

Übungsblatt 10 zu “Programmiersprachen”

Berthold Hoffmann, Studiengang Informatik (hof@tzi.de)
Besprechung am 17. 1. 2005

Aufgabe 1: Datenstrukturen in objektorientierten Sprachen

Wie lassen sich in “rein” objektorientierten Sprachen wie JAVA und EIFFEL *annäherungsweise* kartesische Produkte und disjunkte Vereinigungen repräsentieren?

Wie weit gelingt die Annäherung?

Untersuche die Beispiele `Date` und `Number` aus dem Kapitel “Werte” der Vorlesung.

Aufgabe 2: Objektorientierte Glückseligkeit

Laut Bertrand Meyer führen folgende 7 Schritte in die objektorientierte Glückseligkeit:

1. Objekte
 - Variablen werden mit ihren Methoden gekapselt.
 - Daten legen die Programmstruktur fest, nicht die Funktionen (*data-driven design*).
2. Datenabstraktion (*information hiding*)
 - Objekte sind abstrakte Variablen; ihre Darstellung bleibt verborgen.
 - Sie dürfen nur mit Zugriffsmethoden überschrieben werden.
3. automatische Speicherverwaltung (*garbage collection*)
 - Nicht mehr benutzte Objekte werden automatisch wieder freigegeben.
4. Klassen
 - Jeder zusammengesetzte Typ ist ein Modul und jeder Modul definiert einen Typ.
 - Von jeder Klasse können beliebig viele Objekte angelegt werden (Instantiierung).
5. Vererbung
 - Klassen können andere Klassen erweitern / beschränken (Unterklassen).
6. Polymorphie, Überschreiben und dynamisches Binden
 - Methoden lassen sich auf Objekte mehrerer Klassen anwenden.
 - Methoden können in Unterklassen verschieden realisiert sein.
 - Methoden werden anhand des dynamischen Typs des Empfängerobjekts bestimmt.
7. mehrfache und wiederholte Vererbung
 - eine Klasse kann von mehreren Klassen erben (*multiple inheritance*).
 - eine Klasse kann mehrmals von einer Klasse erben (*repeated inheritance*).

Wie weit kann bzw. *muss* man in JAVA und C++ auf diesem Weg gehen?

Fehlt sonst noch was zur objektorientierten Glückseligkeit ?