



Informatik I

Exercise session 8

Autumn 2020

Homework

■ Questions?

Heutiges Lernziel: Klassen, Objektorientierung.

Kahoot.

Code formatting

Why good formatting is important?
(exercise by Max Rossmannek)

Klassen - Technisch

Eine Klasse ist eine Einheit mit einem Namen, die Daten und Funktionalität beinhaltet

OrderLine → OrderLine.java

- Die Klasse definiert einen neuen Datentyp.
- Daten sind Variablen und heissen Felder oder Attribute.
- Funktionalität ist vorhanden in Form von Methoden, die in der Klasse definiert sind.
- Klassen sind (typischerweise) separate .java Dateien mit gleichem Namen

Name
■ feld1
■ feld2
■ ...
■ methode1
■ methode2
■ ...

Klasse für Messwert - Erster Versuch

date	time	appraisal	event type	lat [°N]	lon [°E]	source agencygency	depth	Mw
2001/01/03	11:11:20.4	certain	earthquake	46.446	9.982	SED (ECOS- 09)	4.2	3.6

Datei Measurement.java:

```
public class Measurement {  
  
    String date;  
    String time;  
  
    double latitude;  
    double longitude;  
  
    float magnitude;  
  
}
```

Measurement

- String date
- String time
- double latitude
- double longitude
- float magnitude

Methoden in Klassen

```
public class Coordinate {  
  
    double latitude;  
    double longitude;  
  
    /**  
     * Computes the distance to the provided coordinate 'other'.  
     */  
    double distanceTo(Coordinate other){  
        double dl = this.latitude - other.latitude;  
        // complete this as exercise...  
    }  
}
```

Methoden in Klassen

```
public class Coordinate {  
  
    double latitude;  
    double longitude;  
  
    /**  
     * Computes the distance to the provided coordinate 'other'.  
     */  
    double distanceTo(Coordinate other){  
        double dl = this.latitude - other.latitude;  
        // complete this as exercise...  
    }  
}
```


Datenkapselung / Information Hiding

Steuern, welche Daten und welcher Code woher zugänglich ist.

Zugriffsmodifikatoren:

- **private**: Sichtbar aus Code derselben Klasse
- **protected**: Sichtbar aus Code derselben Klasse oder Unterklasse (später)
- **public**: Von überall sichtbar

default / Standard

(package)

Name

- **private** field1
- **protected** field2
- ...

- **private** method1
- **public** method2
- ...

Koordinate: Zugriffsmethoden

```
public class Coordinate {  
    private double latitude;  
    private double longitude;  
  
    public double getLatitude(){  
        return latitude;  
    }  
  
    public void setLatitude(double lat){  
        assert lat >= -90 && lat <= 90;  
        this.latitude = lat;  
    }  
    //...
```

Use case

Products and orders

Keep track of available products, prices, and of customers' orders

Products and orders

Implement an order-management system that tracks available products, prices, orders (with multiples of the same product in the same order), and prices.

The system should automatically calculate the total price of each order.

Products and orders: usage

Design the classes so that, in the `main` method:

- we can construct an array of products (with their prices);
- we can construct an array of orders;
- we can add items to the orders;
- we can print out products and orders (and their prices).

Products and orders: change prices

In `main`, add code to change the price of a product and print the orders and products again.

