

Große Übung

Praktische Informatik 1

2006-01-11

Holger Füßler

fuessler@informatik.uni-mannheim.de

<http://www.informatik.uni-mannheim.de/pi4/people/fuessler>

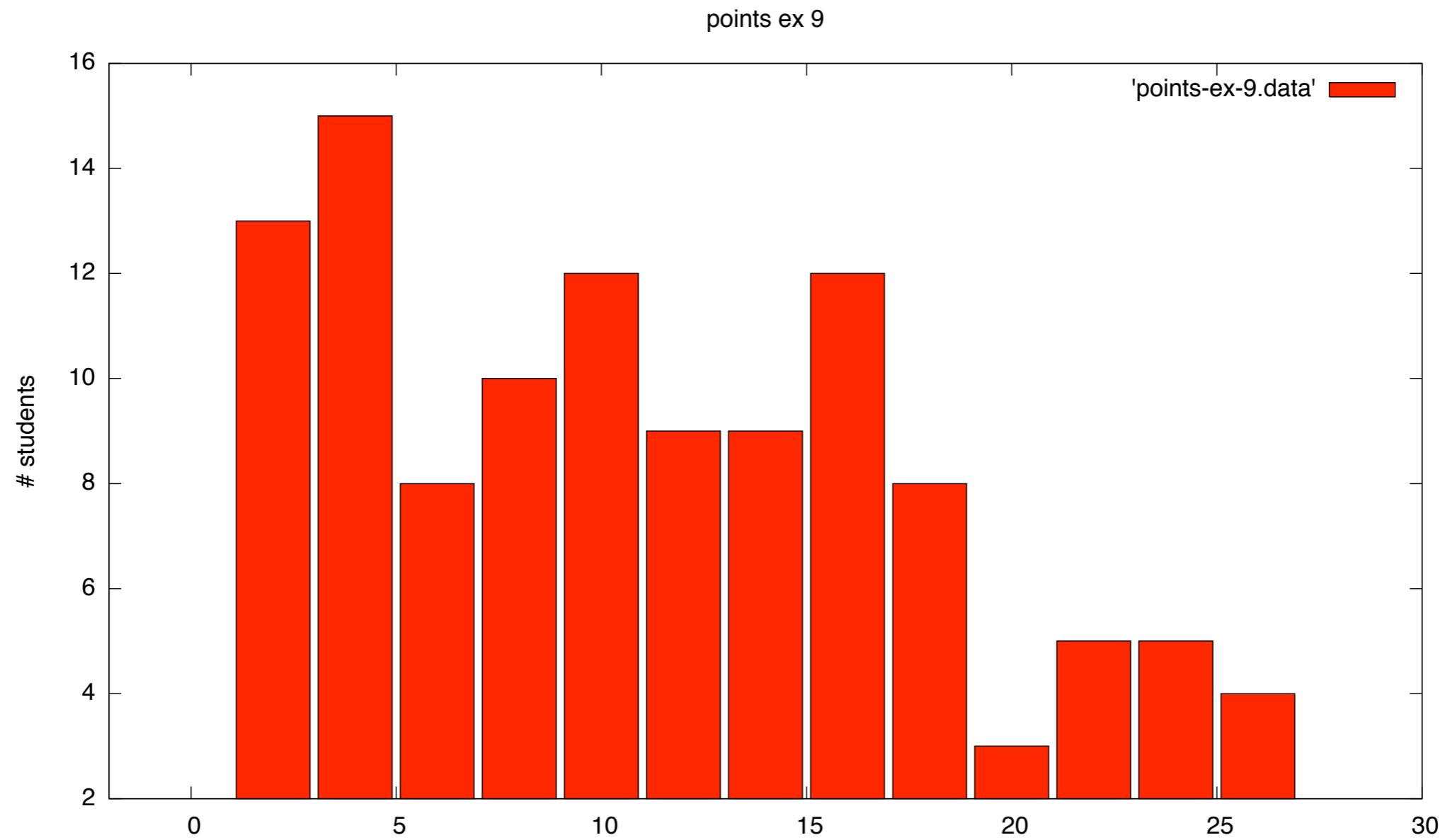
1: Announcements / Orga

Weihnachtsklausur

