

# Eine Fachklasse in Java

```

public class Person {
private String name;

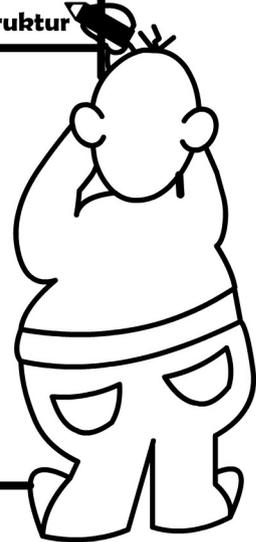
public Person(){
}

public setName(pName){
this.name = pName;
}

public String getName(){
return name;
}
}

```

**HINWEISE**  
 Immer  
 die  
 gleiche  
 Struktur

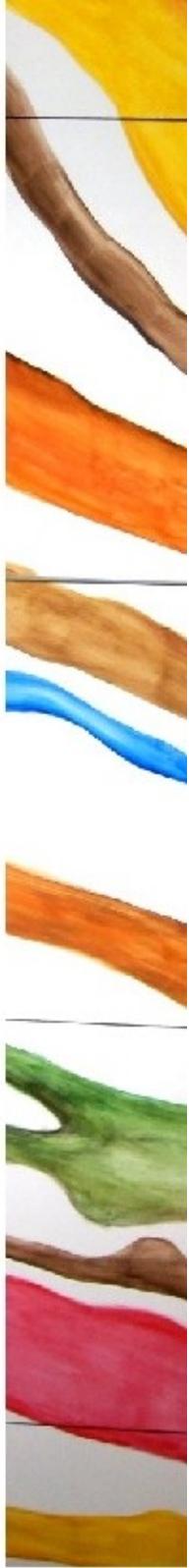


**Fachklassen**

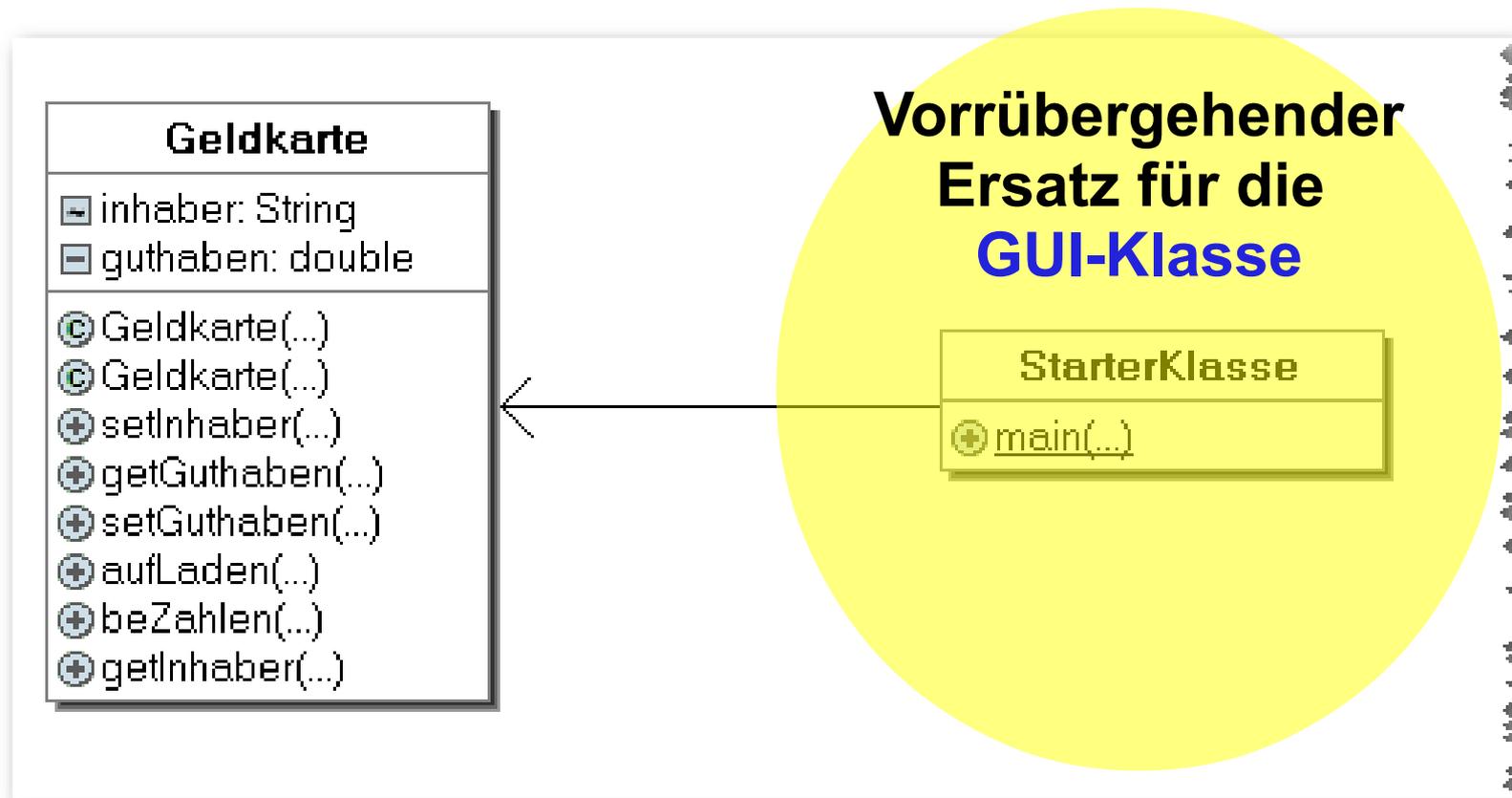
**K**aufmännische  
**S**chule  
**W**angen

