Einstieg in die Informatik mit Java

Schnittstellen

Gerd Bohlender

Institut für Angewandte und Numerische Mathematik

Gliederung

Gliederung

Einführung

Schnittstellen sind ähnlich aufgebaut wie Klassen, aber einfacher:

- Sie dienen zur Beschreibung abstrakter Konzepte, wie z.B. „reelle Funktion“, „Relation“, usw.
- Schnittstellen dürfen nur Konstanten enthalten, keine Variablen.
- Schnittstellen dürfen nur abstrakte Methoden enthalten, keine Implementierungen (statt `{...}` wird nur ; angegeben).
- Referenzen auf Schnittstellen sind erlaubt, aber Instanzen können nicht gebildet werden.
- Klassen können Schnittstellen implementieren; dann dürfen Instanzen der Klasse über die Referenzen der Schnittstelle angesprochen werden.

Gliederung

Definition einer Schnittstelle

Syntax

```
interface Name {
      ... //nur Konstanten, keine Variablen!
      ... //nur abstrakte Methoden, keine Implementierung!
}
```

Syntax abstrakte Methode

```
Modifizierer Ergebnistyp Methodenname (formale Argumentliste);
```

- Abstrakte Methoden deklarieren nur die Methode.
- Sie besitzen keinen Anweisungsteil.
- Die Definition der Methoden muss später bei der Implementierung der Schnittstelle erfolgen.
- Eine Schnittstelle stellt einen *Kontrakt* dar, der von der Implementierung erfüllt werden muss.

Eigenschaften von Schnittstellen

- Schnittstellen können keine Instanzen bilden,
- Referenzen sind allerdings erlaubt.

Beispiel

```
interface Funktion {
  double auswerten (double x);
}
```

Verwendung z.B. in einer Methode:

```
void ausgeben (Funktion f, double x) {
  System.out.println (f.auswerten(x));
}
```

Gliederung

Implementierung einer Schnittstelle

- Eine Klasse kann eine (oder mehrere) Schnittstellen implementieren.

Syntax

```
class Implementierung implements Schnittstelle { ... }
```

Bemerkung:

- Dabei müssen normalerweise alle Methoden der Schnittstellen implementiert werden.
- Werden nicht alle Methoden implementiert, dann ist die Klasse abstrakt und muss durch `abstract class` gekennzeichnet werden.
- Eine Klasse darf auch mehrere Schnittstellen implementieren, diese werden mit Komma aufgelistet.

Beispiel Funktion, Fortsetzung

Beispiel

```
class Xhoch3 implements Funktion {
   double auswerten (double x) {
        return x*x*x;
   }
}
```

Verwendung z.B. in main:

Beispiel

```
Funktion f;          // nur Referenz
// Instanz einer Implementierung der Schnittstelle bilden
f = new Xhoch3(); // Typen werden angepasst
double differenz = f.auswerten(2) - f.auswerten(1);
ausgeben (f, 5);  // Methode ausgeben für f aufrufen
```

Eigenschaften

- Alle Methoden einer Schnittstelle sind `public abstract`. Dies braucht nicht explizit angegeben zu werden.
- Widersprüchliche Angaben wie z.B. `final` oder `private` sind verboten.
- Alle Datenfelder einer Schnittstelle sind `public static final`. Dies braucht ebenfalls nicht explizit angegeben zu werden.
- Angaben wie z.B. `private` oder `protected` sind verboten.