

Objektorientierte Programmierung Grundlagen

Übung 7 für die Prüfungsvorleistung

Klassen und Objekte und Vererbung

Aufgabe 1 zunächst eine einfache Programmieraufgabe (8 Punkte)

Schreiben Sie eine Klassenmethode `reverseInt()`, die als Argument eine ganze Zahl `n` vom Typ `long` erhält und als Ergebnis einen `long`-Wert zurückliefert, der die Ziffern von `n` in umgekehrter Reihenfolge enthält.

Beispiel:

```
reverseInt(1234) == 4321  
reverseInt(17) == 71  
reverseInt(454) == 454
```

Hinweis (für Fortgeschrittene): Bitte verwenden Sie keine Zeichenketten!

Aufgabe 2 Klassen und Objekte (5+8 Punkte)

Die Klasse `DemoKlasse` hat ein `private` Attribut `wert` vom Typ `int`, das von einem allgemeinen Konstruktor initialisiert wird.

Die Zugriffsmethoden `getWert()` und `setWert()` dienen zum lesenden bzw. schreibenden Zugriff auf das Attribut `wert`.

Die Objekt-Operation `addieren()` erhält als Argument ein `DemoKlasse`-Objekt `obj`; sie liefert ein *neues* `DemoKlasse`-Objekt zurück, dessen Attributwert die Summe der Attributwerte des aktuellen Objekts und des Objekts `obj` ist. Das aktuelle Objekt und das Argument `obj` dürfen nicht verändert werden.

- a) Geben Sie eine UML-Spezifikation der Klasse `DemoKlasse` an, die auch die Argumente und Rückgabewerte der Operationen umfasst.
- b) Implementieren Sie die Klasse `DemoKlasse` in Java.

Aufgabe 3 Vererbung

(4+4+4+1 Punkte)

- a) Implementieren Sie eine Klasse `GeomObjekt`, die als Basisklasse geometrischer Objekte dienen soll. Die Klasse soll die x- und y-Koordinaten des Objekts aufnehmen. Für die Attribute `x` und `y`, die auch in von `GeomObjekt` abgeleiteten Klassen zugreifbar sein sollen, besitzt `GeomObjekt` einen allgemeinen Konstruktor. `GeomObjekt` soll weiterhin über eine parameterlose Methode `void print()` verfügen. Von der Klasse `GeomObjekt` darf kein Exemplar erzeugt werden können.
- b) Implementieren Sie eine Klasse `Kreis`, die `GeomObjekt` um den Durchmesser `d` des Kreises erweitert. Das Attribut `d` soll nur innerhalb der Klasse `Kreis` zugreifbar sein. Die Klasse `Kreis` soll einen allgemeinen Konstruktor besitzen und eine eigene parameterlose Methode `void print()` erhalten, welche die Zeichenkette "Kreis" ausgibt.
- c) Implementieren Sie eine Klasse `Rechteck`, die `GeomObjekt` um Länge `l` und Breite `b` des Rechtecks erweitert. Die Attribute `l` und `b` sollen nur innerhalb der Klasse `Rechteck` zugreifbar sein. Die Klasse `Rechteck` soll einen allgemeinen Konstruktor besitzen und eine eigene parameterlose Methode `void print()` erhalten, welche die Zeichenkette "Rechteck" ausgibt.
- d) Welches der folgenden UML-Diagramme gibt die Beziehung der beiden Klassen `GeomObjekt` und `Rechteck` korrekt wieder? Kreuzen Sie die entsprechende Abbildung an.

