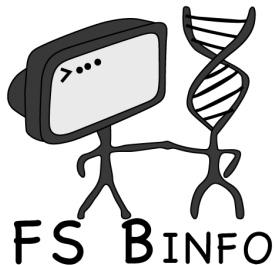


Vererbung, Packages, allgemeine Tipps



Fabian Schmich

Fachschaft Bioinformatik

19. Januar 2010

Zeitplan

- 27.10 Einführung in Linux und die Shell
 - 03.11 Arbeiten mit Dateien auf der Shell
 - 10.11 Einführung in Java
 - 17.11 Umgang mit Arrays, Schleifen und Dateien
 - 24.11 Einführung in objekt-orientierte Programmierung
 - 01.12 Rekursion und Iteration
 - 08.12 File-I/O, Schleifen, Javadoc
 - 15.12 Programmier-Aufgabe zur Wiederholung
-
- 12.01 Debugging
 - 19.01 **Vererbung, Packages, allgemeine Tipps**
 - 26.01 Wiederholung

Webseite: <http://www.bioinformatik-muenchen.com/bioinfocom/informatik-tutorium>



Der Plan

- Programmstrukturierung durch Packages
- Objektmodellierung mit Vererbung
- Nützliche Eclipse Shortcuts
- Eclipse *Run Configurations* und Java auf der Konsole
- Linksammlung

Programmstrukturierung durch Packages

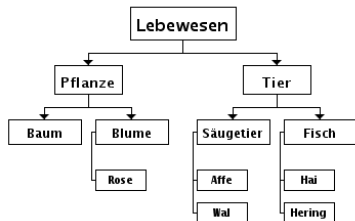
- Organisation von Klassen Sourcecode in hierarchischen Strukturen
- Ausnutzen von sog. *namespaces*: z.B. Benutzen des selben Klassennamens in verschiedenen Packages (`java.util.Date` bzw `java.sql.Date`)
- Zugriffskontrolle:

Modifier	Class	Package	Subclass	World
<code>public</code>	Y	Y	Y	Y
<code>protected</code>	Y	Y	Y	N
no modifier	Y	Y	N	N
<code>private</code>	Y	N	N	N



Vererbung I

- Objekte stehen oft in hierarchischen *ist-eine-Art-von* Beziehungen
- Klasse wird durch Schlüsselwort *extends* zur *Subklasse* einer *Superklasse*
- Ohne explizite *extends* Anweisung erbt jede Klasse von *Object*

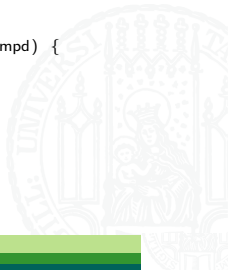


Vererbung II

- Klassen erben sowohl Attribute als auch Methoden
- Attribute und Methoden müssen jedoch *sichtbar* sein (`public` oder falls im selben package `protected`)
- Durch die Methode `super()`; wird der Konstruktor der Superklasse aufgerufen
- Geerbte Attribute und Methoden können *überlagert* werden
- Java hat keine richtige Mehrfachvererbung, Abhilfe durch *Interfaces*

Beispiel Vererbung: Katze

```
1 public class Animal {
2     String habitat;
3     int age;
4     Animal prey;
5
6     public Animal (String habitat , int age, Animal prey) {
7         this.habitat = habitat;
8         this.age = age;
9         this.prey = prey;
10    }
11 }
12
13 public class Cat extends Animal {
14     String furColour;
15     int micePerDay;
16
17     public Cat(String habitat , int age, Animal prey , String fur , int mpd) {
18         super(habitat , age, prey);
19         this.furColour = fur;
20         this.micePerDay = mpd;
21     }
22 }
```



Übung zu Vererbung: Basisklasse Person

- Person hat einen Namen
- Instanzvariablen sollen `private` sein
- Implementiere `getter` und `setter` Methoden
- Implementiere eine einfache Methode `print`, die Person auf der Konsole ausgibt
- Implementiere eine Methode `equals`, die Person mit einer anderen Person vergleicht und einen `boolean` zurück gibt
- Initialisiere ein Person Object in der `main` Methode und benutze die implementierten Methoden

Übung zu Vererbung: Erweiterte Klasse Student

- Ein Student ist eine Person und erbt damit Attribute und Methoden von Person
- Student hat eine Matrikelnummer
- Implementiere **nötige** *getter* und *setter* Methoden
- Überlagere die Methode *equals*, die dann Student mit einem anderen Student vergleicht
- Implementiere eine *main* Methode und benutze sowohl geerbte als auch neue Methoden von Student

args der main Methode und Eclipse *Run Configurations*

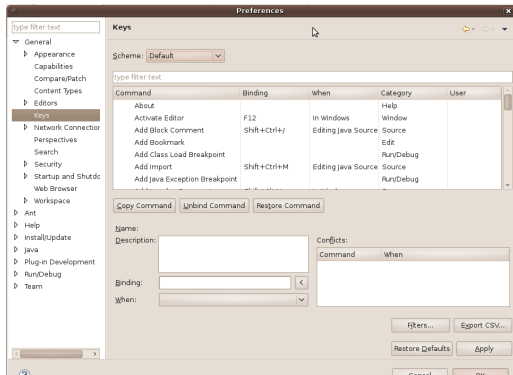
- Signatur der main Methode: `public static void main(String[] args)`
- args ist ein Array von Strings, die dem Programm (z.b. per Kommandozeile) übergeben wurden
- args ist damit das übliche Mittel um User Input zu verarbeiten
- In Eclipse kann man Kommandozeilen Argumente mit *Run* → *Run Configurations...* → *Arguments* übergeben

Übung args

- Ändere die `main` Methode, so dass sie einen Namen und eine Matrikelnummer in `args` erwartet
- Speichere Namen und Matrikelnummer aus `args` in Variablen und erzeuge Student Objekt
- Teste mit den *Eclipse Run Configurations*

Eclipse Shortcuts I

- Shortcuts machen das Leben einfacher, weil man schneller programmiert
- Anpassen und nachlesen von Shortcuts unter *Window* → *Preferences* → *General* → *Keys*



Eclipse Shortcuts II: Auswahl

Shortcut	Funktion
Ctrl+Shift+O	Autom. Hinzufügen von Imports
Ctrl+Shift+F	Autom. Code Formatierung
F3	Springe zu Deklaration
Alt+Shift+R	Klasse/Attribut/Methode umbenennen
Ctrl+Space	Kontext Assistent
Str+7	Aktuelle Zeile auskommentieren

Sammlung nützlicher Links und Co

- Java Insel:
<http://openbook.galileocomputing.de/javainsel8/index.htm>
- Java 6.0 Dokumentation:
<http://java.sun.com/javase/6/docs/>
- LMU PST Zusatzangebot:
<http://www.pst.ifi.lmu.de/Lehre/wise-09-10/zusatzangebot>
- Fachschaftsseite mit Forum:
<http://www.bioinformatik-muenchen.com/bioinfocom/>
- Fachschafts email:
fsbinfo@lists.lrz-muenchen.de