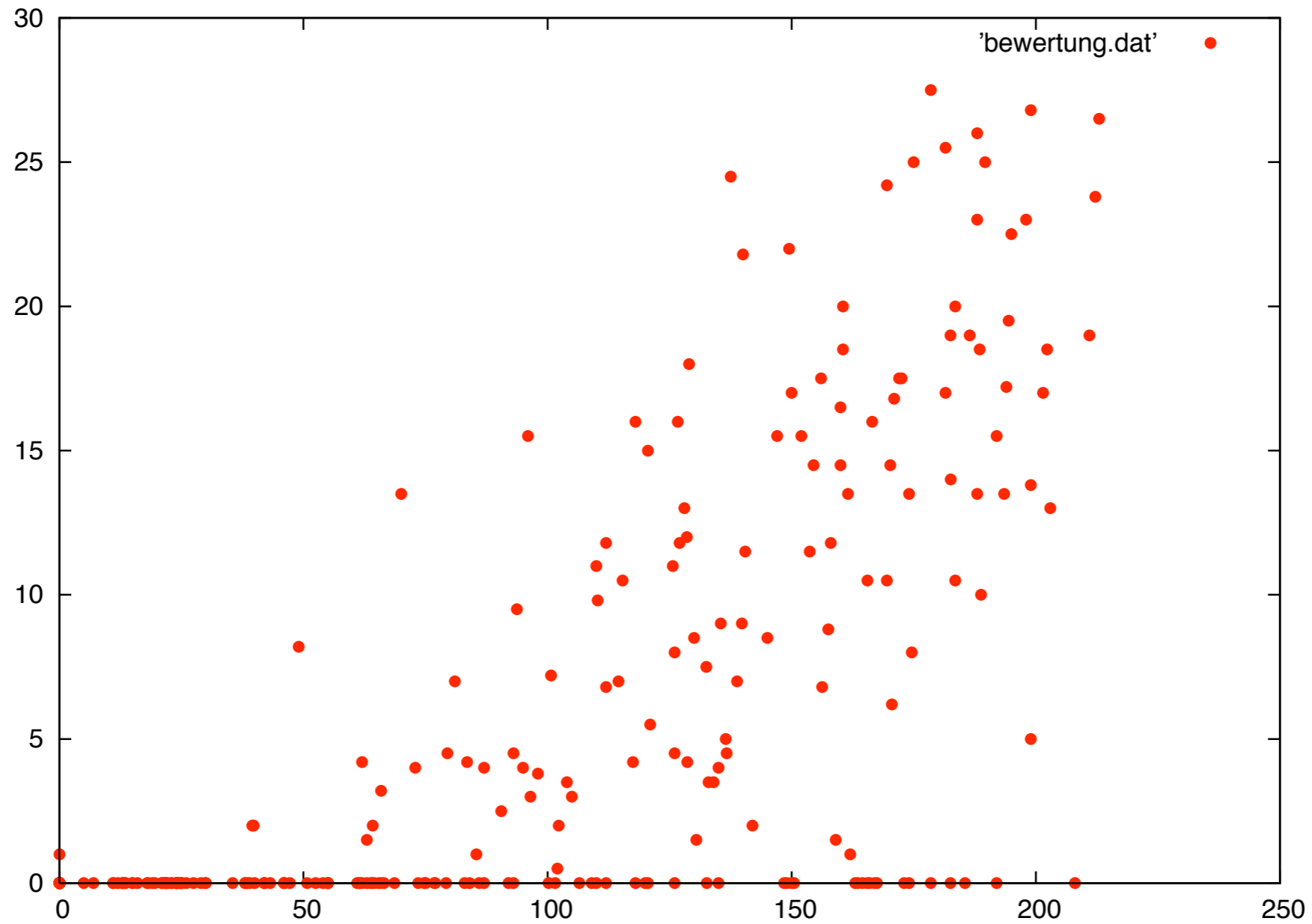


- ist ziemlich schlecht ausgefallen
- DESHALB: Auf den folgenden Übungsblättern jeweils *-Punkte, d.h. Punkte, die nicht im Nenner zählen, so dass auch bei schlechter Weihnachtsklausur das Scheinzulassungskriterium erreicht werden kann.

