

---

**Holger Fübler**A5, 6, Raum B 219  
68131 Mannheim  
Telefon: (0621) 181-2605  
Email: fuessler@informatik.uni-mannheim.de**Robert Schiele**B6, 29, Raum C0.04  
68131 Mannheim  
Telefon: (0621) 181-2214  
Email: rschiele@uni-mannheim.de

---

**Praktische Informatik I**  
**Wintersemester 2005/2006****5. Übungsblatt**  
**Abgabe: 30. November 2005**

---

---

**Allgemeine Bearbeitungshinweise**

---

Verwenden Sie für die Programmieraufgaben die in den Aufgaben angegebenen Klassennamen und geben Sie den Klassen-Quellcode sowohl auf Papier als auch per e-mail bei Ihrem Tutor ab. Verwenden Sie Javadoc-Kommentare zur Dokumentation.

---

**Aufgabe 1****6 Punkte**

---

Implementieren Sie eine Klasse `MeineMatrix`, welche ein zweidimensionales Integer-Feld enthält.

Die Klasse soll folgendes Interface implementieren:

```
1 public interface Matrix {  
    void setzeWert(int zeile, int spalte, int wert);  
    int liesWert(int zeile, int spalte);  
    void tauscheZeilenSpalten();  
}
```

Hierbei soll `setzeWert` ein Element der Matrix auf den übergebenen Wert setzen, `liesWert` diesen Wert zurückgeben und `tauscheZeilenSpalten` die Zeilen und Spalten tauschen, also die Matrix an der Diagonalen spiegeln.

Welche Eigenschaft muss für das zweidimensionale Integer-Feld hierfür gelten?

Was muss Ihre Klasse außerdem noch enthalten, damit sie nutzbar ist?

Implementieren Sie ein kleines Testprogramm, welches Ihre Klasse testet.

Implementieren Sie eine Klasse `ExtendedString`, die von der Klasse `MyString` erbt. Die neue Klasse soll zusätzlich zur geerbten Funktionalität die Möglichkeit haben, den repräsentierten String in ROT13/ROT5-Verschleierung (siehe Blatt 3) auszugeben.

```
package pi1.blatt5.util;

public class MyString {
    protected String value;

    public MyString(String value) { this.value = value; }

    public MyString() { }

    public String getValue(){ return value; }

    public void setValue(String value){ this.value = value; }
}
```

Aufgabe 2 a)

3 Punkte

Erzeugen Sie die Klasse `pi1.blatt5.ExtendedString` als Erweiterung (=Subklasse) von `pi1.blatt5.util.MyString`. Beachten Sie dabei, das neue Paket zu deklarieren. Implementieren Sie die bekannte ROT13+ROT5-Methode als Klassenmethode, die einen String übernimmt und den verschleierte String zurückgibt. Die Methode soll nur für Klassen des packages `pi1.blatt5` ausführbar sein. Die ganze Klasse jedoch soll öffentlich zugreifbar sein.

Aufgabe 2 b)

4 Punkte

Versehen Sie die Klasse `ExtendedString` mit einem Konstruktor, der einen `String` (die init-Zeichenkette) und einen `boolean` übernimmt. Der bool'sche Wert soll dabei angeben, ob der Initial-String im Klartext vorliegt (`true`) oder verschleiert ist (`false`) istt. Intern soll die Zeichenkette immer im Klartext gespeichert werden. Außerdem soll ein Konstruktor ohne Argumente und einer mit nur einem String implementiert werden (hierbei soll angenommen werden, dass der Text dann im Klartext vorliegt).

Aufgabe 2 c)

3 Punkte

Schreiben Sie eine (öffentliche) Instanzmethode `getShiftedString()`, die den ROT13+ROT5 geshifteten String zurückgibt. Ferner soll eine Methode `setValue(String value, boolean isCleartext)` geschrieben werden, die wie der entsprechende Konstruktor arbeitet.

