

III.2. Objekte, Klassen und Methoden

- 1. Grundzüge der Objektorientierung
- 2. Methoden, Unterprogramme und Parameter
- 3. Datenabstraktion
- 4. Konstruktoren
- 5. Vordefinierte Klassen

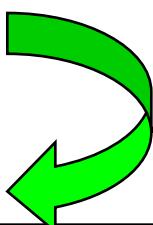
Selektoren

```
public class Rechteck {  
    //Objektattribute  
    double laenge, breite;      int strichstaerke;  
  
    //Selektoren  
    public void setLaenge (double x) {  
        laenge = x; }  
  
    public double getLaenge () {  
        return laenge; }  
  
    ...  
}
```

```
Rechteck r, s;
```

```
...
```

```
r.laenge = s.laenge + 2;
```



```
r.setLaenge (s.getLaenge() + 2);
```

Selektoren

```
public class Rechteck {  
    //Objektattribute  
    double flaeche, breite;      int strichstaerke;  
  
    //Selektoren  
    public void setLaenge (double x) {  
        flaeche = x * breite; }  
  
    public double getLaenge () {  
        return flaeche / breite; }  
    ...  
}
```

```
Rechteck r, s;
```

```
...
```

```
r.setLaenge (s.getLaenge() + 2);
```

Zugriffsmodifikatoren

Einschränkung des Zugriffs auf Attribute und Methoden:

- **private:**

Komponente nur innerhalb der Klasse bekannt

- **kein Schlüsselwort:**

Komponente nur innerhalb des Pakets bekannt

- **public:**

Komponente überall bekannt

Zugriffsmodifikatoren

```
public class Rechteck {  
  
    //Objektattribute  
    private double laenge, breite; private int strichstaerke;  
  
    //Selektoren  
    public void setLaenge (double x) {  
        laenge = x; }  
  
    public double getLaenge () {  
        return laenge; }  
  
    ...  
}
```

```
Rechteck r, s;
```

```
...
```

```
r.laenge = s.laenge + 2;
```

nicht mehr
möglich!

```
r.setLaenge (s.getLaenge() + 2);
```

Schnittstellendokumentation

Class Rechteck

java.lang.Object
Rechteck

public class Rechteck
extends java.lang.Object

Constructor Summary

Constructors

Constructor

Rechteck()

Description

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type

Method

double

getLaenge()

void

setLaenge(double x)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Details

Rechteck

public Rechteck()

Method Details

setLaenge

public void setLaenge(double x)

getLaenge

public double getLaenge()

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	
double		getLaenge()
void		setLaenge(double x)

Method Details

setLaenge

public void setLaenge(double x)

getLaenge

public double getLaenge()

Beispiel: Datentyp Ordner

■ "Ordner" als Konzept

- **enthält** Texte
- Texte können **abgelegt** und **entnommen** werden
- ein Ordner kann **beschriftet** werden
- ein Ordner kann **leer** oder **voll** sein

Abstrakte
Beschreibung
des Konzepts Ordner

Datenkapselung als
Entwurfsprinzip!

■ Schnittstelle:

```
void legeTextAb (String t)
String entnehmeText ()
boolean istVoll ()
boolean istLeer ()
void beschrifte (String t)
String liesBeschriftung ()
```

Verwendung des Datentyps Ordner

```
Ordner o = new Ordner () ;

o.beschrifte ("Kleine Gedichte") ;

if    (o.istVoll ())
    IO.println("Ordner ist bereits voll");
else o.legeTextAb ("Herr Ribeck auf Ribeck . . .");

if    (o.istVoll ())
    IO.println("Ordner ist bereits voll");
else o.legeTextAb ("Von drauss vom Walde komm ich her . . .");

IO.println (o.liesBeschriftung ());
IO.println ("-----");

if (!o.istLeer()) IO.println (o.entnehmeText ());
if (!o.istLeer()) IO.println (o.entnehmeText ());
```

Implementierung des Datentyps Ordner

```
/** Datentyp Ordner zur Speicherung von Texten
 * @author Juergen Giesl
 */
public class Ordner {

    private static final int maxTexte = 20;
    private String [] ordnerInhalt = new String [maxTexte];
    private int anzahlTexte = 0;
    private String beschriftung = "";

    /** @return true, falls der Ordner voll ist, sonst false
     */
    public boolean istVoll () {
        return anzahlTexte == maxTexte;
    }

    /** @return true, falls der Ordner leer ist, sonst false
     */
    public boolean istLeer () {
        return anzahlTexte == 0;}
```

Es wird ein Array verwendet

Implementierung des Datentyps Ordner

```
/** @param t Text, der vorne im Ordner abgelegt wird */
public void legeTextAb (String t) {
    ordnerInhalt [anzahlTexte] = t;
    anzahlTexte++;
}

/** Liest zuletzt eingegebenen Text und loescht ihn.
 * @return letzten abgelegten Text */
public String entnehmeText () {
    String t = ordnerInhalt [anzahlTexte-1];
    ordnerInhalt [anzahlTexte-1] = "";
    anzahlTexte--;
    return t;
}

/** @param t Beschriftung des Ordners */
public void beschrifte (String t) {
    beschriftung = t;
}

/** @return Beschriftung des Ordners */
public String liesBeschriftung () {
    return beschriftung;
}
```

Schnittstellendokumentation

Constructor Summary

Constructors

Constructor and Description

Ordner()

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
void	beschreibe(java.lang.String t)
java.lang.String	entnehmeText()
	Liest zuletzt eingegebenen Text und loescht ihn.
boolean	istLeer()
boolean	istVoll()
void	legeTextAb(java.lang.String t)
java.lang.String	liesBeschriftung()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Ordner

public Ordner()

Method Detail

istVoll

public boolean istVoll()

Returns:
true, falls der Ordner voll ist, sonst false

istLeer

public boolean istLeer()

Returns:
true, falls der Ordner leer ist, sonst false

legeTextAb

public void legeTextAb(java.lang.String t)

Parameters:
t - Text, der vorne im Ordner abgelegt wird

Method Detail

Method S

All Methods

Modifier

void

java.lang.String

boolean

boolean

void

java.lang.String

istVoll

public boolean istVoll()

Returns:
true, falls der Ordner voll ist, sonst false

istLeer

public boolean istLeer()

Returns:
true, falls der Ordner leer ist, sonst false

legeTextAb

public void legeTextAb(java.lang.String t)

Parameters:
t - Text, der vorne im Ordner abgelegt wird