

Prüfung **(Lösung)**
Informatik I (D-BAUG)

Hermann Lehner, Felix Friedrich

ETH Zürich, 28.01.2019.

Name, Vorname:

Legi-Nummer:

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen konnte, und dass ich die allgemeinen Richtlinien gelesen und verstanden habe.

I confirm with my signature that I was able to take this exam under regular conditions and that I have read and understood the general guidelines.

Unterschrift:

Allgemeine Richtlinien:

General guidelines:

1. Dauer der Prüfung: 60 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für von Ihnen gesprochene Sprachen). 4 A4 Seiten handgeschrieben oder ≥ 11 pt Schriftgrösse.
3. Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.
4. Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben (und direkt darunter, falls mehr Platz benötigt wird). Ungültige Lösungen sind deutlich durchzustreichen! Korrekturen bei Multiple-Choice Aufgaben bitte unmissverständlich anbringen!
5. Es gibt keine Negativpunkte für falsche Antworten.
6. Störungen durch irgendjemanden oder irgendetwas melden Sie bitte sofort der Aufsichtsperson.
7. Wir sammeln die Prüfung zum Schluss ein. Wichtig: Stellen Sie unbedingt selbst sicher, dass Ihre Prüfung von einem Assistenten eingezogen wird. Stecken Sie keine Prüfung ein und lassen Sie Ihre Prüfung nicht einfach am Platz liegen. Dasselbe gilt, wenn Sie früher abgeben wollen: Bitte melden Sie sich lautlos, und wir holen die Prüfung ab. Vorzeitige Abgaben sind nur bis 15 Minuten vor Prüfungsende möglich.
8. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen.
9. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

- Exam duration: 60 minutes.*
- Permitted examination aids: dictionary (for languages spoken by yourself). 4 A4 pages hand written or ≥ 11 pt font size.*
- Use a pen (black or blue), not a pencil. Please write legibly. We will only consider solutions that we can read.*
- Solutions must be written directly onto the exam sheets in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be crossed out clearly. Provide corrections to answers of multiple choice questions without any ambiguity!*
- There are no negative points for wrong answers.*
- If you feel disturbed by anyone or anything, let the supervisor of the exam know immediately.*
- We collect the exams at the end. Important: You must ensure that your exam has been collected by an assistant. Do not take any exam with you and do not leave your exam behind on your desk. The same applies when you want to finish early: Please contact us silently and we will collect the exam. Handing in your exam ahead of time is only possible until 15 minutes before the exam ends.*
- If you need to go to the toilet, raise your hand and wait for a supervisor.*
- We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.*

Question:	1	2	3	4	5	6	Total
Points:	14	12	10	10	10	4	60
Score:							

Aufgabe 1: Java Grundlagen (14P)

Betrachten Sie den folgenden Code.

Consider the following code.

```

1  public static double[] m(double[] a) {
2      if (a.length < 1) return null;
3      double min = 0;
4      int minIndex = 0;
5      for(byte i = 0; i < a.length ; ++i){
6          if (a[i] < min) {
7              min = a[i];
8              minIndex = i;
9          }
10     }
11     int[] res = new int[2];
12     res[0] = min;
13     res[1] = minIndex;
14     return res;
15 }
```

- (a) Aufwärmübung: Markieren Sie im Code die Stellen (Einkreisen!), an denen der Wert von `min` geschrieben wird.
Occurrence of 'min' on line 3 and 7

Warming-up: Mark the places (Circles!) in the code where the value of min is written.

- (b) Der Compiler gibt die folgenden zwei Fehler aus. Ändern Sie eine Zeile des Programms, um diese beiden Fehler zu beseitigen. Schreiben Sie die Zeilennummer und den neuen Code in die untenstehenden Felder.

The following two compiler error are reported. Change one line in the program to fix those two errors. Write the line number and the new code in the fields below.

```
./Main.java:12: error: incompatible types: possible lossy conversion
from double to int
```

```
    res[1] = min;
           ^
```

```
./Main.java:14: error: incompatible types: int[] cannot be converted
to double[]
```

```
    return res;
           ^
```

2 errors

Zeile / Line:

Code / Code:

- (c) (i) An wie vielen Stellen wird der Wert von `minIndex` gelesen?
 (ii) Welchen Wert nimmt `i` während der Ausführung von `m` zuletzt an (solange es existiert)?