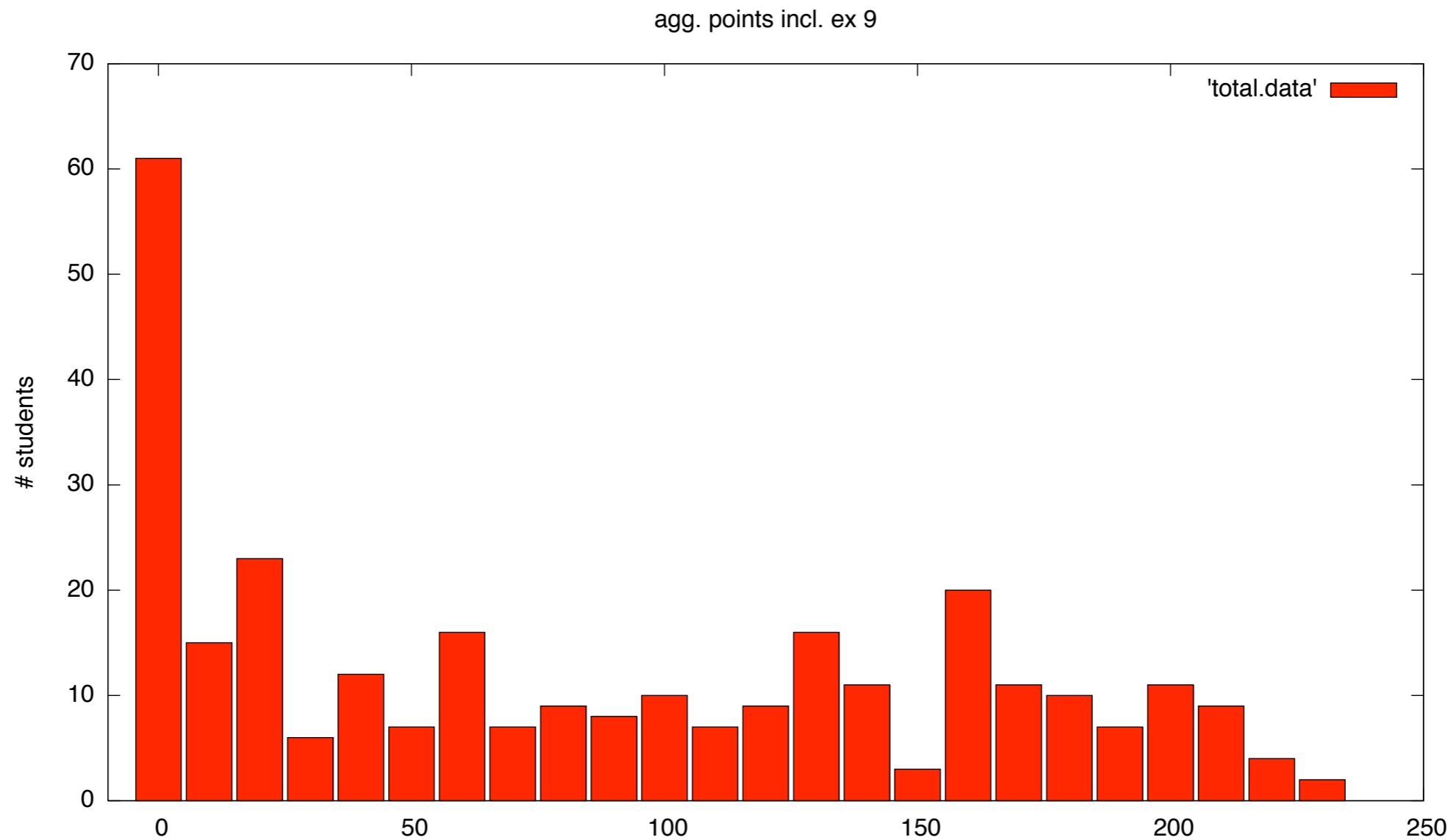
Statistiken



Korrelation



Statistik Gesamt (bis 7)



