

# Übung 09: Vererbung und Dynamische Bindung

Abgabetermin: 12. 01. 2010

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Gruppe: G1 (Prähofer)

G2 (Wolfinger)

G3 (Wolfinger)

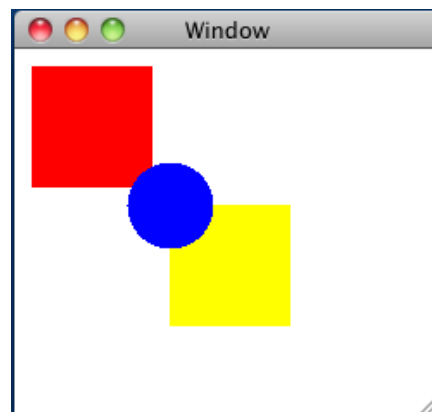
G4 (Jahn)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 09.1	12	<input type="checkbox"/>	Java-Programm Testergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 09.2	12 +6 Bonus	<input type="checkbox"/>	Java-Programm Testergebnisse	Java-Programm	<input type="checkbox"/>	

## Aufgabe 09.1: Zeichenprogramm (Restrukturierung OOP)

Implementieren Sie das in Aufgabe 07.1 beschriebene Zeichenprogramm als objektorientiertes Programm. Nehmen Sie Ihre frühere Implementierung und machen Sie daraus ein objektorientiertes Programm.

```
Figure: [L]ine, [R]ect, [C]ircle: R
Location (x, y): 10 10
Width: 70
Height: 70
Color (R G B): 255 0 0
Figure: [L]ine, [R]ect, [C]ircle: R
Location (x, y): 90 90
Width: 70
Height: 70
Color (R G B): 255 255 0
Figure: [L]ine, [R]ect, [C]ircle: C
Location (x, y): 90 90
Radius: 25
Color (R G B): 0 0 255
Figure: [L]ine, [R]ect, [C]ircle .
All Figures:
- Rectangle (10 10) - 70 | 70 - RGB=(255 0 0))
- Rectangle (90 90) - 70 | 70 - RGB=(255 255 0))
- Circle (90 90) - 25 - RGB=(0 0 255)
```

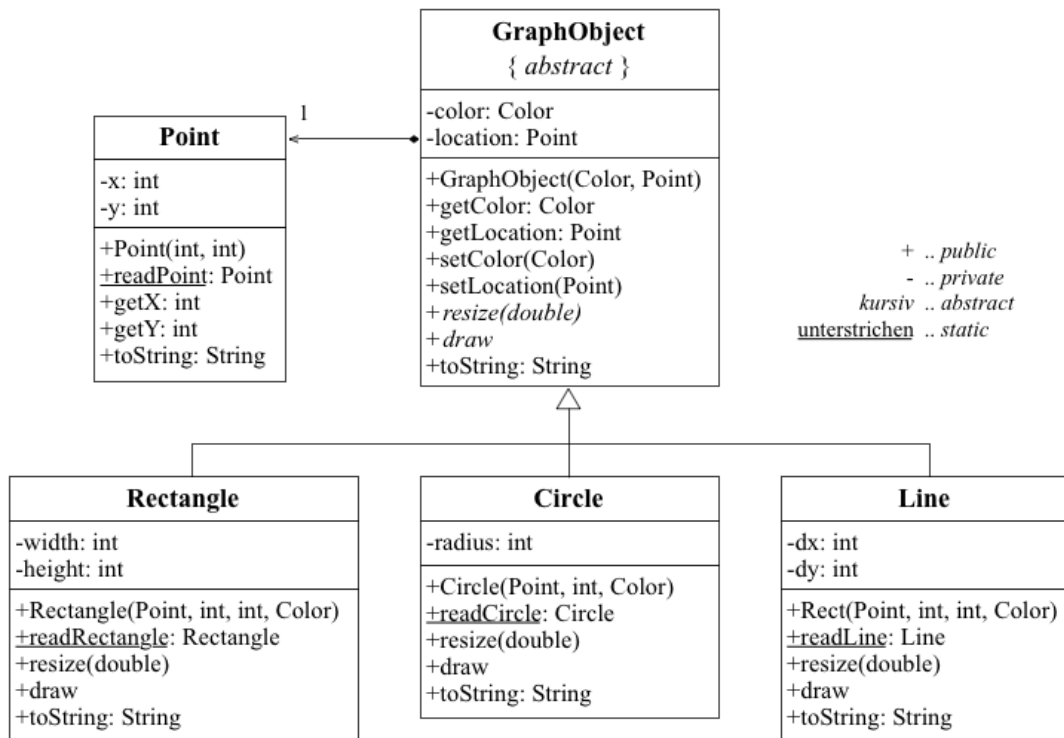


Führen Sie eine abstrakte Basisklasse *GraphObject* ein und leiten Sie Unterklassen *Rectangle*, *Circle* und *Line* ab. Ergänzen Sie für alle graphischen Objekte eine Farbe und verwenden Sie zum Zeichnen die Methoden für ausgefüllte Objekte, zB *fillRectangle* statt *drawRectangle*. Orientieren Sie Ihre Programmstruktur am Klassendiagramm und führen Sie bei Bedarf weitere Methoden ein.

