

## Modellierung und Programmierung 1 – Übungsserie 3

Abgabetermin: 06.12.2022, 23:00 Uhr  
Abgabeformat: pdf für die UML, zip für die Programme  
Max. Punkte: 52

### Erweiterung, Interfaces, Exceptions, Aggregation

#### 1. Exception `GeometricObjectException` (8 Punkte)

- a) Modellieren Sie eine Klasse `GeometricObjectException` mit folgenden Eigenschaften
- `GeometricObjectException` erweitert die Exception `InvalidParameterException`
  - Ein Konstruktor ohne Parameter
  - Ein Konstruktor mit folgenden Eigenschaften
    - Parameter: Zeichenkette `message`
    - Verhalten: Aufruf des Konstruktors der Basisklasse mit der Nachricht "GeometricObject: " gefolgt von `message`
  - Hinweis: Geben Sie bei der Basisklasse ausschliesslich alle relevanten Methoden an.
- b) Implementieren Sie das Modell aus Aufgabenteil a)

#### 2. Klasse `Wuerfel` (11 Punkte)

- a) Modellieren Sie entsprechend der folgenden Beschreibung die Klasse `Wuerfel`:
- Attribute:
    - Wählen Sie eine minimale Menge von Attributen, welche es Ihnen erlauben, alle Methoden zu implementieren.
    - Alle Attribute werden über den Konstruktor initialisiert.
  - Methoden:
    - Konstruktor
      - \* Parameter: siehe oben
      - \* Funktionalität:
        - Falls einer der Parameter kleiner 0 ist, soll eine `GeometricObjectException` geworfen werden.  
Nachricht:  
"Wuerfel: Parameter <Parametername> < 0; Wert: " gefolgt von dem Wert des Parameters
        - Ansonsten werden die Attribute mit den Werten der Parameter initialisiert
    - `getA`: gibt die Länge einer Seite des Würfels zurück
    - `getRaumdiagonale`: gibt die Raumdiagonale des Würfels zurück
    - `getVolumen`: gibt das Volumen des Würfels zurück
    - `getOberflaeche`: gibt die Größe der Oberfläche des Würfels zurück
- b) Implementieren Sie das Modell aus Aufgabenteil a) unter Verwendung von `java.lang.Math.sqrt`.

---

Allgemeine Hinweise:

- Verwenden Sie ausschließlich die in den Aufgabenstellungen angegebenen Klassen und deren Methoden aus Bibliotheken.
- Achten Sie auf angemessene Sichtbarkeiten.
- Achten Sie auf angemessene Benennungen der Parameter und Attribute.
- Raumdiagonale  $d$  eines Würfels mit Seiterlänge  $a$ :

$$d = \sqrt{3} \cdot a$$

### 3. Interface **GeometricObject** (6 Punkte)

- a) Erweitern Sie das Modell der Klasse **Wuerfel** wie folgt:
  - i. Fügen Sie Modell der Klasse **Kugel** aus Serie 1 hinzu.
  - ii. Extrahieren Sie aus den Klassen **Kugel** und **Wuerfel** das Interface **GeometricObject** mit den gemeinsamen Methoden der beiden Klassen.
  - iii. Modellieren Sie die Implementierungsrelationen **Kugel** implementiert **GeometricObject** und **Wuerfel** implementiert **GeometricObject**.
- b) Ändern Sie Ihre Implementierung wie folgt:
  - i. Fügen Sie die Klasse **Kugel** aus Serie 1 hinzu.
  - ii. Implementieren Sie das Interface **GeometricObject**.
  - iii. Erweitern Sie die Implementierungen der Klassen **Wuerfel** und **Kugel** so, dass Sie das Interface **GeometricObject** implementieren.

### 4. Klasse **GeometrischeObjekte** (19 Punkte)

- a) Modellieren Sie entsprechend der folgenden Beschreibung die Klasse **GeometrischeObjekte**:
  - Attribute:
    - Speichern Sie geometrische Objekte wie Kugeln und Wuerfel in *genau einer* Liste.
    - Die Liste wird im Konstruktor initialisiert.
  - Methoden:
    - Konstruktor
      - \* Parameter: Keine
      - \* Funktionalität: Initialisierung der Liste
    - **add**: fügt ein geometrisches Objekt zur Liste hinzu
    - **getAnzahl**: gibt die Anzahl der geometrischen Objekte zurück
    - **ausgeben**: Geben Sie alle in der Liste gespeicherten geometrischen Objekte aus
      - \* Geben Sie Kugeln entsprechend Serie 1 aus.
      - \* Geben Sie für Würfel die Werte für A, die Raumdiagonale, das Volumen und die Oberfläche aus.
      - Beispiel: **Wuerfel** mit **a = 5**, **Raumdiagonale = ...**
    - \* Verwenden Sie hierzu die Methoden **System.out.print** oder **System.out.println**.
- b) Kombinieren Sie das Modell aus Aufgabenteil “Interface **GeometricObject**” mit dem Modell der Klasse **GeometrischeObjekte**
  - Modellieren Sie die Aggregationsbeziehung
  - Begründen Sie die Wahl der Multiplizitäten (separate Datei ‘Begrueundung-Multiplizitaeten.txt’ im zip-File)

- 
- c) Implementieren Sie das Modell aus Aufgabenteil a) und b).

Allgemeine Hinweise:

- Verwenden Sie ausschließlich die in den Aufgabenstellungen angegebenen Klassen und deren Methoden aus Bibliotheken.
- Achten Sie auf angemessene Sichtbarkeiten.
- Achten Sie auf angemessene Benennungen der Parameter und Attribute.

## 5. Implementierung der Klasse `GeometricObjectsMain` (8 Punkte)

a) Implementieren Sie eine Klasse `GeometricObjectsMain` mit folgenden Eigenschaften

- Die Klasse enthält ausschließlich die `main`-Methode. Hilfsmethoden der `main`-Methode sind erlaubt.
- Instanzieren Sie in der `main`-Methode zwei Kugeln mit den Radien 3,14 und 275,836.
- Instanzieren Sie in der `main`-Methode drei Würfel mit den Seitenlänge 3,14, 275,836 und -207,15.
- Erzeugen Sie eine Instanz von `GeometrischeObjekte` und fügen Sie alle erfolgreich instanziierten Würfel und Kugeln hinzu. Behandeln Sie eventuell auftretende Fehler, indem Sie eine Fehlermeldung ausgeben.
- Geben Sie alle Elemente, welche in der Instanz von `GeometrischeObjekte` gespeichert sind, aus.

Allgemeine Hinweise:

- Verwenden Sie ausschließlich die in den Aufgabenstellungen angegebenen Klassen und deren Methoden aus Bibliotheken.
- Achten Sie auf angemessene Sichtbarkeiten.
- Achten Sie auf angemessene Benennungen der Parameter und Attribute.