Funktionale und objektorientierte Programmierkonzepte

TECHNISCHE UNIVERSITÄT DARMSTADT

Präsenz-Sprechstunde B

Simon Hock, Nhan Huynh, Daniel Mangold

Organisatorisches

TECHNISCHE UNIVERSITÄT DARMSTADT

- 24.11.2021 Präsenzsprechstunde im Raum in Raum S103/**223**!
- Erinnerung: Themenvorschläge im Forum
- Zusätzliche Materialien wie die Präsentationen sind im Moodle Forum zu finden.

Vererbung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Grundlegendes Konzept der Objektorientierung
- Modellierung von Hierarchien in der realen Welt mit Hilfe von Klassen
- Basisklasse: Generalisierung
  - Verallgemeinerung eines Objekts mit seinem Verhalten
- Abgeleitete Klassen: Spezialisierung
  - Eigene Ausprägungen und eigenes Verhalten

Beispiel Typhierachie

TECHNISCHE UNIVERSITÄT DARMSTADT

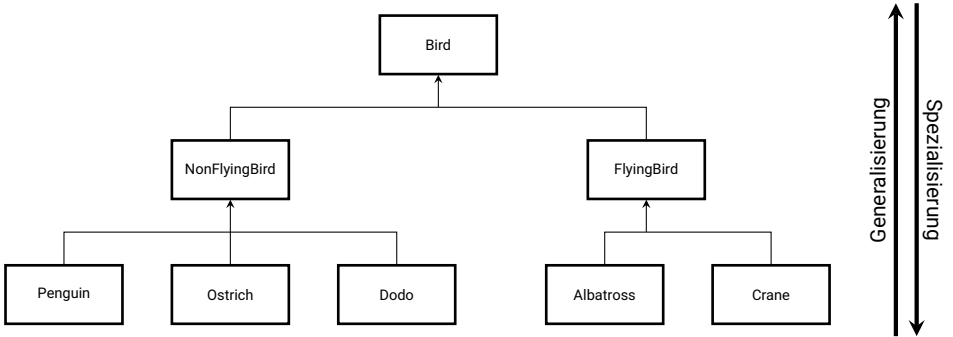

Abbildung: Typhierarchie - Modellierung von Vögeln

Überschreiben von Methoden

TECHNISCHE UNIVERSITÄT DARMSTADT

- Eigene Implementierung einer geerbten Methode
- Erlaubt das Funktionalitäten in der abgeleiteten Klasse zu verändern
- Gleichzeitig können aber die anderen Methoden der Oberklasse immer noch weiter verwendet werden.
- `@Override` Annotation (optional)
  - Informiert den Compiler, dass das Element ein in einer Oberklasse deklariertes Element überschreiben soll.

Erweiterung der Klasse Robot

```
1  public class FastRobot extends Robot {
2
3    public FastRobot(int x, int y, Direction direction, int numberOfCoins) {
4      super(x, y, direction, numberOfCoins);
5    }
6
7    @Override
8    public void move() {
9      super.move();
10     super.move();
11   }
12 }
```

Welche Implementierung wird ausgeführt?

TECHNISCHE UNIVERSITÄT DARMSTADT

- Implementierung von `Robot`?
- Implementierung von `FastRobot`?

```
1 Robot robot = new FastRobot(0,0,Direction.UP,0);
2 robot.move();
```

Überladen von Methoden

TECHNISCHE UNIVERSITÄT DARMSTADT

- Methoden mit gleichem Namen können innerhalb einer Klasse definiert werden.
- Der Rückgabetyp kann beliebig sein, aber die Parameterlisten müssen verschieden sein.

Information: Methoden werden nicht ausschließlich anhand ihres Namens identifiziert, sondern auch über die Typen, Anzahl und Reihenfolge ihrer Argumente (Parameter) - ihre Signatur.

Überladen von Methoden - Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  public class Point {
2
3    private final double x;
4    private final double y;
5
6    public Point(double x, double y){
7      this.x = x;
8      this.y = y;
9    }
10 }
```

```java
public Point copy() {
  return new Point(x, y);
}

public Point[] copy(int n) {
  Point[] copies = new Point[n];
  for (int i = 0; i < copies.length; i++) {
    copies[i] = new Point(x, y);
  }
  return copies;
}
```

Konstruktoren

TECHNISCHE UNIVERSITÄT DARMSTADT

- Haben keinen Rückgabewert und der Name ist gleich dem Namen der Klasse.
- Konstruktoren werden nur einmal aufgerufen und zwar, wenn man ein neues Objekt erstellt.
- Konstruktoren werden nicht vererbt.
- Falls kein Konstruktor definiert wird, legt der Compiler einen leeren Konstruktor ohne Parameter an.
  - Es gibt mindestens einen Konstruktor.
  - Konstruktoren können überladen werden.

```
1 Robot robot = new Robot(0,0,Direction.UP,0);
```

```java
public Robot(int x, int y) {
  super(x, y);
  setGlobalWorld();

  world.checkXCoordinate(x);
  world.checkYCoordinate(y);

  world.addRobot(this);
}
```

Schlüsselwort `this`

TECHNISCHE UNIVERSITÄT DARMSTADT

- Referenzvariable, die auf das aktuelle Objekt verweist.
- Das Schlüsselwort `this` wird hauptsächlich in drei Situationen verwendet:
  - Vermeidung von Mehrdeutigkeit von Variablenreferenzen
  - Aktuelles Objekt als Argument, das an ein anderes Objekt übergeben wird
  - Alternativer Konstruktoraufruf

Vermeidung von Mehrdeutigkeit von Variablenreferenzen

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  public class Point {
2
3    private final double x;
4    private final double y;
5
6    public Point(double x, double y){
7      this.x = x;
8      this.y = y;
9    }
10 }
```

Aktuelles Objekt als Argument, das an ein anderes Objekt übergeben wird

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1 public Point(Point point) {
2   x = point.x;
3   y = point.y;
4 }
5
6 public Point copy() {
7   return new Point(this);
8 }
```

Alternativer Konstruktoraufruf

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1 public Point(double x, double y){
2   this.x = x;
3   this.y = y;
4 }
5
6 public Point() {
7   this(0, 0);
8 }
```

Enums

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kurzform von Enumeration - Aufzählung
- Abgleitete Klassen von java.lang.Enum
- Bietet die Möglichkeit, vordefinierte Konstanten für Variablen festzulegen
- Nützliche Methode: `int ordinal()`: Gibt die Postion des Enums in der Enumdeklaration zurück.

Enum Direction - FOPBot

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1 public enum Direction {
2   UP,
3   RIGHT,
4   DOWN,
5   LEFT
6 }
```

TECHNISCHE UNIVERSITÄT DARMSTADT

Selbstständiges Arbeiten