

Prüfung
Informatik I (D-BAUG)

Hermann Lehner, Felix Friedrich

ETH Zürich, 13.08.2018.

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 Code auf der rechten Seite. Wir setzen die Existenz einer Methode `double log2(double x)` voraus. `log2` gibt den Logarithmus zur Basis 2 zurück. Beispiele `log2(2)` gibt 1 zurück, `log2(7)` gibt 2.807... zurück und `log2(8)` ergibt 3.

Consider the code on the right hand side. Assume the existence of a method `double log2(double x)` returning the base-2 logarithm of x . E.g. `log2(2)` is 1, `log2(7)` is 2.807... and `log2(8)` is 3.

- (a) Der Compiler wirft einen Fehler aus für die Zeile, welche mit `DOES NOT COMPILE` markiert ist. Geben Sie die korrigierte Zeile in folgender Box an.

A compiler error is reported for the line marked with `DOES NOT COMPILE`. Provide the fix in the following box.

- (b) Markieren Sie im Code die Stellen (Einkreisen!), an denen sich `i` ändert.

Mark the places (Circles!) in the code where the value of `i` changes.

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

(i) In how many places is the value of `size` read?

(ii) What is the last value of `i` during an execution of `m`?

- (i) (ii) -1 0 1

- (d) Was passiert, wenn `m(0)` aufgerufen wird?

What happens when calling `m(0)`?

- Es wird `invalid input` ausgegeben und der Aufruf terminiert mit einem leeren Array (der Länge 0) als Resultat. */invalid input is output and the call terminates with an empty array (of length 0) as the result*
- Es wird `invalid input` ausgegeben und der Aufruf terminiert mit `null` als Resultat. */invalid input is output and the call terminates with null as the result.*
- Der Aufruf terminiert mit einer Exception, welche `invalid input` ausgibt. *The call crashes with an exception saying invalid input*

```

1
2 public static int[] m(int value) {
3     assert 0 < value : "invalid input";
4
5     int size = log2(value) + 1; // DOES NOT COMPILE
6     int[] repr = new int[size];
7     int n = value;
8
9     for (int i = size - 1; i > 0; --i) {
10        repr[i] = n % 2;
11        n = n / 2;
12    }
13
14    ;
15
16    return repr;
17 }
```

- (e) (i) Was ist der grösste Wert, welchen size während irgendeiner Ausführung von m annehmen kann?
 (ii) Was ist der "kleinste" numerische Datentyp, welcher genügen würde, um die Variable size zu speichern.
 (iii) Der Datentyp der Variable repr ist int[] – welcher kleinstmögliche Datentyp würde hier genügen?

*(i) What is the largest value variable size can take on any execution of m?
 (ii) What is the "smallest" numerical data type that would suffice for variable size?
 (iii) The type of variable repr is int[] - but which smaller data type would suffice here?*

/3P

(i) (ii) (iii)

- (f) Geben Sie die Rückgabewerte von m(7) and m(8) an (ignorieren Sie den fehlen Code in Zeile 14).

Provide the return values of m(7) and m(8) (ignore the missing code in line 14).

/4P

m(7) m(8)

- (g) m(n) sollte die Binärzahldarstellung der Zahl n (n > 0) zurückgeben, z.B. {1,1} für m(3), das funktioniert aber nicht. Bringen Sie das in Ordnung, indem Sie Zeile 14 ausfüllen.

m(n) is expected to return the binary representation of n (n > 0), e.g. {1,1} for m(3), but it doesn't just yet. Fix it by filling in line 14.

/2P

Aufgabe 2: Methoden und Rekursion (12P)

Wie versprochen: Die folgenden zwei Aufgabenteile sind direkt aus dem Buch kopiert!

As promised: The following two task parts are directly copied from the book!

- /2P (a) Welche Ausgabe erzeugt folgende Methode, wenn man sie mit `p(5)` aufruft?

What output is produced by the following method, if called with `p(5)`?

```
static int p(int x) {
    if (x == 0) return 1;
    else {
        int y = p(x-1);
        Out.println(x+y);
        return y + 1;
    }
}
```

Ausgabe / *Output:*

- /2P (b) Welche der folgenden Aussagen sind richtig? Eine direkt rekursive Methode. . .

Which of the following statements are correct? A directly recursive method. . .

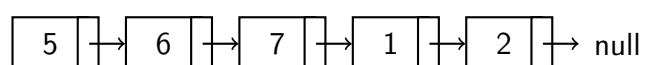
- . . . darf nicht auf globale Variablen zugreifen. / *. . . must not access global variables*
- . . . legt bei jedem Aufruf einen neuen Satz ihrer lokalen Variablen an. / *. . . creates a new set of its local variables upon each invocation.*
- . . . muss mit `return` beendet werden. / *. . . must be terminated using return.*
- . . . ruft sich selbst auf. / *. . . calls itself.*

- /6P (c) Unten sehen Sie eine **einfach verkettete Liste** mit fünf Elementen. Jedes Element der Liste ist eine Instanz der Klasse `Node`, welche auch unten angegeben ist und welche eine Ganzzahl enthält und eine Referenz `next` auf den nächsten `Node`.