Autor: Christine Janischek  
 Thema: Fachklasse in Java

**Wege zeigen, öffnen, gehen**



# KLASSENDIAGRAMM und MAIN()-Methode



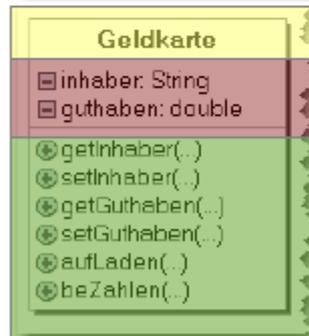
Autor: Christine Janischek  
Thema: Fachklasse in Java

**K**aufmännische  
**S**chule  
**W**angen

**Wege zeigen, öffnen, gehen**

# MODELL und JAVA-QUELLCODE

Anstatt an einem bestimmten Objekt aufzurufen (z.B. geldkarte1) sagen wir einfach er soll die **aktuelle geldkarteX** nutzen. Dazu rufen wir Methoden mit „**this.methodName ()**“ auf.



```
public class Geldkarte
{
    // Anfang Attribute
    private String inhaber;
    private double guthaben;
    // Ende Attribute

    //Parameterloser default Konstruktor
    public Geldkarte(){
    }

    // Anfang Methoden
    public String getInhaber(){
        return inhaber;
    }

    public void setInhaber(String pInhaber){
        this.inhaber = inhaber;
    }

    public double getGuthaben(){
        return guthaben;
    }

    public void setGuthaben(double pGuthaben){
        this.guthaben = guthaben;
    }

    public String aufLaden(double pBetrag){
        String mmeldung="";
        return mmeldung;
    }

    public String beEahlen(double pBetrag){
        String mmeldung="";
        return mmeldung;
    }

    // Ende Methoden
}

```

## KONSTRUKTOREN einer KLASSE

Merke: Der Konstruktor ist notwendig um Objekte (geldkarte1-3) erzeugen zu können.

Eine Klasse kann mehrere Konstruktoren besitzen!

```
//Deklaration e. parameterlosen default Konstruktors  
public Geldkarte() {  
  
}
```

```
//Deklaration e. Konstruktors mit Parametern  
public Geldkarte(String pInhaber, double pGuthaben) {  
    this.inhaber = pInhaber;  
    this.guthaben = pGuthaben;  
}
```