Implementieren Sie für die Klassen *Rectangle*, *Circle* und *Line*:

- einen Konstruktor, zB initialisiert *Circle(Point p, int radius, Color c)* ein Objekt der Klasse *Circle*
- eine statische *read*-Methode, die mit Hilfe der *In*-Klasse ein Objekt einliest; zB liest *readCircle* ein *Circle*-Objekt
- eine *toString*-Methode, die den Inhalt eines Objekts als Text liefert; zB liefert *toString* der Klasse *Circle* eine Zeichenkette für ein *Circle*-Objekt  

```
Circle (100 100) - 80 - RGB=(0 0 255)
```
- eine *draw*-Methode, die mit Hilfe der *Window*-Klasse die Figur zeichnet
- eine *resize*-Methode, um die Grösse der Figur zu ändern; ein Faktor größer 1 vergrößert die Figur, ein Faktor kleiner 1 verkleinert die Figur



Überarbeiten Sie auch Ihr Hauptprogramm. Dort wo sie bisher drei Arrays vom Typ *Line[]*, *Rectangle[]* und *Circle[]* verwendet haben um die Figuren zu speichern, verwenden Sie ein einziges Array vom Typ *GraphObject[]*.

Hinweis: Die Klasse *Window.java* können Sie von der LVA-Homepage herunterladen. Importieren Sie die *Color*-Klasse mit `import java.awt.Color` am Beginn Ihres Programms.

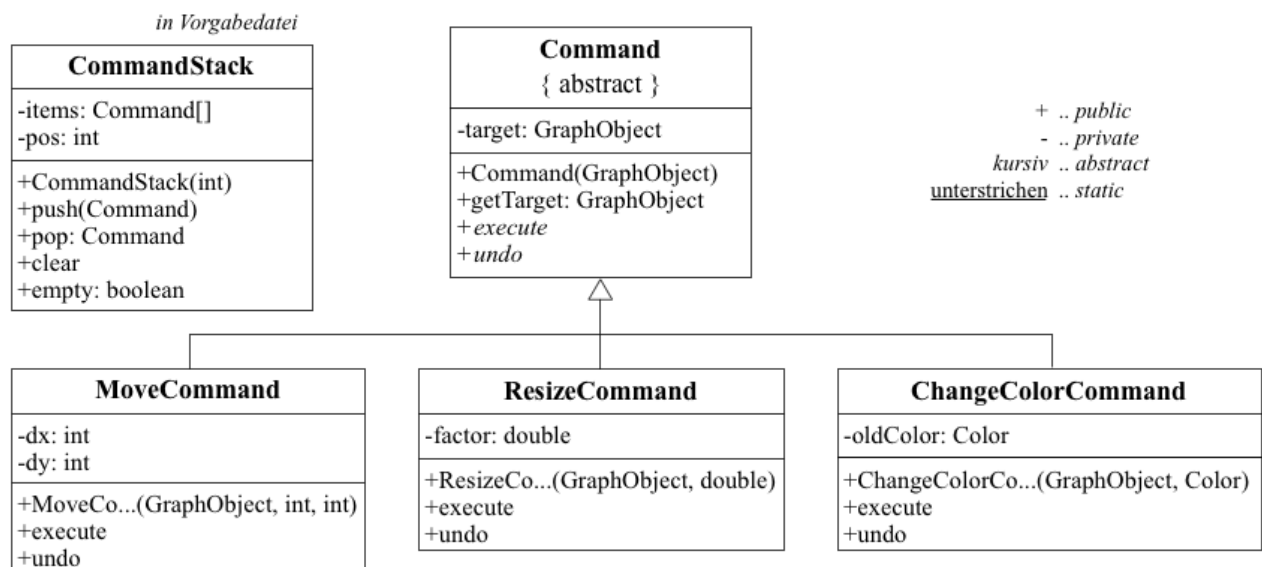
Abzugeben sind: Java-Programm, Testergebnisse

## Aufgabe 09.2: Zeichen-Kommandos mit Undo/Redo-Funktion

Erweitern Sie das Zeichenprogramm aus Aufgabe 1. Man soll grafische Objekte verschieben (M=Move), deren Größe ändern (S=Resize) oder deren Farbe ändern (O=Change Color) können. Verpacken Sie dabei die Operationen in Kommando-Objekte wie im Klassendiagramm dargestellt.

Implementieren Sie für die Klassen *MoveCommand*, *ResizeCommand* und *ChangeColorCommand*:

- einen Konstruktor, zB initialisiert *MoveCommand(GraphObject obj, int dx, int dy)* ein Objekt der Klasse *MoveCommand*
- eine *execute*-Methode, um die Änderung durchzuführen



Freiwillige Zusatzaufgabe (+6 Bonuspunkte): Implementieren Sie eine Undo-/Redo-Funktion für die Operationen. Speichern Sie dazu alle Operationen auf einem Undo-Stack vom Typ *CommandStack* (in der Vorgabedatei). Bei Undo holen Sie die letzte Operation vom Undo-Stack und rufen deren *undo()* Methode auf. Legen Sie diese Operation dann auf den Redo-Stack. Bei Redo holen Sie die letzte Operation vom Redo-Stack und rufen deren *execute()* Methode auf und legen Sie diese Operation wieder auf den Undo-Stack. Wird eine neue Operation eingegeben, löschen Sie den Redo-Stack.

Folgender Beispieldialog zeigt wie die Operationen in das Zeichenprogramm integriert werden sollen. Der einzugebende Index spezifiziert die Position eines graphischen Objekts im Array aus Aufgabe 1. Zeichnen Sie nach jeder Operation alle Figuren neu.

```
Figure: [L]ine, [R]ect, [C]ircle  
Command: [M]ove, Re[s]ize, Change C[o]lor, [U]ndo,  
Re[d]o: S  
Index: 2  
Faktor: 2.0  
Command: [M]ove, Re[s]ize, Change C[o]lor, [U]ndo,  
Re[d]o: O  
Index: 2  
Color (R G B): 0 255 0
```

Abzugeben sind: Java-Programm, Testergebnisse