*Below a **singly linked list** with 5 elements is shown. Each element of the list is an instance of class `Node` which is given below and contains an integer value as well as a field referring to the next `Node`.*

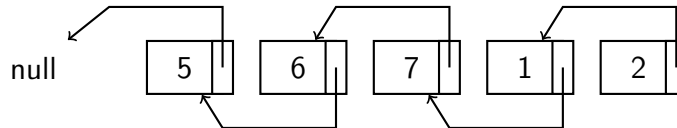
```
class Node {
    int value;
    Node next;

    Node(int value) {
        this.value = value;
    }
}
```



Vervollständigen Sie die folgende rekursive Methode, so dass sie die Liste wie in der folgenden Abbildung gezeigt umdreht, wenn der erste Knoten (mit Wert 5) an die Methode übergeben wird.

Complement the following recursive method such that it reverses the list as shown in the picture below when the first node (with value 5) is passed to the method.



```

/**
 * Reverse a non-empty list.
 * @param node the head of the list to reverse, must not be null
 * @return the new head of the list (i.e. the last element of the list
 *         that is being reversed)
 */
Node reverse(Node node) {
    if (node.next == null) { // base case: list of 1 element
        [ ]
    } else {
        // recursively reverse the rest of the list:
        Node newHead = reverse(node.next);
        // now append node to reversed list
        [ ]
        return newHead;
    }
}

```

(d) Geben Sie einen kurzen informellen Beweis, dass Ihre Methode terminieren wird, vorausgesetzt die Liste ist endlich.

Provide a short informal proof that your method will terminate, if provided with a finite list.

/2P

/10P

Aufgabe 3: Dynamische Datenstrukturen (10P)

Eine Map (Wörterbuch) enthält Zuordnungen von Schlüsseln zu Werten. Die Anordnung der Elemente in der Map ist unerheblich.

In der folgenden Implementation ist eine Map gespeichert als verkettete Liste von Schlüsseln, bei denen jedes Element zusätzlich den zugehörigen Wert speichert (oder `null`, wenn kein Wert). Im folgenden ist die Datenstruktur angegeben, welche ein Einzelement der Map speichert (ein $\langle \text{key}: \text{value} \rangle$ Paar).

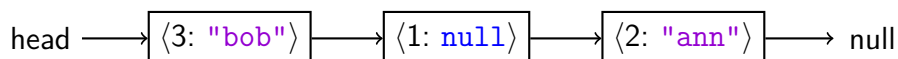
```
class ElementNode {
    ElementNode next;
    int key;
    String value;

    ElementNode (int k, String v, ElementNode n){
        key = k;
        value = v;
        next = n;
    }
}
```

Nachfolgend ist der Inhalt der Map $\{\langle 3: \text{"bob"} \rangle, \langle 2: \text{"ann"} \rangle\}$ gezeigt.

Diese wurde wie folgt generiert: in einer initial leeren Map wurden die Werte $\langle 2: \text{"ann"} \rangle$, dann $\langle 1: \text{"eve"} \rangle$ und dann $\langle 3: \text{"bill"} \rangle$ gesetzt, und dann wurde der Wert zum Schlüssel 1 gelöscht: ($\langle 1: \text{null} \rangle$), und schliesslich wurde das Paar $\langle 3: \text{"bob"} \rangle$ gesetzt (welches den Wert von 3 überschreibt).

Um einen Schlüssel von der Map zu löschen, muss dieser nicht von der Map gelöscht werden, sondern es wird der Wert zu `null` gesetzt.



Vervollständigen Sie den Code so dass die Datenstruktur `ListMap` wie erwartet und dokumentiert funktioniert.

A map (dictionary) holds mappings from keys to values. The order of the elements in the map is irrelevant.

In the following implementation a map is stored as a linked list of keys where each element additionally contains the corresponding value (or `null`, if there is no value). Listed in the following is the data structure corresponding to a single element of the map (a $\langle \text{key}: \text{value} \rangle$ pair).

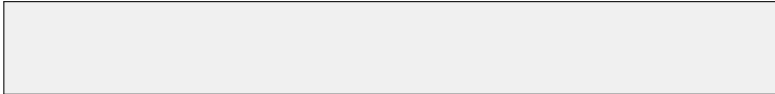
Presented in the following is the content of the map $\{\langle 3: \text{"bob"} \rangle, \langle 2: \text{"ann"} \rangle\}$.


It has been generated as follows: to an initially empty map, the pairs $\langle 2: \text{"ann"} \rangle$, then $\langle 1: \text{"eve"} \rangle$, and then $\langle 3: \text{"bill"} \rangle$ were set, then the value for key 1 ($\langle 1: \text{null} \rangle$) was removed, and finally the pair $\langle 3: \text{"bob"} \rangle$ was set (which overwrites the value for 3).