# DIE TEST-METHODE (mit Konstruktoren Objekte erzeugen)

```
public class StarterKlasse
{
    public static void main(String[] args) {
        //Objekt geldkarte1 erzeugen
        Geldkarte geldkarte1 = new Geldkarte();

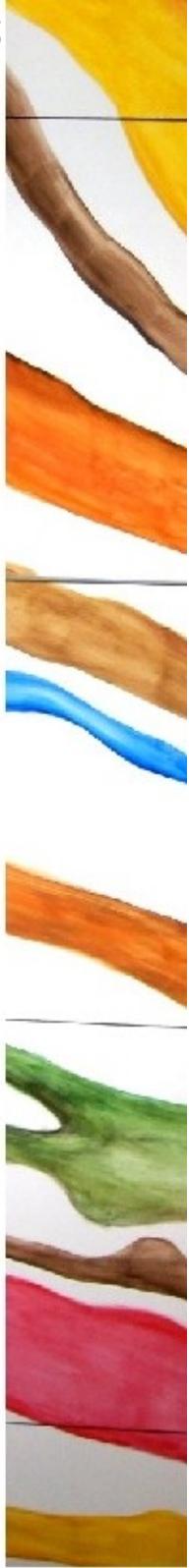
        //Attributwerte initialisieren (setzen)
        geldkarte1.setInhaber("Christian Wulff");
        geldkarte1.setGuthaben(50.00);

        //Attributwerte (auf der Konsole) ausgeben,dazu (vom Haufen) holen
        System.out.println("Inhaber: " + geldkarte1.getInhaber());
        System.out.println("Guthaben: " + geldkarte1.getGuthaben());
    }
}
```

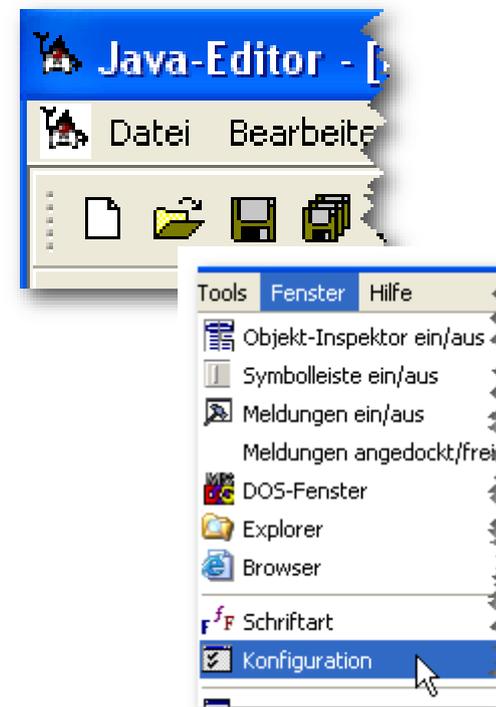
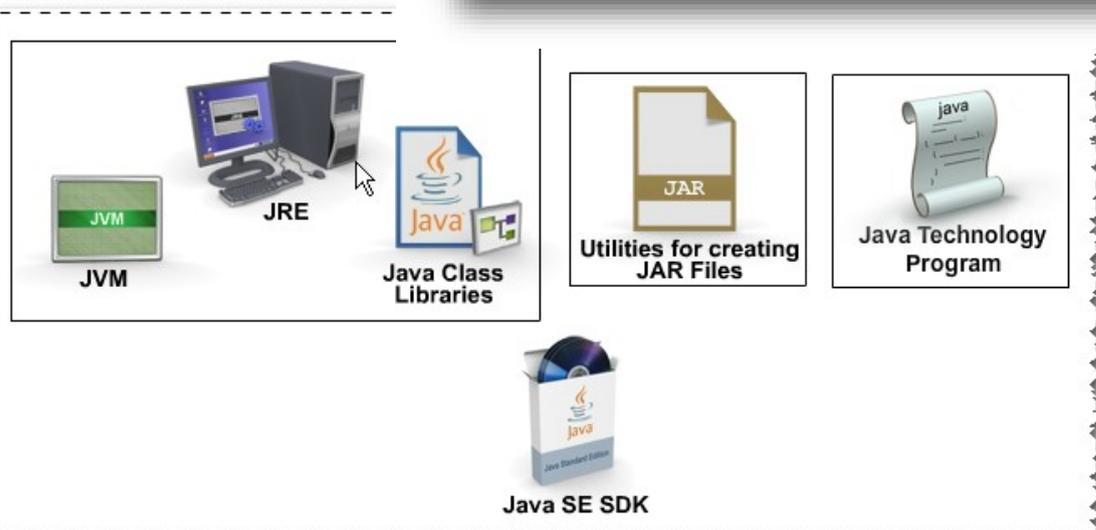
Autor: Christine Janischek  
Thema: Fachklasse in Java

**K**aufmännische  
**S**chule  
**W**angen

**Wege zeigen, öffnen, gehen**



# ENTWICKLUNGSUMGEBUNG konfigurieren



Grafik ist möglicherweise urheberrechtlich geschützt!  
[Oracle Corporation]