Konsequenzen

- Überschlagen Sie mal, wie viele Punkte sie noch erreichen können. Noch zwei Blätter (11 + 12) mit ca 30 + 10*-Punkte
- die benötigte Punktzahl wird so ungefähr $391 / 2 = 195,5$ sein
- wenn Sie so in dem Bereich liegen, in dem Sie das noch schaffen können,
 - sprechen Sie Ihren Tutor an
 - geben Sie Vollgas, wenn Sie den Schein brauchen

2: Allgemeines zum Lösen von Klausuren

Zeiteinteilung

- beachten Sie die vorgesehenen Zeiten
- Sortieren Sie Augabentypen z.B.
 1. *Fehlersuchaufgaben*
 2. *Java-Analyse-Aufgaben*
 3. *Algorithmen-Entwicklung*
- Bestes Punkte / Zeit Verhältnis zuerst

Für jede Aufgabe

- schreiben Sie **immer** etwas hin
- definieren Sie geschickt “Probleme weg”

Generell

- Machen Sie **keine** formalen Fehler
 - Namen vergessen
 - Lösungen auf's falsche Blatt geschrieben
 - etc...

3: Kritik an der Klausur

Kritik-Sammlung

- zu schwer, zu wenig Zeit
- nur Transfer-Aufgaben
- mehr kleine Teilaufgaben
- langer Text schwer zu erfassen
- Löschen in B-Bäumen widersprüchlich definiert
- Primzahl-Aufgabe zu schwer
- veralbert vorgekommen (Yankee, Lima etc.)

zu schwer, zu wenig Zeit

- das war z.T. Absicht
- das wird an der Uni oft passieren
- verabschieden Sie sich vom “Hundert-Prozent-Anspruch”
- i.d.R. mit 40-50% bestanden, mit 85-90% 1,0

Nur Transfer-Aufgaben

- Reproduktions-Aufgaben sind in der Informatik schwer zu machen
- es geht um Begreifen, nicht um Wissen
- eigentlich war nur Aufgabe 7 Transfer, der Rest war eingeübt

Primzahl, veralbern

- Veralbern war wirklich keine Absicht
- Primzahl-Aufgabe war natürlich die Schwerste von allen, obwohl ich nicht geahnt hätte, dass Sie sie sooo schwer empfinden
- Auch hier gilt: Oft ist eine Aufgabe dabei für Einser-Kandidaten

B-Bäume löschen

- mir war nicht ganz bewusst, wie lückenhaft das Skript diesbezüglich ist
- die ex-ante Kritik ist nicht bis zu Herrn Schiele und mir gedrungen

Konsequenzen für uns

- weniger Aufgaben insgesamt (vielleicht so vier oder fünf)
- kleinere Teilaufgaben
- mehr “Sachen aus der Vorlesung erklären”
- kein B-Baum-Löschen
- wir bemühen uns um mehr Übersichtlichkeit

Trotzdem

- Wesentlich für Sie ist das
- Begreifen der Vorlesung und Übung
- Anwenden und Verstehen von Java
- Bessere Vorbereitung nötig

