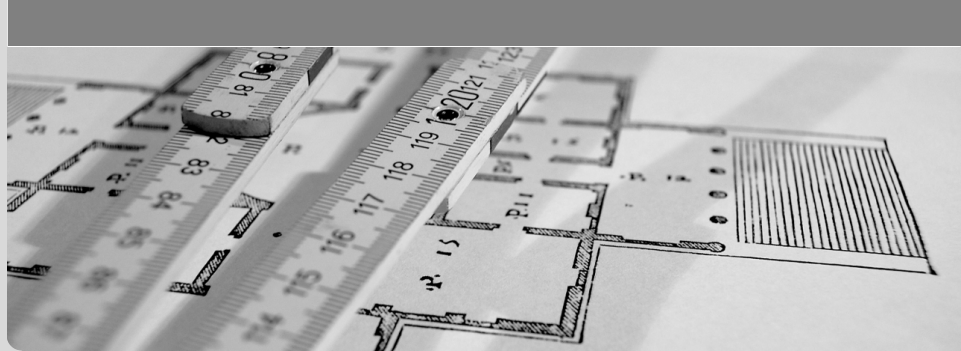


Programmieren

04. Tutorium

1. Übungsblatt, Diverses, Wiederholung, Methoden, Kontrollstrukturen, Übungen
Robin Rüde | 24. November 2014



- 1 Organisatorisches
- 2 Eclipse und Checkstyle
- 3 Arrays
 - Arrays
 - Matrizen
- 4 Graphen
 - Definition und Darstellung im Rechner
- 5 Packages
 - Definition und Einsatz
- 6 Diverses
 - Einlesen von Dateien
 - Typkonvertierung
- 7 Aufgaben

1

Organisatorisches

- IDEs dürfen für das **nächste** Blatt verwendet werden
- Empfehlung: Eclipse Luna
- siehe Ilias-Wiki (bald)

2

Eclipse und Checkstyle

- IDEs wie Eclipse vereinfachen einiges (Build-Zyklus, Fehlermeldungen, Vorschläge)
- Checkstyle ist zur Kontrolle von Stil-Richtlinien
- Eclipse-Checkstyle-Plugin aus dem Marketplace
- Checkstyle-Datei wird für nächstes Blatt gestellt
- Beispiel: Einfaches Hello-World Projekt in Eclipse

3

Arrays

- Sammlung von Objekten/Werten gleichen Typs
- feste Größe!
- Zugriff auf Elemente über Index: `name[index]`
- Array der Länge `n` hat Indizes von 0 bis `n-1`
- `name.length` liefert Länge des Arrays
- Behandlung intern wie Objekte (Referenzen!)

Beispiel:

```
// Deklaration:
```

```
Typ[] name;
```

```
// Deklaration + Laenge n angeben:
```

```
Typ[] name = new Typ[n];
```

```
// Deklaration mit Werten und Laenge n:
```

```
Typ[] name = { v0, ..., vn-1 };
```



```
class ArrayExample {  
    public static void main(String[] args) {  
        int[] a = new int[10];  
        a[0] = 1;  
        a[1] = 1;  
        for (int i = 2; i < 10; i++) {  
            a[i] = a[i-1] + a[i-2];  
        }  
        printArray(a);  
    }  
  
    public static void printArray(int[] a) {  
        for (int b: a) {  
            System.out.println(b);  
        }  
    }  
}
```

Was gibt das Programm aus?

- mehrdimensionale Arrays
- Matrizen können mit verschachtelten Arrays dargestellt werden
- Zuerst Zeile, dann Spalte
- Iteration durch geschachtelte for-Schleifen

```
int[] [] a = new int[10][10]; // 10 * 10 - Matrix
int[] s3 = a[3]; // Zugriff 4. Spalte
int x = a[3][1]; // Zugriff 4. Spalte, 2. Zeile
// Iteration:
for (int i = 0; i < a.length; i++) {
    for (int j = 0; j < a[i].length; j++) {
        a[i][j] // i+1. te Zeile und j+1. te Spalte
    }
}
```

4

Graphen

- Graphen haben Knoten und Kanten
- Gerichtete Graphen: Kanten mit Pfeil
- Ungerichtete Graphen: Kanten ohne Pfeil
- siehe Tafel

