
Übungen zu Kapitel 14

14.1 Kovarianz von Arrays [ESS05]:

Verwenden Sie die Klasse `Konto` und ihre Unterklasse `Sparkonto`. Deklarieren Sie ein `Konto`-Array `kontoArr` und ein `Sparkonto`-Array `sparArr`. Erzeugen Sie einige `Konto`-Objekte und füllen Sie damit das Array `kontoArr`; erzeugen Sie einige `Sparkonto`-Objekte und füllen Sie damit das Array `sparArr`. Legen Sie ein `Object`-Array `objArr` an. Prüfen Sie, ob der Compiler folgende Zuordnung akzeptiert:

```
anyArr = kontoArr = sparArr;
```

Aus der Tatsache dass für die *Klassen* gilt: `Sparkonto extends Konto extends Object` schließt der Compiler, dass auch ein *Array* der Unterklasse `Sparkonto` einem *Array* seiner Oberklasse `Konto` und dieses wiederum einem Array der Basisklasse `Object` zugewiesen werden kann, d.h. auch die *Arrays* sind automatisch typkompatibel.

Unter der *Kovarianz von Arrays* versteht man allgemein: In Java sind Arrays kovariant, da ein *Array von Unterklassenobjekten* automatisch eine Unterklasse eines *Arrays von Oberklassenobjekten* ist. Somit ist ein `Object`-Array: `Object [] objArr` typkompatibel zu *jedem anderen* Java-Array. Informieren Sie sich über damit verbundene Probleme, z.B. in [ESS05].

14.2 Erweiterung der Klasse `Bank`:

Erweitern Sie die Klasse `Bank` des Kapitels um Funktionalität zum Löschen eines Kontos. Ferner soll es möglich sein, die Summe der bei der Bank angelegten Gelder getrennt nach Kontotypen ausgeben zu lassen. Für Girokonten soll ermittelt werden, wie groß die Summe der überzogenen Kontostände ist.

14.3 Verwenden von Interfaces:

Modellieren Sie die Zusammenhänge des ausführlichen Bank-Beispiels durch Interfaces:

Bilden Sie jeden Kontotyp durch ein eigenes Interface ab. Die verschiedenen Interfaces stehen in einer Vererbungs-Hierarchie. Eine Klasse `KontoTypen` soll alle Interfaces implementieren. In der ausführbaren Klasse `Bank` soll bei Typangaben und Methodenaufrufe überall wo möglich mit den Interfaces gearbeitet werden. Objekte werden nur von `KontoTypen` erzeugt und diese verwendet.

14.4 Kritik an Vererbungs-Polymorphie:

Folgende anspruchsvolle Artikel sind in der Zeitschrift *Informatik-Spektrum* erschienen und enthalten eine sehr interessante Diskussion über Nutzen und Nachteile objektorientierter Programmierung, insbesondere auch bezüglich Vererbung und Polymorphie. Lesen Sie die Artikel und versuchen Sie, die Argumente auf Grundlage Ihrer Java-Kenntnisse zu verstehen:

M.Broy, J.Siedersleben: "Objektorientierte Programmierung und Software-Entwicklung", Informatik Spektrum, Februar 2002.

S.Jähnichen, S.Herrmann: "Was, bitte, bedeutet Objektorientierung?", Informatik Spektrum, August 2002.

H.v.Drachenfels: "Komponentenorientierte Programmierung im Kleinen", Informatik Spektrum, April 2005.

14.5 Interfaces:

- Verwenden Sie ein Interface und versuchen Sie, Instanzen davon zu erzeugen. Wie lautet die Fehlermeldung des Compilers? Erläutern Sie diese Meldung!
- [MER04]: Wenn ein Interface von vielen Klassen implementiert worden ist, welche Folge hat dann das Hinzufügen einer weiteren Methode zum Interface für die bestehenden Klassen?