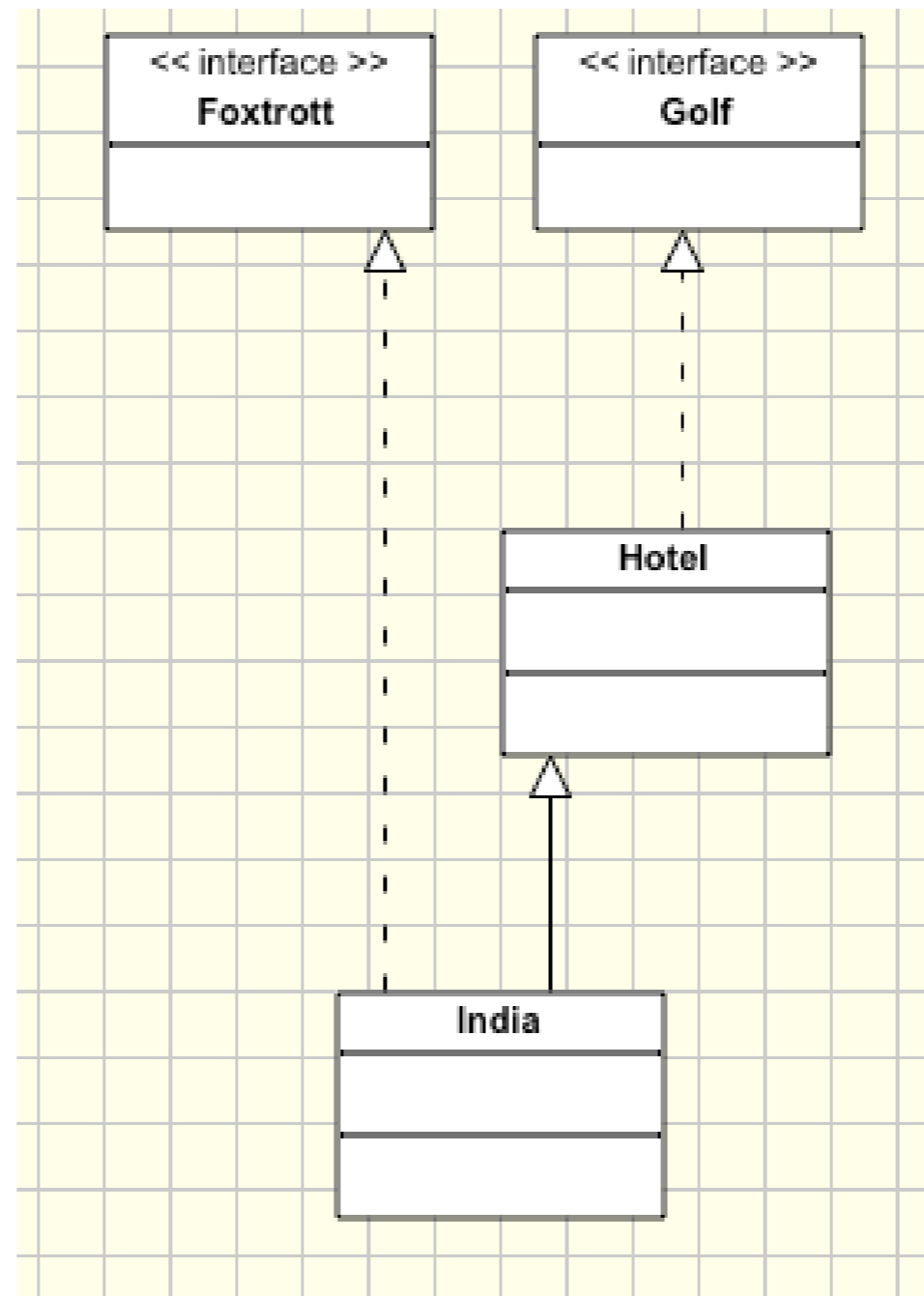
4: Klausur-Besprechung

Aufgabe 5

- wirklicher Typ
 - lima: India
 - mike:India
- Schnittstelle
 - lima: Golf
 - mike: Hotel

```
41 class Oscar {  
    public static void main(String papa[]) {  
        Golf lima = new India();  
        Hotel mike = (India)lima;  
        lima.alfa("X-Ray");  
46        mike.alfa("Yankee");  
        mike.bravo();  
        mike.charlie ();  
        India.delta ();  
    }  
51 }
```

Struktur



lima.alpha("X-Ray")

```
1 interface Foxtrot {
    String echo = "Uniform";
    void alfa ();
}

6 interface Golf {
    String echo = "Victor";
    void alfa(String juliett );
}

41 class Oscar {
    public static void main(String papa[]) {
        Golf lima = new India();
        Hotel mike = (India)lima;
        lima.alfa("X-Ray");
46 mike.alfa("Yankee");
        mike.bravo();
        mike.charlie ();
        India.delta ();
    }
51 }
```

```
11 class Hotel implements Golf {
    public void alfa(String kilo) {
        System.out.println("Quebec");
    }
    public static void bravo() {
16 System.out.println("Romeo");
    }
    public void charlie() {
        bravo();
    }
21 public static void delta() {
        System.out.println(echo);
    }
}

26 class India extends Hotel implements Foxtrot {
    public void alfa() {
        System.out.println("Sierra");
    }
    public static void bravo() {
31 System.out.println("Tango");
    }
    public void charlie() {
        bravo();
    }
36 }
```