*(i) In how many places is the value of `minIndex` read?
 (ii) What is the last value of `i` during an execution of `m` (as long as it exists)?*

/2P

(i) (ii) `a.length-1` `a.length` `a.length+1`

- (d) Das Programm stürzt bei der Ausführung der folgenden zwei Anweisungen ab. Geben Sie den Grund an und korrigieren Sie das Programm auf einer Zeile.

The program crashes when executing the following two statements. State the reason and fix the program on one line.

/2P

```
double[] a = null;
m(a);
```

Grund: / Reason:

`null gets dereferenced on line 2`

Korrektur: / Correction:

Zeile / Line: Code / Code: `if (a == null || ...`

- (e) Der Code soll die kleinste Zahl in einem Array sowie deren Position zurückgeben.
 (i) Wie viele Elemente darf das Array `a` maximal enthalten damit der Code funktioniert wie beschrieben?
 (ii) Es gibt noch andere Einschränkungen. Geben Sie ein Array mit drei Zahlen an, für welchen der Algorithmus eine falsche Antwort liefert.
 (iii) Welche Zeile im Programm würden Sie ändern, um das Problem zu beheben?

*The code is supposed to return the smallest number in an array as well as its position.
 (i) What is the maximum number of elements in the Array `a` such that the code can work as described?
 (ii) There are also other limitations. Provide an array containing three numbers for which the algorithm provides a wrong answer.
 (iii) Which line in the program would you change to fix the problem?*

/5P

(i) (ii) (iii)

- (f) Gegeben ein Array `a` mit den Werten `[1.0, 2.0, 3.0, 4.0]`. Welche Werte enthält `a` nach der Ausführung von `m(a)`?

Let's assume an array `a` with values `[1.0, 2.0, 3.0, 4.0]`. What values does `a` contain after calling `m(a)`?

/2P

`still the same: [1.0, 2.0, 3.0, 4.0]`

Aufgabe 2: Methoden und Rekursion (12P)

/2P (a) Aufgabe aus dem Buch:

Welche Ausgabe erzeugt folgende Methode, wenn man sie mit `p(4)` aufruft?

```
static void p (int x) {  
    if (x > 0) {  
        p(x -1);  
        Out.println(x);  
    }  
}
```

Task from the book:

What output is produced by the following method, if called with `p(4)`?

Ausgabe / Output:

1 2 3 4

/10P (b) Beim sogenannten Rucksackproblem ist eine Menge von Gegenständen gegeben, die jeweils ein bestimmtes Gewicht und einen bestimmten Wert haben. Es soll eine Teilmenge der Gegenstände ausgewählt werden, die einen möglichst hohen Gesamtwert besitzt, ohne aber ein gegebenes Maximalgewicht zu überschreiten.

Das Problem kann rekursiv gelöst werden: Entweder man nimmt den aktuell ersten Gegenstand oder nicht. In beiden Fällen verbleibt das (jeweils unterschiedliche) Teilproblem, aus den weiteren Gegenständen die beste Teilmenge auszuwählen. Der Fall mit dem höheren Gesamtwert "gewinnt".

Wir modellieren Gegenstände als Objekte vom Typ `Item`. Das Feld `weight` gibt das Gewicht des Gegenstandes an, das Feld `value` den Wert des Gegenstandes.

In the so-called rucksack problem, we are given a set of items, each with a certain weight and a certain value. We want to choose a subset of items such that their total value is as large as possible but without exceeding a given maximum weight.

The problem can be solved recursively: Either the currently first item is picked or not. In both cases, it remains a (different) subproblem to pick the best subset from the remaining items. The case with the larger total value "wins".

We model items as objects of type `Item`. The field `weight` represents the weight of the item, the field `value` the value of the item.

```
class Item {  
    int weight;  
    int value;  
}
```

Vervollständigen Sie die Methode `maxValue` auf der gegenüberliegenden Seite, so dass `maxValue(0, w)` den grösstmöglichen Gesamtwert für das gegebene Maximalgewicht `w` berechnet und zurückgibt.

Complement the method `maxValue` on the opposite page such that `maxValue(0, maxWeight)` computes and returns the largest-possible total value for the given maximum weight `w`.

Hinweis: Die Methode `Math.max(a, b)` berechnet das Maximum der beiden Ganzzahlen `a` und `b`.

