

Einführung in die Programmierung

by André Karge
Übung - Packages, Kapselung

Heute

- Packages
- Kapselung

Anmerkungen

Umlaute im Programmcode

- Bitte verwendet nirgendwo in eurem Programm Umlaute (vor allem nicht als Bezeichner von Variablen, Objekten oder Klassen)
- Wenn ihr trotzdem eure Variablen in Deutsch benennen wollt, nutzt die International verfügbaren Buchstaben des lateinischen Alphabets: ä → ae, ü → ue, ö → oe, ß → ss
- Wenn ihr eine andere Dateiformatierung verwendet, als derjenige, der euer Programm compiled, werden Umlaute falsch dargestellt und der Compiler bricht ab.

Packages

Packages - Insel 3.6

- muss in definiertem Schema geschrieben werden (siehe Vorlesung)
- Pakete können überall eingebunden werden (damit alle Klassen, die das Paket kapselt)
- Pakete und imports stehen immer am Anfang von Java-Dateien
- Stichwort *import*
- **Wichtig:** Die Packages geben einen Ordnerpfad an
- Bspw: *package de.uni_weimar.medien.karge.proggen* muss im Ordner *meinProjekt/src/de/uni_weimar/medien/karge/proggen* liegen

Beispiel - Import

```
import java.util.Random; // lädt das Paket Random aus util
//import java.util.*; //lädt alle Pakete in util
public class MyClass {
    // attribute
    // methoden
}
```

Packages - Insel 3.6

Beispiel - Paketdefinition

```
package de.uni_weimar.medien.karge.proggen; //definiert, dass die folgende Klasse teil dieses Pakets ist
public class MyClass {
    public static void sayHello() {
        System.out.println("Hello");
    }
}
```

Beispiel - Einbinden des Pakets (in einem anderen Paket)

```
package de.uni_weimar.medien.karge.apps

import de.uni_weimar.medien.karge.proggen.MyClass; // Einbindung der Klasse
//import de.uni_weimar.medien.karge.proggen.*; // Einbindung des gesamten Packages

public class Test {
    public static void main(String[] args) {
        MyClass.sayHello(); // Aufruf der statischen Methode sayHello aus der Klasse im Paket
    }
}
```

Packages - Insel 3.6

Übungsaufgabe - (10min)

1. Erstellen Sie ein neues Projekt mit 2 Packages:
de.uni_weimar.medien.proggen.application und *de.uni_weimar.medien.proggen.aufg1*
2. Erstellen Sie zwei Klassen im Paket *aufg1*: *Person* und *Guess*, mit jeweils einer statischen Methode
3. Erstellen Sie eine Klasse *Application* im Paket *application* mit einer main-Methode
4. Importieren Sie das Paket *aufg1* in der Datei der Klasse *Application* und testen Sie die statischen Methoden der importierten Klassen

Packages - Insel 3.6

Standard Packages

- Java bietet eine Hand voll Pakete, die Standardmäßig ausgeliefert werden
- Beispiele:
 - ▶ java.lang (fundamentale Java Klassen wie String)
 - ▶ java.util (Hilfsklassen wie Random)
 - ▶ java.io (System Klassen wie input und output Streams)
 - ▶ java.text (Klassen zur Arbeit mit Text)
- Eine vollständige Liste für alle Standardpaketen findet ihr hier: [Link](#)

Kapselung

Kapselung

- Dient dem *Information Hiding*
- haben wir schon letzte Woche gesehen
- wir verbieten den direkten Zugriff auf bestimmte Attribute und Methoden und erlauben den Zugriff nur über explizite Funktionen
- Ein Teil davon: private Attribute mit Getter & Setter Methoden
- Beispielsweise für Modellierung von Kontos:

Kapselung

```
class Konto {  
    double geld;  
    public Konto() {...} // Constructor  
    public void überweiseGeldAn(double geld,  
                                Konto ziel) {  
        ziel.geld += geld;  
        this.geld -= geld;  
    }  
}
```

Kapselung

```
class Konto {  
    double geld;  
    public Konto() {...} // Constructor  
    public void überweiseGeldAn(double geld,  
                                Konto ziel) {  
        ziel.geld += geld;  
        this.geld -= geld;  
    }  
}
```

- Direkter Zugriff auf das Geld in den Kontos
- Ein Kontoinhaber kann demnach den Kontostand von jedem anderen Konto einsehen
- schlecht!

Kapselung

```
class Konto {  
    double geld;  
    public Konto() {...} // Constructor  
    public void überweiseGeldAn(double geld,  
                                Konto ziel) {  
        ziel.geld += geld;  
        this.geld -= geld;  
    }  
}
```

- Direkter Zugriff auf das Geld in den Kontos
- Ein Kontoinhaber kann demnach den Kontostand von jedem anderen Konto einsehen
- schlecht!

```
class Konto {  
    private double geld; // information hiding  
    public Konto() {...} // Constructor  
    public boolean geldEinzahlen(double betrag) {  
        if(betrag > 0) { // check geld nicht negativ  
            this.geld += betrag;  
            return true;  
        }  
        return false;  
    }  
    public boolean geldAbheben(double betrag) {  
        // verschiedene checks  
        if(betrag > 0 && this.geld - betrag >= 0) {  
            this.geld -= betrag;  
            return true;  
        }  
        return false;  
    }  
    public void überweiseGeldAn(double betrag,  
                                Konto ziel) {  
        if(this.geldAbheben(betrag) && betrag > 0)  
            ziel.geldEinzahlen(betrag);  
    }  
}
```

Fragen?