

Einführung in die Objekt-Orientierte Modellierung und Programmierung

Praktikum 3

Abgabetermin: 05.12.2025, 23:00 Uhr

Abgabeformat: pdf für die UML-Diagramme, zip für die Programme

Maximale Anzahl an Punkten: 159

Erweiterung, Interfaces, Exceptions, Aggregation

1. Exception GeometrischesObjektException (13 Punkte)

Anforderungen Klasse GeometrischesObjektException:

- GeometrischesObjektException erweitert die Exception `java.lang.Exception`
- Ein Konstruktor ohne Parameter
- Ein Konstruktor mit folgenden Eigenschaften:
 - Parameter:
 - * Zeichenkette `message`
 - Verhalten:
 - * Aufruf des Konstruktors der Basisklasse mit der Nachricht „GeometrischesObjekt:“ gefolgt von `message`

a) Modellieren Sie die Klasse GeometrischesObjektException entsprechend den Anforderungen.

- Geben Sie bei der Basisklasse ausschliesslich alle relevanten Methoden an.

b) Implementieren Sie die Klasse GeometrischesObjektException entsprechend den Anforderungen und dem Modell.

2. Klasse Wuerfel (30 Punkte)

Anforderungen Klasse Wuerfel:

- Attribute:
 - Ein Fließkommazahl `seitenlaenge`
- Methoden:
 - Konstruktor:
 - * Parameter:
 - eine Fließkommazahl `seitenlaenge`
 - * Funktionalität:
 - Falls der Parameter kleiner 0 ist, soll eine GeometrischesObjektException geworfen werden.
Nachricht: „Wuerfel: Parameter {Parametername} < 0; Wert: {Parameterwert}“
 - Ansonsten wird das Attribut mit dem Wert des Parameters initialisiert.
 - `getSeitenlaenge`: gibt die Länge einer Seite des Würfels zurück

- `getVolumen`: gibt das Volumen des Würfels zurück
- `getOberflaeche`: gibt die Größe der Oberfläche des Würfels zurück
- `toString`: erzeugt einen String mit folgenden Informationen
 - * Text: “Würfel mit”
 - * “Seitenlänge = ” und der Seitenlänge
 - * “Volumen = ” und dem Volumen
 - * “Oberfläche = ” und der Oberfläche

- a) Modellieren Sie die Klasse `Wuerfel` entsprechend den Anforderungen.
- b) Implementieren Sie die Klasse `Wuerfel` entsprechend den Anforderungen und dem Modell.

3. Klasse Quader (42 Punkte)

Anforderungen Klasse `Quader`:

- Attribute:
 - Drei Fließkommazahlen `laenge`, `breite`, `hoehe`
 - Methoden:
 - Konstruktor:
 - * Parameter:
 - drei Fließkommazahlen `laenge`, `breite`, `hoehe`
 - * Funktionalität:
 - Falls einer der Parameter kleiner 0 ist, soll eine `GeometrischesObjektException` geworfen werden.
Nachricht: „Quader: Parameter {Parametername} < 0; Wert: {Parameterwert}“
 - Ansonsten werden die Attribute mit den Werten der Parameter initialisiert.
 - `getLaenge`: gibt die Länge des Quaders zurück
 - `getBreite`: gibt die Breite des Quaders zurück
 - `getHoehe`: gibt die Höhe des Quaders zurück
 - `getVolumen`: gibt das Volumen des Quaders zurück
 - `getOberflaeche`: gibt die Größe der Oberfläche des Quaders zurück
 - `toString`: erzeugt einen String mit folgenden Informationen
 - * Text: “Würfel mit”
 - * “Länge = ” und der Länge
 - * “Breite = ” und der Breite
 - * “Höhe = ” und der Höhe
 - * “Volumen = ” und dem Volumen
 - * “Oberfläche = ” und der Oberfläche
- a) Modellieren Sie die Klasse `Quader` entsprechend den Anforderungen.
 - b) Implementieren Sie die Klasse `Quader` entsprechend den Anforderungen und dem Modell.

4. Interface GeometrischesObjekt (11 Punkte)

- a) Erweitern Sie das Modell wie folgt:
 - i. Extrahieren Sie aus den Klassen `Quader` und `Wuerfel` das Interface `GeometrischesObjekt` mit den gemeinsamen Methoden der beiden Klassen.

ii. Modellieren Sie die Implementierungsrelationen `Quader` implementiert `GeometrischesObjekt` und `Wuerfel` implementiert `GeometrischesObjekt`.

b) Ändern Sie Ihre Implementierung wie folgt:

- i. Implementieren Sie das Interface `GeometrischesObjekt`.
- ii. Erweitern Sie die Implementierungen der Klassen `Wuerfel` und `Quader` so, dass diese das Interface `GeometrischesObjekt` implementieren.

5. Klasse `GeometrischeObjekte` (37 Punkte)

Anforderungen Klasse `GeometrischeObjekte`:

- Attribute:
 - Speichern Sie geometrische Objekte wie Quader 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
 - `berechneGesamtesVolumen`: berechnet die Summe der Volumina aller in der Liste gespeicherten Objekte
 - `berechneGesamteOberflaeche`: berechnet die Summe der Oberflächen aller in der Liste gespeicherten Objekte
 - `ausgeben`: gibt aus
 - * die Anzahl der in der Liste gespeicherten geometrischen Objekte
 - * alle in der Liste gespeicherten geometrischen Objekte
 - * das gesamte Volumen der in der Liste gespeicherten geometrischen Objekte
 - * die gesamte Oberfläche der in der Liste gespeicherten geometrischen Objekte

- a) Modellieren Sie die Klasse `GeometrischeObjekte` entsprechend den Anforderungen.
- b) Kombinieren Sie das Modell aus Aufgabenteil “Interface GeometrischesObjekt” mit dem Modell der Klasse `GeometrischeObjekte`
 - Modellieren Sie die Aggregationsbeziehung
 - Begründen Sie die Wahl der Multiplizitäten bei der Abnahme
- c) Implementieren Sie die Klasse `GeometrischeObjekte` entsprechend den Anforderungen und dem Modell.

Allgemeine Anforderungen:

- Verwenden Sie aus Bibliotheken ausschließlich die Klassen
 - `java.util.List`
 - `java.util.ArrayList`
 - `System.out.println`und deren Methoden.
- Achten Sie auf angemessene Sichtbarkeiten.
- Achten Sie auf angemessene Benennungen der Parameter und Attribute.

6. Klasse GeometrischeObjekteMain (26 Punkte)

- a) Implementieren Sie eine Klasse `GeometrischeObjekteMain` mit folgenden Eigenschaften
- Die Klasse enthält ausschließlich die `main`-Methode. Hilfsmethoden der `main`-Methode sind erlaubt.
 - Behandeln Sie eventuell auftretende Fehler, indem Sie eine Fehlermeldung ausgeben.
 - Instanziieren zwei Quader mit den Seitenlängen (3, 4, 5), (16, -1, 15) sowie (200, 200, 200).
 - Instanziieren drei Würfel mit den Seitenlängen -207,15, 4 und 200.
 - Erzeugen Sie eine Instanz von `GeometrischeObjekte` und fügen Sie alle erfolgreich instanzierten Würfel und Quader hinzu.
 - Geben Sie alle Elemente, welche in der Instanz von `GeometrischeObjekte` gespeichert sind, aus.

Allgemeine Anforderungen:

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