Tip: the method `Math.max(a,b)` computes the maximum of the two integers `a` and `b`.

```
class Rucksackproblem {
    // the array of Items, filled from somewhere (irrelevant)
    Item[] items;

    // Parameter i: The current index in the array above
    // Parameter w: The remaining available weight
    int maxValue(int i, int maxWeight) {
        // no items left
        if (i >= items.length) {
            return 0;
        }

        int weight = items[i].weight; // weight of current item
        int value = items[i].value; // value of current item

        // current item is heavier than the remaining available weight
        if (weight > maxWeight) {
            return maxValue(i + 1, maxWeight);
        } else {
            // we may or may not want to pick the current item ...

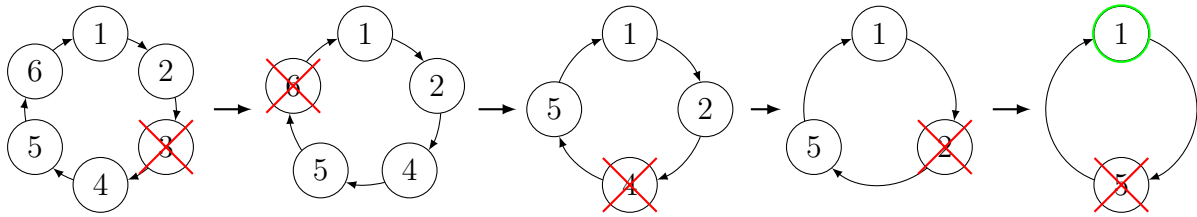
            return Math.max(
                maxValue(i + 1, maxWeight),
                maxValue(i + 1, maxWeight - weight) + value
            );
        }
    }
}
```

/10P

Aufgabe 3: Dynamische Datenstrukturen (10P)

Wir betrachten ein Auszählspiel bei dem sich n Kinder im Kreis aufstellen. Dabei wird jeweils reihum das k te Kind ausgezählt, welches den Kreis verlassen muss. Danach wird von dort weitergezählt. Dies wird solange fortgesetzt, bis nur noch ein Kind übrig ist, welches dann gewonnen hat.

Die folgende Grafik verdeutlicht das Spiel mit $n = 6$ und $k = 3$.



Let's look at a count-out game where n children stand in a circle. In turns each k th child is being counted-out, who must then leave the circle. After this, counting starts from there again. This is repeated until there is only one child left, which wins the game.

The following picture illustrates the game with $n = 6$ and $k = 3$.

Die Aufgabe besteht darin, eine Ring-Liste aufzubauen, welche die Kinder im Kreis repräsentiert, sowie das Spiel mithilfe dieser Datenstruktur zu simulieren, um die Position des Gewinners zu erörtern.

Die einzelnen Knoten in der Ring-Liste sind Objekte der Klasse Child, welche wie folgt definiert ist.

The task is to build a ring-list that represents the children in the circle, as well as simulating the game using this data structure in order to determine the position of the winner.

The individual nodes of the ring-list are objects of class Child that is defined as follows:

```
class Child {
    int number;
    Child next;

    Child(int number){
        this.number = number;
    }
}
```

Vervollständigen Sie die Methode countOut.

Complement the method countOut.

```
class Main {

    // PRE: n >= 2, k >= 1
    // POST: the number of the last remaining child.
    private static int countOut(int n, int k){

        // Part 1: Create a ring of children
        Child first = new Child(1); // create first child
        Child previous = first;
        Child current = null;
        for (int i = 2; i <= n ; ++i){ // create all other children

            current = new Child(i);
            previous.next = current; // point from previous to current
            previous = current; // move previous for next iteration

        }

        current.next = first; // close the ring

        // Part 2: Count-out each k-th child in the ring
        do {

            for (int i = 1; i < k; ++i){ // skip k-1 children
                current = current.next;
            }
            current.next = current.next.next; // remove the k'th child.

        } while (current.next != current); // only one child left.
        return current.number;
    }

    public static void main (String[] args) {
        Out.print("Number of children. n = ");
        int n = In.readInt();
        Out.print("Each k'th child is counted-out. k = ");
        int k = In.readInt();
        int lastNumber = countOut(n, k);
        Out.println("The child at position " + lastNumber + " wins!");
    }
}
```

Aufgabe 4: Klassen, Kapselung und Floating Points (10P)

Vektoren in der euklidischen Ebene sind definiert durch Länge und Richtung, können aber auch durch die kartesischen Koordinaten x und y repräsentiert werden.

- /2P (a) Definieren Sie die Felder x und y vom Typ `double`, welche die kartesischen Koordinaten des Vektors enthalten. Schreiben Sie einen Konstruktor um die beiden Werte zu setzen. Die Felder der Klasse sollen von ausserhalb der Klasse nicht zugreifbar sein.

