

Programmier-Praktikum Blatt 2

Bemerkung: Für jedes Java-Programm ist eine geeignete Ein-/Ausgabe vorzusehen, anhand derer die Funktion demonstriert werden kann. Zu beachten ist, daß das GUI von der Funktion getrennt realisiert werden muß. Es ist also nicht ausreichend, ausschließlich eine Klasse zu implementieren, die die Funktion realisiert und gleichzeitig das GUI.

Aufgabe 2.1: Geometrische Objekte

Es soll eine Klassenhierarchie `GeoObject` implementiert werden. Die Wurzelklasse ist abstrakt und bietet folgende Operationen an:

- berechne Flächeninhalt
- berechne Kantenumfang

Folgende Subklassen sollen realisiert werden:

- Rechteck
- Quadrat (ist ein Spezialfall vom Rechteck! → Vererbung)
- Kreis
- rechtwinkliges Dreieck (Vereinfachung: Katheten sind achsen-parallel zum Koordinatensystem)

Die geometrischen Daten der Objekte sollen im kartesischen Koordinatensystem gespeichert werden.

In einer Anwendung soll die Klassenhierarchie verwendet werden. Es soll möglich sein, neue geometrische Objekte anzulegen, transient zu speichern (also keine Speicherung in einer Datei!), und in einer Liste textuell (Bsp. `Rechteck.1`, `Kreis.1`, `Kreis.2`) anzuzeigen, um eine Auswahl zu ermöglichen.

Für die Lösung dieser Aufgabe sollen die gespeicherten Objekte *nicht* graphisch dargestellt werden. Das GUI soll lediglich die Eingabe der Eckpunkte sowie den Aufruf der einzelnen Funktionen und die Anzeige der Ergebnisse ermöglichen.

Aufgabe 2.2: Kundenverwaltung

Es soll eine Kundenverwaltung in Java implementiert werden. Hierzu sollen die folgenden Daten, in einer geeigneten Weise (Klasse) gespeichert werden:

- Vorname
- Nachname
- Geburtsdatum
- Familienstand
- Adresse

Die Kundendaten werden beim Beenden der Anwendung gespeichert und beim Starten geladen. Eine Möglichkeit die Daten zu speichern besteht in der Verwendung der Serialisierung der Daten, die in der Klassenbibliothek von Java angeboten wird.

Es soll eine beliebige Anzahl von Kunden verwaltet werden können. Um die Kunden verwalten zu können, sollen folgende Operationen auf der Kundendatei ausführbar sein, wobei vereinfachend davon ausgegangen werden kann, daß alle Kunden unterschiedliche Nachnamen besitzen.

1. Einfügen eines neuen Kunden
2. Löschen eines bestehenden Kunden
3. Ändern des Familienstandes und der Adresse
4. Ausgabe aller Daten zu einem Kunden

Beim Einfügen eines neuen Kunden wird eine eindeutige Kundennummer vergeben und angezeigt. Alle weiteren Operationen sind mit der Kundennummer durchführbar.

Fehler sind korrekt zu behandeln. Es sollen z.B. bereits bekannte Kunden nicht mehr eingefügt werden können.

Aufgabe 2.3: Kundenverwaltung (Änderung)

In Aufgabe 2.2 haben Sie alle Daten mit Hilfe der Serialisierung gespeichert. Ändern Sie das nun so ab, daß Sie die Serialisierung nicht mehr verwenden¹. Sie können die Daten speichern, indem Sie alle Attribute von allen Objekten, die persistente Daten verwalten, textuell speichern. Diese textuell gespeicherten Daten müssen Sie beim Laden wieder einlesen und an neu erzeugten Objekten passenden Typs wieder setzen. Zu beachten sind hierbei die Beziehungen der Objekte untereinander.

¹Wenn Sie in Java Software entwickeln, können Sie Serialisierung verwenden, die jedoch auch Nachteile besitzt, die hier nicht diskutiert werden sollen. Andere Programmiersprachen hingegen besitzen *nicht* die Funktionalität der Serialisierung. Daher ist es sinnvoll wenn Sie auch mindestens eine weitere Methode zum Speichern von Daten kennen, die mit vielen anderen Sprachen funktioniert.