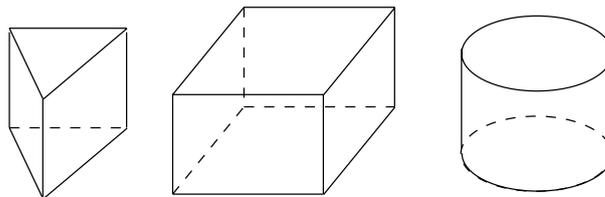


**Abgabe:** Mo, 7.01.08 zentrale Abgabe über <https://grundstudium.in.tum.de/info1abgabe>

## Praktikum Grundlagen der Programmierung

### Aufgabe 40 (Ü) Vererbung

*Prismen* oder *Zylinder* sind geometrische Körper, die durch Parallelverschiebung ihrer Grundfläche im Raum entstehen. Die folgende Abbildung zeigt als Beispiele für Prismen ein Dreiecksprisma, einen Quader und einen Zylinder, die durch Verschiebung eines Dreiecks, eines Rechteckes bzw. eines Kreises entstehen:



**Würfel** sind durch ihre Höhe eindeutig bestimmt.

**Quader** sind durch ihre Höhe, Länge und Breite bestimmt.

**Kreiszylinder** sind durch ihre Höhe und den Radius ihrer Grundfläche bestimmt.

**regelmässige Vielecksprismen** sind durch ihre Höhe und die Seitenlänge ihrer jeweiligen Grundfläche bestimmt. In Frage kommen zum Beispiel

- *regelmässige Dreiecksprismen*;
- *quadratische Quader* oder
- *regelmässige n-Ecksprismen*.

An Operationen sollen die Körper die Berechnung von Umfang und Flächeninhalt ihrer *Grundfläche*, die Berechnung ihrer *Mantelfläche*, *Oberfläche* und des *Volumens* zur Verfügung stellen, sowie den *Vergleich* ihrer Volumina mit anderen geometrischen Körpern unterstützen.

In dieser Aufgabe soll eine Klassenhierarchie für diese geometrischen Körper in UML modelliert und in Java implementiert werden.

- Geben Sie ein Klassendiagramm zur Modellierung der oben genannten Körper in UML an.
- An welcher Stelle Ihrer Klassenhierarchie müssen die in a) spezifizierten Operationen implementiert werden, damit möglichst viel in Unterklassen wiederverwendet werden kann?
- Greifen Sie 3 geometrische Körper aus Ihrem Klassenmodell heraus und implementieren Sie diese in Java. Schreiben Sie zudem ein kleines Testprogramm.

### **Hinweise:**

- Fassen Sie gleichartige Attribute und Methoden in einer geeigneten Oberklasse zusammen.
- Die Fläche eines gleichseitigen Dreiecks mit Seitenlänge  $a$  ist  $\frac{a^2}{4}\sqrt{3}$
- Die Fläche eines regelmässigen Sechsecks mit Seitenlänge  $a$  ist  $\frac{3a^2}{2}\sqrt{3}$ .
- Die Java-Klasse `Math` stellt in der Klassenvariablen `Math.PI` einen Wert für  $\pi$  sowie eine Klassenmethode `Math.sqrt()` zur Berechnung der Quadratwurzel zur Verfügung.

## Aufgabe 41 (Ü)    Überladen vs. Überschreiben

Betrachten Sie folgenden Klassendeklarationen:

```
public class A {
    public void f() {
        System.out.println("bin_Methode_f_in_Klasse_A");
    }
    public void g() {
        System.out.println("bin_Methode_g_in_Klasse_A");
    }
}
public class B extends A {
    public void f() {
        System.out.println("bin_Methode_f_in_Klasse_B");
    }
}
public class C {
    private A a;
    public C(A a) {
        this.a = a;
    }
    public void g(){
        System.out.println("bin_Methode_g_in_Klasse_C");
        a.f();
    }
}
public class D {
    public void f(A a) {
        a.f();
        System.out.println("bin_Methode_f_in_Klasse_D_mit_Typ_A");
    }
    public void f(B b) {
        b.f();
        System.out.println("bin_Methode_f_in_Klasse_D_mit_Typ_B");
    }
}
public static void main (String[] s){
    A a0 = new A();
    a0.f();
    a0.g();
    A a1 = new B();
    a1.f();
    a1.g();

    C c0 = new C(a0);
    c0.g();
    C c1 = new C(a1);
    c1.g();

    D d = new D();
    d.f(a0);
    d.f(a1);
    d.f(new B());
}
```

Welche Ausgaben erwarten Sie für die jeweiligen Methodenaufrufe? Begründen Sie Ihre Antwort.