- /4P (b) Schreiben Sie die beiden Methoden `getLength` und `getAngle` um Länge und Richtung des Vektors zu erhalten. Diese sollen von ausserhalb der Klasse zugreifbar sein. Es soll möglichst wenig Genauigkeit verloren gehen.

- /2P (c) Vervollständigen Sie die Methode `dot`, welche als Argument einen zweiten Vektor erhält und daraus das Skalarprodukt berechnet.

- /2P (d) Vervollständigen Sie die Methode `isOrthogonal`, welche prüft ob zwei Vektoren orthogonal zueinander stehen, indem geprüft wird, ob deren Skalarprodukt Null ist.

Tipps

- Denken Sie immer an die drei Flieskommazahlen Regeln!
- Berechnung der Länge:
$$l = \sqrt{x^2 + y^2}$$
- Berechnung der Richtung:
$$\theta = \text{atan2}(y, x) * 180/\pi$$
- Skalarprodukt
$$v_1 \cdot v_2 = v_1.x * v_2.x + v_1.y * v_2.y$$
- Sie dürfen die Funktionen `Math.sqrt`, `Math.pow`, `Math.abs`, `Math.atan2` sowie die Konstante `Math.PI` der Standardbibliothek verwenden.

Vectors in the euclidean plane are defined by their length and direction, but they can also be represented by the cartesian coordinates x and y .

Define the fields x and y of type `double` that contain the cartesian coordinates of the vector. Write a constructor to set the two values. The fields must not be accessible from outside the class.

Write the two methods `getLength` and `getAngle` to get the length and direction of the vector. Those should be accessible from outside the class. Aim to loose as little precision as possible.

Complement the method `dot` that takes another vector as argument and computes the dot-product of the two vectors.

Complement the method `isOrthogonal` that checks if two vectors are orthogonal. This is done by checking if their dot-product is zero.

Hints

- Always have the three floating-point rules in mind!
- Computing the length:
$$l = \sqrt{x^2 + y^2}$$
- Computing the direction:
$$\theta = \text{atan2}(y, x) * 180/\pi$$
- Dot-product
$$v_1 \cdot v_2 = v_1.x * v_2.x + v_1.y * v_2.y$$
- You may use the functions `Math.sqrt`, `Math.pow`, `Math.abs`, `Math.atan2` as well as the constant `Math.PI` from the standard library.


```
public class Vector2d {
    // Two numbers that differ less than epsilon are considered the same.
    private static final double epsilon = 1e-10;

    // Fields and Constructor (Task a)
    private double x, y;

    public Vector2d(double x, double y){
        this.x = x;
        this.y = y;
    }

    // Methods getAngle and getLength (Task b)
    public double getAngle() {
        return Math.atan2(y, x) * 180 / Math.PI;
    }

    public double getLength() {
        return Math.sqrt(x*x + y*y);
    }

    // Dot-product (Task c)
    public double dot(Vector2d other){
        return this.x * other.x + this.y * other.y;
    }

    // Check if two vectors are orthogonal (Task d)
    public boolean isOrthogonalTo(Vector2d other){
        return Math.abs(this.dot(other)) < this.epsilon;
    }
}
```

Aufgabe 5: Vererbung und Polymorphie (10P)

Gegeben ist folgende Klassenstruktur.

Given the following class structure.

```
public abstract class Fruit {
    private String name;
    private String origin;

    protected Fruit(String name, String origin) {
        this.name = name;
        this.origin = origin;
    }

    public abstract String color();

    public void eat(){
        Out.print( "Eating a " +
            this.color() + " " + this.getName() + " (" + this.origin + ")");
    }

    public String getName() {
        return this.name;
    }
}
```

```
public class TropicalFruit extends Fruit {
    private boolean isEnjoyable;

    public TropicalFruit(String name, boolean isEnjoyable) {
        super(name, "tropical");
        this.isEnjoyable = isEnjoyable;
    }

    @Override
    public void eat() {
        super.eat();
        if (this.isEnjoyable) {
            Out.print(": mmmh");
        } else{
            Out.print(": brrr");
        }
    }

    @Override
    public String color() {
        return "";
    }
}
```

```

public class Banana extends TropicalFruit {
    public Banana() {
        super("banana", true);
    }

    public Banana(String type) {
        super(type + " banana", true);
    }

    @Override
    public String color() {
        return "yellow";
    }
}

```

- (a) Geben Sie für die folgenden Codeabschnitte an ob sie kompilieren. Wenn ja, schreiben Sie hin, was der Code ausgibt. Wenn nein, erklären Sie, warum nicht.