mike.alpha("Yankee")

```
1 interface Foxtrot {
    String echo = "Uniform";
    void alfa ();
}

6 interface Golf {
    String echo = "Victor";
    void alfa(String juliett );
}

41 class Oscar {
    public static void main(String papa[]) {
        Golf lima = new India();
        Hotel mike = (India)lima;
        lima.alfa("X-Ray");
46     mike.alfa("Yankee");
        mike.bravo();
        mike.charlie ();
        India.delta ();
    }
51 }
```

```
11 class Hotel implements Golf {
    public void alfa(String kilo) {
        System.out.println("Quebec");
    }
    public static void bravo() {
16     System.out.println("Romeo");
    }
    public void charlie() {
        bravo();
    }
21     public static void delta() {
        System.out.println(echo);
    }
}

26 class India extends Hotel implements Foxtrot {
    public void alfa() {
        System.out.println("Sierra");
    }
    public static void bravo() {
31     System.out.println("Tango");
    }
    public void charlie() {
        bravo();
    }
36 }
```


mike.bravo()

```
1 interface Foxtrot {
    String echo = "Uniform";
    void alfa ();
}

6 interface Golf {
    String echo = "Victor";
    void alfa(String juliett );
}

41 class Oscar {
    public static void main(String papa[]) {
        Golf lima = new India();
        Hotel mike = (India)lima;
        lima.alfa("X-Ray");
46     mike.alfa("Yankee");
        mike.bravo();
        mike.charlie ();
        India.delta ();
    }
51 }
```

```
11 class Hotel implements Golf {
    public void alfa(String kilo) {
        System.out.println("Quebec");
    }
16     public static void bravo() {
        System.out.println("Romeo");
    }
    public void charlie() {
        bravo();
    }
21     public static void delta() {
        System.out.println(echo);
    }
}

26 class India extends Hotel implements Foxtrot {
    public void alfa() {
        System.out.println("Sierra");
    }
    public static void bravo() {
31     System.out.println("Tango");
    }
    public void charlie() {
        bravo();
    }
36 }
```

mike.charlie()

```
1 interface Foxtrot {
    String echo = "Uniform";
    void alfa ();
}

6 interface Golf {
    String echo = "Victor";
    void alfa(String juliett );
}

41 class Oscar {
    public static void main(String papa[]) {
        Golf lima = new India();
        Hotel mike = (India)lima;
        lima.alfa("X-Ray");
46 mike.alfa("Yankee");
        mike.bravo();
        mike.charlie ();
        India.delta ();
    }
51 }
```

```
11 class Hotel implements Golf {
    public void alfa(String kilo) {
        System.out.println("Quebec");
    }
    public static void bravo() {
16     System.out.println("Romeo");
    }
    public void charlie() {
        bravo();
    }
21     public static void delta() {
        System.out.println(echo);
    }
}

26 class India extends Hotel implements Foxtrot {
    public void alfa() {
        System.out.println("Sierra");
    }
    public static void bravo() {
31     System.out.println("Tango");
    }
    public void charlie() {
36     bravo();
    }
}
```

India.delta()

```
1 interface Foxtrot {  
    String echo = "Uniform";  
    void alfa ();  
}
```

```
6 interface Golf {  
    String echo = "Victor";  
    void alfa(String juliett );  
}
```

```
41 class Oscar {  
    public static void main(String papa[]) {  
        Golf lima = new India();  
        Hotel mike = (India)lima;  
        lima.alfa("X-Ray");  
46     mike.alfa("Yankee");  
        mike.bravo();  
        mike.charlie ();  
        India.delta ();  
    }  
51 }
```

```
11 class Hotel implements Golf {  
    public void alfa(String kilo) {  
        System.out.println("Quebec");  
    }  
    public static void bravo() {  
16     System.out.println("Romeo");  
    }  
    public void charlie() {  
        bravo();  
    }  
21     public static void delta() {  
        System.out.println(echo);  
    }  
}
```

```
26 class India extends Hotel implements Foxtrot {  
    public void alfa() {  
        System.out.println("Sierra");  
    }  
    public static void bravo() {  
31     System.out.println("Tango");  
    }  
    public void charlie() {  
        bravo();  
    }  
36 }
```

Aufgabe 6

Implementieren Sie eine Methode

`static int[][] matrixAdd(int m1[][], int m2[][])` derart, dass sie die beiden Eingabematrizen `m1` und `m2` addiert und die Ergebnismatrix zurückgibt.