- Adjazenzmatrix:
 - $|\text{Knoten}| * |\text{Knoten}|$ Matrix
 - Knoten $i \rightarrow j$ gerichtet verbunden: $a[i][j] = 1$
 - Ungerichtet verbunden: $a[i][j] = 1 = a[j][i]$;
- weitere Darstellungen in GBI und Algorithmen

5

Packages

- Klassen können über **Pakete** gruppiert werden
- Schlüsselwort `package paketname;` am Anfang der java-Datei
- Pakete entsprechen Verzeichnissen im Dateisystem
- Konventionen
 - Paketnamen werden klein geschrieben
 - Umkehrung der eigenen Domain und dann ein Paketname:
`edu.kit.paketname`
- Zugriff auf Klassen im gleichen Paket wie gehabt

- Verwendung des qualifizierten Namens der Klasse
- Beispiel
 - Paketname: edu.kit.geometry
 - Klasse darin: Vector
 - Verzeichnis: /edu/kit/geometry, Datei: Vector.java

```
package edu.kit.geometry;  
public class Vector { ... }
```

Verwendung:

```
edu.kit.geometry.Vector myVector;  
myVector = new edu.kit.geomtry.Vector();
```


- Zugriff über qualifizierten Namen umständlich
- deswegen: import von packages
- Syntax: import paketname;
- **Muss** direkt nach der eigenen package-Deklaration stehen

Beispiel:

```
import edu.kit.geomtry.Vector;  
// ...  
    public static void main(String[] args) {  
        Vector myVector = new Vector();  
    }  
// ...
```

6

Diverses

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

class ReadFile {
    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Wrong usage");
            return;
        }
        String filename = args[0];

        try {
            BufferedReader reader = new BufferedReader(new FileReader(filename));
            while (reader.ready()) { // lines remaining
                String line = reader.readLine();
                System.out.println(line); // or some method call ;-)
            }
        } catch (IOException e) { // in case of errors (needed!)
            System.out.println("Something went wrong");
        }
    }
}
```

```
int a = 3;  
short b = 4;  
int c = a + b;  
System.out.println(c);
```

- b wird automatisch in int konvertiert.
- Nie Informationsverlust des Wertes
- Verlust von Metainformationen bzgl. Präzision/Klasse
- Jeder Datentyp kann nach String konvertiert werden, da
- jede Klasse implizit eine toString()-Methode hat

```
double d = 5.5;  
int x = (int) d; // x == 5
```

- Cast-Operator: (<Typ>)
- Fehler beim Kompilieren oder Ausführen wenn Cast nicht möglich

7

Aufgaben

Aufgabe: Kleines Einmaleins

- Level 0:
Schreibe ein Programm, das alle Zahlen zwischen 1 und 10 ausgibt.
- Level 1:
Schreibe ein Programm, das alle Kombinationen von Zahlen zwischen 1 und 10 ausgibt. (*doppelte for-Schleife*)
- Level 2:
Schreibe ein Programm, das das Einmaleins in der Form " $x * y = z$ " ausgibt, wobei x und y hintereinander alle Zahlen zwischen 1 und 10 sind.
- Level 3:
Schreibe ein Programm, das das Einmaleins in Tabellenform ausgibt
- Level 4:
Schreibe ein Programm, das das Einmaleins als korrekt ausgerichtete Tabelle mit | als Rändern ausgibt

- Level 1:
Schreibe eine Schleife, die mithilfe der Funktion `Math.sin` die Sinuswerte aller Zahlen von 0 bis 2π in 0.1-Schritten ausgibt.
- Level 2:
Passe den Wertebereich der Sinuskurve so an, dass die Ausgabe nur Zahlen zwischen 0 und 30 enthält.
- Level 3:
Schreibe ein Programm, das eine Sinuskurve in die Konsole malt.

- Implementiere eine komponentenweise Matrixaddition für 2 Matrizen a und b .
- Beachte: Matrixaddition ist nur für zwei gleichgroße Matrizen definiert
- Gib einen Fehler aus, falls eine Matrixaddition nicht möglich ist.
- Überprüfe deinen Code durch Eingabe von 2 Matrizen und Ausgabe der korrekten berechneten Matrix

Ende

Fragen?

Fragen?
Vielen Dank für eure Aufmerksamkeit!