**Aufgabe 43 (H) Autowettrennen**

**(15 Punkte)**

Ziel dieser Aufgabe ist es, ein Autowettrennen mit verschiedenen Fahrzeugtypen in unterschiedlicher Geländelage objektorientiert zu modellieren und implementieren.

- a) In der abstrakten Basisklasse `Fahrzeug` sollen die für alle Fahrzeugtypen gemeinsamen Attribute und Methoden festgehalten werden. Ein Fahrzeug hat die folgenden allgemeinen Merkmale:
- aktuelle Geschwindigkeit und Maximalgeschwindigkeit in km/h
  - aktuellen Tankfüllstand und Tankkapazität in Liter
  - eine Methode `void volltanken()`, die das Fahrzeug bis zur maximalen Füllmenge auftankt
  - eine Methode `int bewaeltigeStrecke(Strecke s)`, die für jedes Fahrzeug den Tankfüllstand und die aktuelle Geschwindigkeit entsprechend Streckentyp und Resttreibstoff anpasst. Die Methode soll die Zeit in Minuten zurückgeben, die das Fahrzeug zur Bewältigung der Strecke `s` benötigt. Falls das Fahrzeug die Strecke nicht mit dem Resttreibstoff und minimalem Spritverbrauch bewältigen kann, so soll `-1` zurückgegeben werden.

Implementieren Sie die konkreten Fahrzeugtypen: `Gelaende-`, `Rallye-`, `Rennwagen` mit nachfolgenden konkreten Eigenschaften.

Fahrzeugtyp	maxGeschwindigkeit	bergig	sandig	eben	Tankkapazität	Verbrauch pro 100km
Gelaendewagen	180km/h	97%	80%	100%	150.5 l	$\max(13, (\frac{v}{38})^2) l$
Rallyewagen	200km/h	75%	100%	95%	100.5 l	$\max(15, (\frac{v}{36})^2) l$
Rennwagen	250km/h	60%	40%	100%	55.0 l	$\max(10, (\frac{v}{35})^2) l$

Hierbei bezeichnet  $v$  die aktuelle Geschwindigkeit des Fahrzeugs. Um eine Strecke `s` zurückzulegen, muss zusätzlich noch der Spritverbrauch mit einberechnet werden. Falls die aktuelle Strecke nicht mit der maximal für diese Strecke möglichen Geschwindigkeit und der aktuellen Tankfüllung bewältigt werden kann, muss eine kleinere Geschwindigkeit gewählt werden. Dies ist in der Methode `int bewaeltigeStrecke(Strecke s)` zu berücksichtigen. **Nutzen Sie bei Ihrer Implementierung Vererbung möglichst gut aus.**

- b) Zur Modellierung der Beschaffenheit einer zu bewältigenden Strecke, sollen Sie die folgenden Geländetypen der Teilstrecken implementieren:
- `BergigeStrecke`
  - `EbeneStrecke`
  - `SandigeStrecke`

Jede Teilstrecke ist mit einer `laenge` (in km) versehen.

- c) Implementieren Sie für die eigentliche Simulation des Autowettrennens die Klasse `Wettrennen`, welches auf einer bestimmten Autorennstrecke abgehalten wird. An diesem Wettrennen soll je ein Fahrzeug pro Fahrzeugtyp teilnehmen. Setzen Sie dazu in der `main`-Methode die Autorennstrecke, als Liste von o.g. Teilstrecken fest. Nach Bewältigung jeder Teilstrecke muss das Fahrzeug vollgetankt werden, wenn es die nächste Teilstrecke mit aktueller Tankfüllung und minimalem Spritverbrauch nicht mehr bewältigen kann. Wägen Sie vor Bewältigung der Teilstrecke ab, ob ein Volltanken oder ein Fahren mit reduzierter Geschwindigkeit zu einem besseren Zeitergebnis für die Teilstrecke führt. Ein Volltanken des Fahrzeugs nimmt zusätzlich Zeit in Anspruch: 1 Minute pro 10 zu tankender Liter. Geben Sie nach der Bewältigung jedes Streckenabschnittes die aktuelle Geschwindigkeit und die für den aktuellen Streckenabschnitt benötigte Zeit des Fahrzeugs aus. Am Ende des Rennens soll der Sieger des Autowettrennens – der für die gesamte Autorennstrecke die wenigste Zeit gebraucht hat – ermittelt und ausgegeben werden.