Eine Matrix wird mit einer anderen Matrix addiert, indem die Matrixelemente einzeln addiert werden.

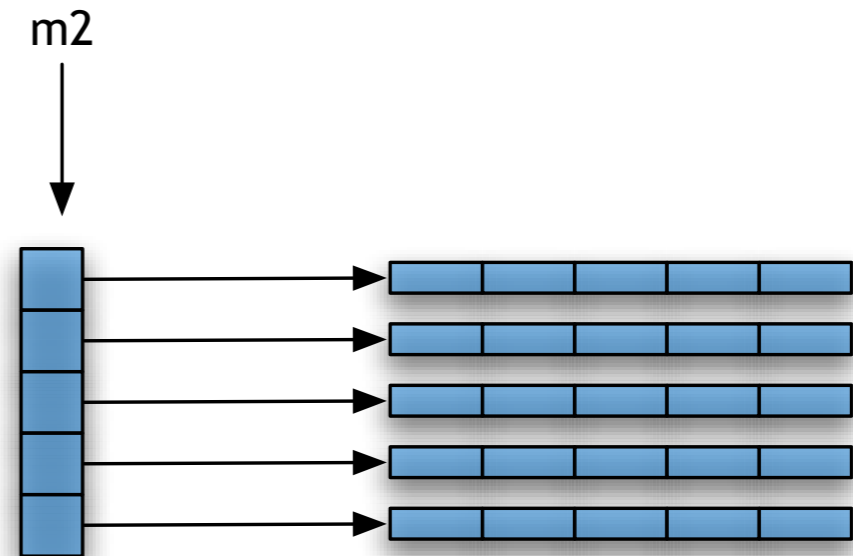
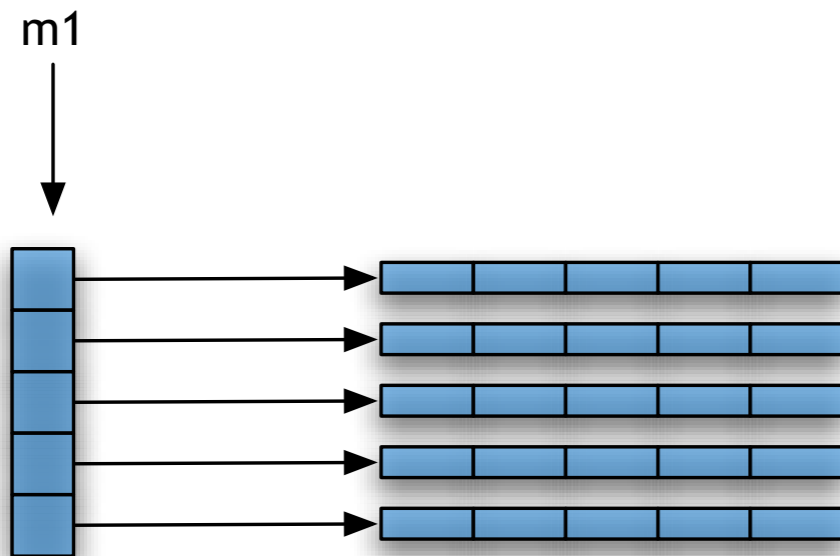
Das heißt: Seien A, B Matrizen mit I Zeilen und J Spalten und α_{ij}, β_{ij} das jeweilige Element in Zeile i und Spalte j . Dann ist $\Gamma = A + B$ gegeben durch $\gamma_{ij} = \alpha_{ij} + \beta_{ij}$ für alle i, j . Eine Matrixaddition ist nur möglich, wenn beide Operanden die selbe Anzahl von Zeilen und Spalten besitzen. Stellen Sie sicher, dass Ihre Implementation bei einer ungültigen Eingabe (zum Beispiel ungültige Matrix oder nicht addierbare Matrizen) statt der Ergebnismatrix eine Nullreferenz zurückgibt.