Thema: Fachklasse in Java



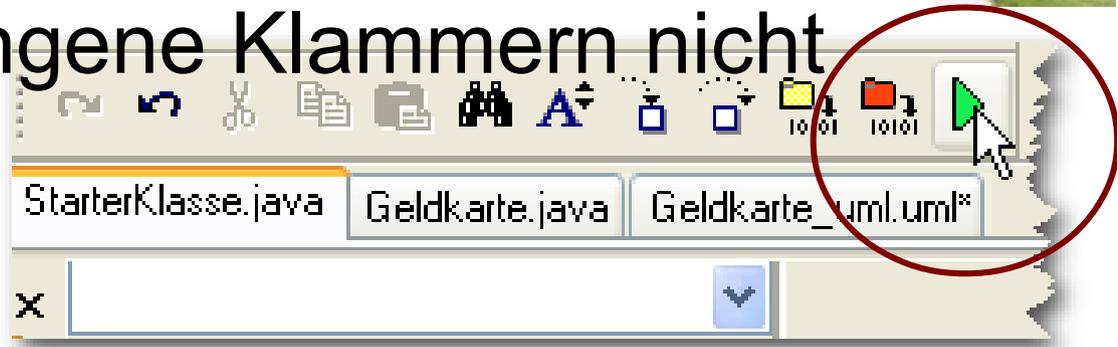
# HINWEIS ZUM DEBUGGING (Fehleranalyse)

- x Jeder fehlende Semikolon (;) erzeugt Fehlermeldungen!
- x Jede fehlende Klammer erzeugt Fehlermeldungen!
- x Keine Umlaute verwenden!
- x Leerzeichen sind auch Zeichen!
- x Runde und geschwungene Klammern nicht

