bekommt ein Projektverzeichnis unter /proj/i4oovs/<loginname>.
- In diesem Projektverzeichnis muss für jede Übungsaufgabe ein Unterverzeichnis angelegt werden, z.B. für die fünfte Aufgabe mit:
  `mkdir /proj/i4oods/`*<loginname>*`/aufgabe5`
- Die Aufgaben müssen mit dem *abgabe*-Programm abgegeben werden. Um z.B. die fünfte Aufgabe abzugeben:
  `/proj/i4oovs/pub/abgabe aufgabe5`
- Aufgaben mit dem Hinweis “0 Punkte” müssen nicht abgegeben werden.
- Die Lösungen müssen eigenständig erstellt worden sein. Verstöße werden geahndet.
- Die *Java Coding Guidelines* müssen befolgt werden (Abweichung führt zu Punktabzug!):
    - Klassennamen beginnen mit einem *Großbuchstaben*.
    - Methodennamen beginnen mit einem *Kleinbuchstaben*.
    - Variablennamen beginnen mit einem *Kleinbuchstaben*.
    - Konstante (final static) Variablen bestehen nur aus *Großbuchstaben*.
    - Variablenamen sind ausdrucksfähig und beschreiben deren Zweck.
- Wenn eine bestimmte Programmstruktur in der Aufgabenstellung verlangt wird, muss die Lösung diese einhalten. Die Lösung muss
    - Namen von Klassen und Packages
    - Sichtbarkeit von Variablen und Methoden
    - Namen, Parametertypen und Rückgabewerte von Methoden

  wie in der Aufgabenstellung beschrieben verwenden.
- Es werden Punkte abgezogen, falls
    - verlangte Funktionalität nicht implementiert wird
    - nicht verlangte Funktionalität implementiert wird
    - das Programm nicht kompiliert.
- Der Code muss lesbar und nachvollziehbar sein. Falls komplizierte Teile vorkommen, sollten diese mit Kommentaren erläutert werden.
- Bei Problemen mit der Aufgabenstellung könnt ihr euch jederzeit an uns wenden:
  *{felser, rrkapitz, reiser, wawi}@informatik.uni-erlangen.de*

## *Übungsaufgabe #1: Whiteboard - Erste Schritte* *0 Punkte* 15.4.2002

Zweck dieser Aufgabe ist die Einführung in die Programmiersprache Java und deren Entwicklungsumgebung. Es wird das Grundgerüst für eine Whiteboard-Applikation erstellt, das in den folgenden Aufgaben zu einem interaktiven, mehrbenutzerfähigen Tool ausgebaut werden soll.
In dieser ersten Aufgaben sollen Klassen für mehrere Grafikobjekte erstellt werden, sowie ein einfacher Grafikframe implementiert werden, auf dem diese Objekte ausgegeben werden können. Die Teilaufgaben a) bis e) werden die Entwicklung einer Lösung schrittweise begleiten.

a) Erstelle ein Interface `Shape`, das die gemeinsamen Methoden aller Graphikobjekte definiert. In Hinblick auf spätere Aufgaben sind mindestens folgende Methoden vorzusehen:
   - `void move(int x, int y)`, um das Objekts nach (x,y) zu bewegen.
   - `int getX()` und `int getY()`, um die Position des Objekts zu ermitteln.
   - `boolean isInside(int x, int y)` testet ob (x,y) innerhalb des Objekts liegt.
   - `void draw(Graphics g)`, um das Objekt auszugeben.

b) Erstelle zwei Klassen `Circle` und `Rect`, die beide das Interface `Shape` implementieren. Die Positionskoordinaten beschreiben die obere linke Ecke des Objektes. Ein Konstruktor soll die Initialisierung mit Positionskoordinaten sowie der Größe, die sich beim Kreis durch den Durchmesser, beim Rechteck durch die Breite und Höhe charakterisieren lässt, erlauben. Für die Methode `isInside` schließt "innerhalb des Objekts" dessen Rand mit ein.

c) Die Funktionalität der Klassen `Circle` und `Rect` kann durch die in `/proj/i4oovs/pub/aufgabe1` bereitgestellten Unit-Test überprüft werden. Um die Tests ausführen zu können muss der `CLASSPATH` um das JAR-Archiv `junit.jar` erweitert werden (z. B. `setenv CLASSPATH ${CLASSPATH}:/local/junit/junit.jar`). Außerdem müssen die Test-Klassen in das Arbeitsverzeichnis kopiert werden. Die Tests können dann wie normale Java-Anwendungen gestartet werden (z. B. `java RectTest`).

d) Schreibe eine Applikation `Whiteboard`, die folgendes macht:
   Beim Starten der Applikation soll eine Frame erzeugt werden, der eine `Whiteboard`-Instanz enthält. Eine `Whiteboard`-Instanz soll bis zu 100 Objekte des Types `Shape` in einem Array speichern können. In der `paint`-Methode des `Whiteboard` sollen alle gespeicherten Objekte über deren `draw`-Methoden ausgegeben werden.
   Zum Testen sollen zunächst statisch einige Kreise und Rechtecke bei der Initialisierung angelegt werden.

e) Nun soll der Anwender die Möglichkeit bekommen, die ausgegebenen Graphikobjekte selbst zu bestimmen. Hierzu sollen der Klassenname des Shape und die (x,y)-Position auf der Kommandozeile als separate Parameter angegeben werden. Ein Aufruf des Whiteboard könnte also beispielsweise so aussehen:
   ```
   java WhiteBoard Circle 20 30 Rect 15 50 Circle 150 60
   ```
   Die angegebenen Klassennamen müssen das Interface `Shape` implementieren. Das Whiteboard erzeugt eine Instanz von dieser Klasse, fügt diese zu seinem Shape-Array hinzu, und bewegt dann das `Shape` an die in der Kommandozeile angegebene Position. Jedes Graphikobjekt sollte hierzu eine Standardgröße größer als 0 haben.
   Es ist vorzusehen, dass zur Laufzeit weitere Shape-Klassen bereitgestellt werden können, die man automatisch (ohne Modifikation von WhiteBoard) verwenden kann. Aus den Namen sind also dynamisch Instanzen zu erzeugen. Hierzu muss man sich folgendes überlegen:
   - Wie erzeugt man ein Klassenobjekt aus einem Klassennamen? Wie erzeugt man eine Instanz aus einem Klassenobjekt?
   - Welche Art von Konstruktor ist hier notwendig?