Gültigkeit

```
if ( (m1 == null) || (m2 == null) )
    return null;

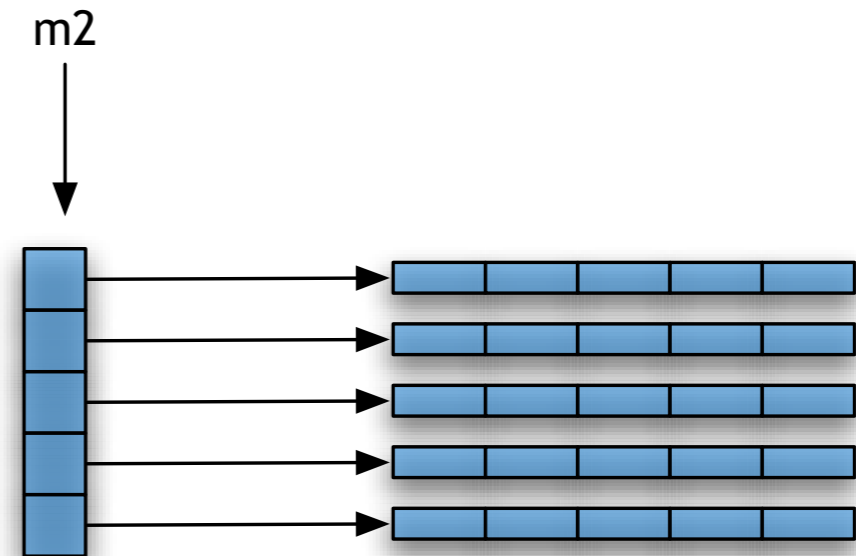
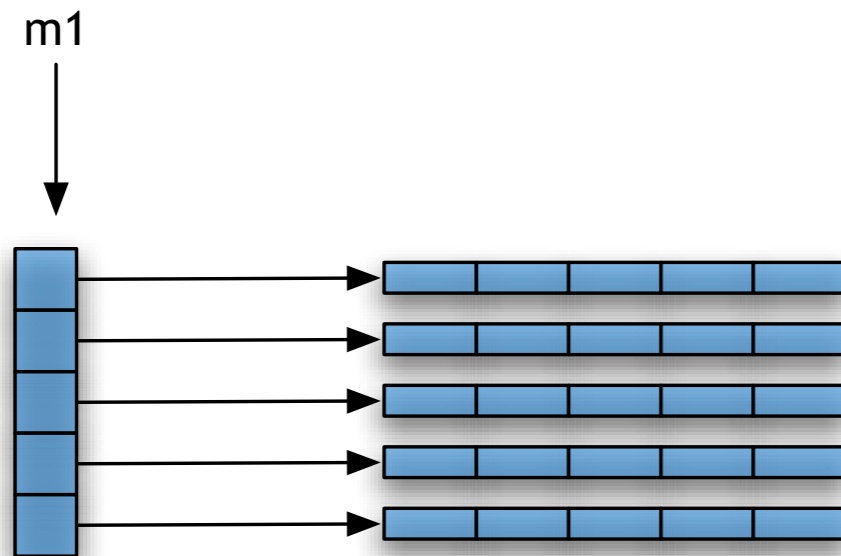
if ( m1.length != m2.length )
    return null;

for (int i = 0; i < m1.length)
    if ( (m1[i] == null) || (m2[i] == null) ) ||
        (m1[i].length != m2[i].length) || (m1[i].length != m1[0].length))
        return null;
```



Addition

```
int m3[m1.length][m1[0].length];  
  
for (int i = 0; i < m1.length)  
    for (int j = 0; j < m1[0].length; j++)  
        m3[i][j] = m1[i][j] + m2[i][j];  
  
return m3;
```



5: Übungsblatt 10

anhand pdfs

**Nicht vergessen:
It ain't over till it's over...**