## Aufgabe 2 d)

3 Punkte

Schreiben Sie eine main-Methode, die ein `ExtendedString`-Objekt vom ersten Kommandozeilenargument erzeugt. Das zweite Kommandozeilenargument ist optional und gibt mit `true|false` an, ob das erste Argument im Klartext (`true`) vorliegt oder ob es sich verschleierten Text (`false`) handelt. Geben Sie nach der Erzeugung des Objektes den Rückgabewert von `getShiftedString()` aus.

---

 Aufgabe 3
 

---

 $\Sigma = 11$  Punkte

Im Folgenden sollen Sie ein Java-Package `geo` erstellen, welches die Klassen `Dreieck`, `Flaeche`, `GleichseitigesDreieck`, `GleichseitigesSechseck`, `Kreis`, `Quadrat`, `Rechteck`, `RechtwinkligesDreieck`, `Sechseck`, `Vieleck` und `Viereck` enthält. Die genannten Klassen stellen die durch ihren Namen bezeichnete geometrische Figur dar.

## Aufgabe 3 a)

2 Punkte

Stellen Sie die Beziehungen zwischen den obigen Objekten im Sinner “Ist-Ein” Beziehung grafisch dar, indem Sie einen hierarchischen Baum zeichnen, in dem jede Klasse durch ein Kästchen mit den jeweiligen Namen gezeichnet wird und dann die einzelnen Kästchen durch einen Pfeil mit ihrer nächstliegenden allgemeineren Form verbunden werden.

Beispiel: Ein gleichseitiges Dreieck ist eine Spezialform eines Dreiecks, also zeichnen Sie einen Pfeil von `GleichseitigesDreieck` auf `Dreieck`. Obwohl ein gleichseitiges Dreieck auch eine Spezialform einer Fläche ist, zeichnen Sie *keinen* direkten Pfeil von `GleichseitigesDreieck` auf `Flaeche`, da das Dreieck die naheliegendere (speziellere) Verallgemeinerung ist.

## Aufgabe 3 b)

7 Punkte

Implementieren Sie alle oben genannten Klassen in dem Java-Package `geo`. Benutzen Sie hierzu das Konzept der Vererbung gemäß der Zeichnung aus der vorigen Teilaufgabe. Die Klasse `Flaeche` ist bereits vorgegeben:

```

1 package geo;
  public abstract class Flaeche {
    public abstract double flaecheninhalt();
    public abstract double umfang();
  }

```

Alle anderen Klassen sollen *nicht* abstrakt sein. Die Klassen sollen alle einen Konstruktor mit Parametern, welche die geometrische Form sinnvoll beschreiben, besitzen. Ein Quadrat kann zum Beispiel durch seine Seitenlänge sinnvoll beschrieben werden, wohingegen man bei einem allgemeinen Vieleck nicht um die Angabe der Koordinaten aller Eckpunkte herumkommt. Außerdem sollen alle Klassen die geerbten abstrakten Methoden implementieren. Der Umfang sollte von jeder Klasse korrekt zurückgege-

ben werden. Der Flächeninhalt muss nur für die Klassen `GleichseitigesDreieck`, `GleichseitigesSechseck`, `Kreis`, `Quadrat`, `Rechteck`, `RechtwinkligesDreieck` und `Sechseck` den korrekten Wert zurückliefern. Für die anderen Klassen darf wegen der Komplexität der Berechnung stattdessen der Wert 0 zurückgegeben werden.

Aufgabe 3 c)

2 Punkte

Schreiben Sie ein kleines Testprogramm, welches die Eigenschaften der Vererbung und der Polymorphie zeigt, indem Sie ein Objekt vom Typ `Quadrat` einer Variablen vom Typ `Flaeche` zuweisen und auf dieser Variablen dann die beiden vorhandenen Methoden ausführen und deren Ergebnis auf dem Bildschirm ausgeben. Das Testprogramm soll sich *nicht* im Java-Package `geo` befinden.