Indicate for the following code fragments if they compile. If yes, write down the output of the code. If no, explain why not.

/10P

```

Fruit fruit = new Banana();
Out.println(fruit.name);

```

Ausgabe / *Output*: **Compiler Error:**

error: name has private access in Fruit

```

Fruit fruit = new Banana();
Banana banana = fruit;
Out.println(banana.color());

```

Ausgabe / *Output*: **Compiler Error:**

error: incompatible types: Fruit cannot be converted to Banana

```

Banana banana = new Banana("mini");
TropicalFruit fruit = banana;
fruit.eat();

```

Ausgabe / Output: **Compiler Error:**

Eating a yellow mini banana (tropical): mmmh

```

TropicalFruit tropicalFruit =
    new TropicalFruit("mango", false);
Fruit fruit = tropicalFruit;
fruit.eat();

```

Ausgabe / Output: **Compiler Error:**

Eating a mango (tropical): brrr

```

Banana banana = new Banana();
Out.println(banana.getName());

```

Ausgabe / Output: **Compiler Error:**

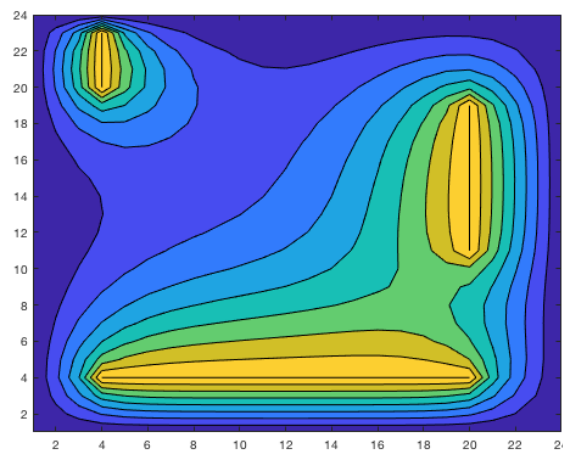
banana

Aufgabe 6: Matlab (4P)

Folgender Matlab Code modelliert die Wärmeausbreitung in der Ebene welche Sie vom Matlab Tutorial kennen.

The following Matlab code models the heat expansion in the plane, which you have seen in the Matlab tutorial.

```
1 Platte (1:24, 1:24) = 20;
2 [hoehe, breite] = size(Platte);
3 start = 1;
4 PlatteAlt = Platte;
5 PlatteNeu = PlatteAlt;
6 for z = 1:50
7     for i = start:hoehe-1
8         for j = start:breite-1
9             PlatteNeu(i, j) = 0.2 * PlatteAlt(i, j) + ...
10                0.2 * (PlatteAlt(i, j-1) + ...
11                PlatteAlt(i, j+1) + ...
12                PlatteAlt(i-1, j) + ...
13                PlatteAlt(i+1, j));
14         end
15     end
16     PlatteNeu (4, 4:20) = 100;
17     PlatteNeu (11:19, 20) = 100;
18     PlatteAlt = PlatteNeu;
19 end
20 contourf(PlatteNeu);
```



- (a) Wenn das Programm ausgeführt wird, zeigt Matlab die folgende Fehlermeldung an. Bringen Sie den Code oben in Ordnung (direkt im Code an einer Stelle).

When we execute the program Matlab displays the following error occurs. Fix the code above appropriately (directly in the code at one place).

/1P

Subscript indices must either be real positive integers or logicals.

Error in contourf (line 10)

```
0.2 * (PlatteAlt(i, j-1) + ...
```

initialize start with 2 (the border must not be re-computed anyway!)

- (b) Da der Code nun funktioniert: Der Code produziert einen schönen Contour Plot. Wird die Ausgabe animiert? Falls ja, wie viele Animationsschritte werden dargestellt?

As the code is working now: The code produces a nice contour plot. Is the plot animated? If yes, how many animation steps are shown.

/1P

Animiert: / *Animated:* Ja / *Yes* Nein / *No*

Anzahl Animationsschritte / *Number of animation steps:*

- (c) Schauen Sie sich den Plot auf der gegenüberliegenden Seite an. Diese kann nicht von dem gezeigten Code stammen. Erklären Sie warum.

Check out the plot on the opposite side. This can't be an output of the shown code. Explain why.

/2P

There are clearly three heat sources in the plot, the code only mentions two.