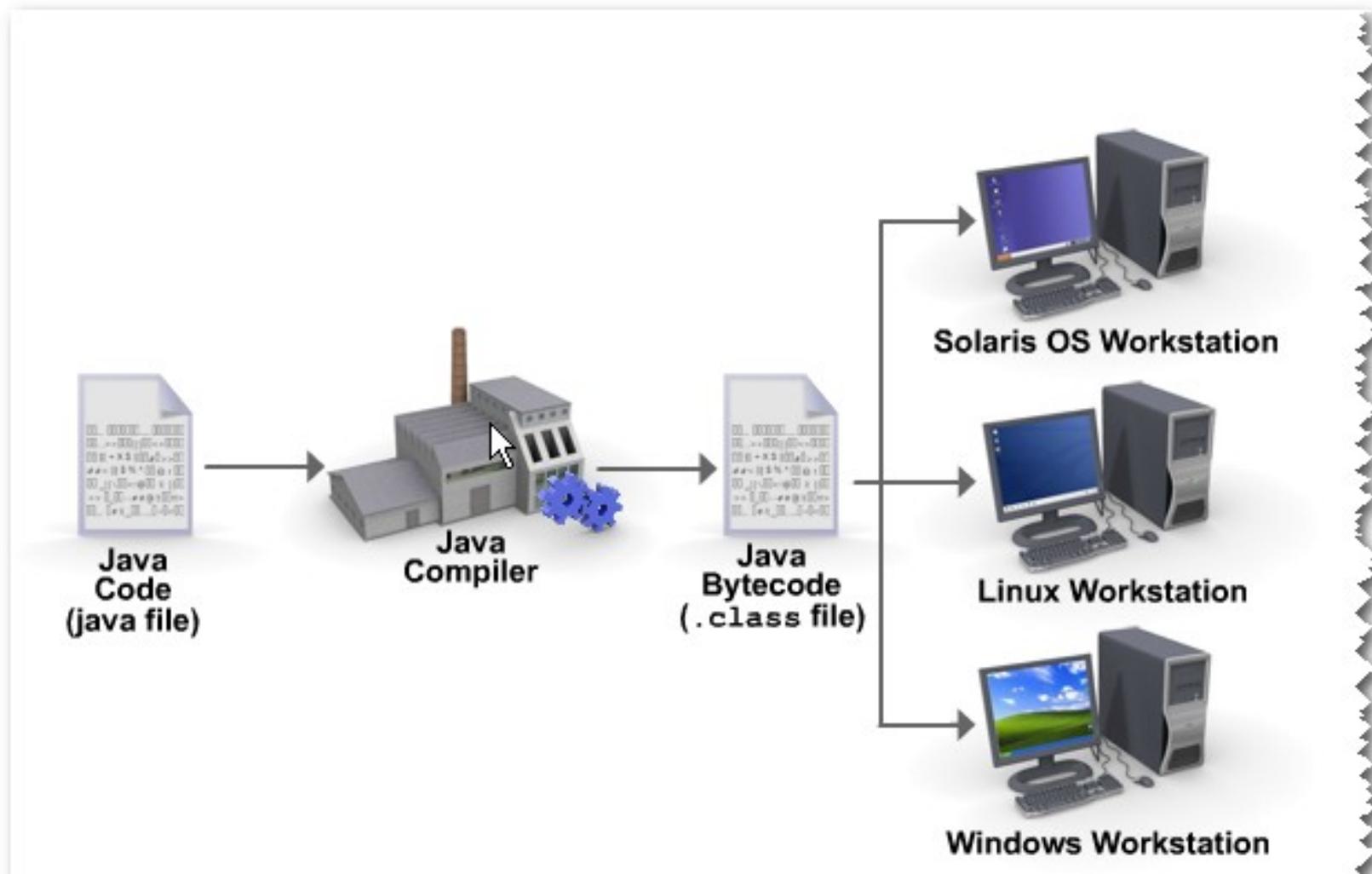
**AUS FEHLERN LERNEN!**

**verwechseln!**

Autor: Christine Janischek  
Thema: Fachklasse in Java



# Der COMPILER

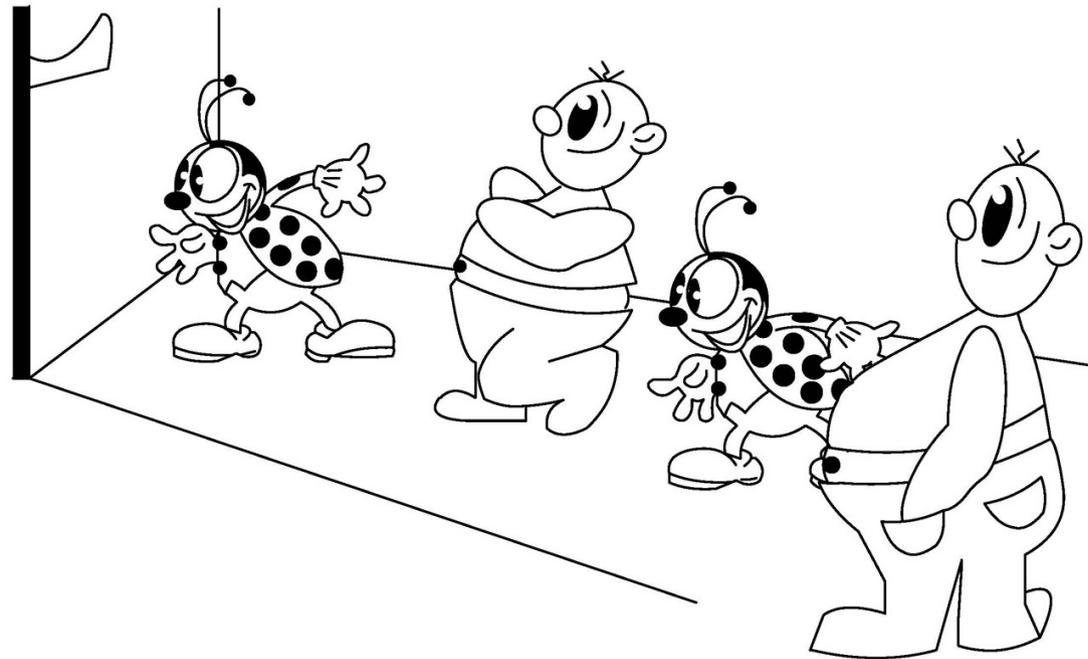


Autor: Christine Janischek  
Thema: Fachklasse in Java

Grafik ist möglicherweise urheberrechtlich geschützt!  
[Oracle Corporation]

# DIE WARTESCHLANGE UND KUMMERBOX

Falls Fragen auftauchen nutzen Sie die Kompetenzen Ihrer Kollegen



**K**aufmännische  
**S**chule  
**W**angen

Autor: Christine Janischek  
Thema: Fachklasse in Java

**Wege zeigen, öffnen, gehen**