To remove a key from the map, it is not deleted from the list, but only its value is set to `null`.

Complement the code such that the data structure `ListMap` would work as expected and documented.

```
class ListMap {
    ElementNode head = null;

    /** Insert the <key: value> pair into the data structure.
     * If the corresponding element exists then replace its value,
     * otherwise add a new element at the front of the list. */
    public void set(int key, String value) {
        ElementNode e = head;
        while (e != null && e.key != key) {
            e = e.next;
        }
        if (e == null) { // not found
            
        } else {
            e.value = value;
        }
    }

    /** Remove the <key: value> pair corresponding to the provided key from
     * the data structure. If the element with this key is already in the list,
     * then set its value to null.
     * Elements are not physically removed from the list. */
    public void remove(int key) {
        ElementNode e = head;
        while (e != null && e.key != key) {
            e = e.next;
        }
        
    }

    /** Output all elements with a value (value != null). */
    public void out() {
        for () {
            if (n.value != null) {
                Out.println("<" + n.key + ": " + n.value + ">");
            }
        }
    }
}
```

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

- /4P (a) Teilchen bewegen sich in einer zweidimensionalen Box: ihre Position ist gegeben durch x und y Koordinaten. Beide Koordinaten haben typ `float`. Implementieren Sie die Klasse `particle`, die ein Teilchen repräsentiert. Vervollständigen Sie das vorgegebene Codeskelett der Klasse `particle` in den vorgegebenen Boxen. Kapseln Sie Daten und schreiben Sie einen passenden Konstruktor.

Simple 2d particles move inside a box: their position is represented with x and y coordinates. Both coordinates have type `float`. Implement a class `particle` that represent a particle. Complete the given skeleton for the class `particle` using the designated gray boxes. Encapsulate the data and write a suitable constructor.

- /2P (b) Schreiben Sie eine Methode `move`, welche das Teilchen bewegt und die x und y Koordinaten entsprechend anpasst.

Write a method `move` that moves the particle and updates the coordinates x and y accordingly.

```
public class Particle {  
    // Fields  
    private static final float epsilon = 0.00001f;
```

```
// Constructor accepting coordinate (x, y)
```

```
/** Moves the particle by updates its position with a translation  
 * vector (dx, dy) */  
public void move(float dx, float dy) {
```

```
}
```

```
// Other operations omitted  
[...]
```


- (c) Schreiben Sie eine Methode, welche den euklidischen Abstand zwischen dem Teilchen `this` und dem Teilchen `other` berechnet und eine welche nachprüft, ob ein Teilchen mit einem anderen Teilchen kollidiert.

Tipps

- Teilchen kollidieren, wenn ihre Koordinaten übereinstimmen.
- Euklidische Distanz:
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
- Sie dürfen die Funktionen `Math.sqrt`, `Math.pow`, `Math.abs` verwenden

Write a method that that calculates the euclidean distance between `this` and `other` particle and one that checks if a particle collides with another particle.

Hints

- *Particles collide when their coordinates coincide.*
- *Euclidean distance:*
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
- *You may use the functions `Math.sqrt`, `Math.pow`, `Math.abs` from the standard library.*

```
[...] // see previous page for fields of class Particle
```

```
/** @return the euclidean distance from another particle 'other' */  
public double distance(Particle other) {
```

```
}
```

```
/** Checks if 'other' and 'this' collides based on their positions  
 * @return true if considered colliding, false otherwise. */  
public boolean hitOther(Particle other) {
```

```
}
```

```
}
```

Aufgabe 5: Vererbung und Polymorphie (10P)

Gegeben ist folgende Klassenstruktur.

Given the following class structure.

```
public abstract class Creature {
    private String name;
    private int health;

    protected Creature(String name, int health) {
        this.name = name; this.health = health;
    }

    public void hasHealth() {
        if (health <= 0) Out.println(name + " is dead!");
        else Out.println(name + " has health " + health);
    }

    public String attackType() {
        return "bites!";
    }

    public abstract String move();

    public void attackPattern() {
        Out.println(name + " " + move() + " and " + attackType());
    }
}
```

```
public class FlyingCreature extends Creature {
    protected FlyingCreature(String name, int health){
        super(name, health);
    }

    public FlyingCreature(String name){
        super(name, 10);
    }

    public String move(){
        return "flies";
    }
}
```

```
public class Dragon extends FlyingCreature {
    public Dragon(String name, int health) {
        super(name, health);
    }

    @Override
    public String attackType(){
        return "spits fire!";
    }
}
```

- (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

```
FlyingCreature d =
    new FlyingCreature(
        "Alex the White Bird");
Creature c = d;
c.attackPattern();
```

Ausgabe / *Output*: Compiler Error:

```
FlyingCreature d =
    new Dragon(
        "Barney the Purple Dragon", 5);
Creature c = d;
Out.println(c.move());
```

Ausgabe / *Output*: Compiler Error:

```
FlyingCreature d =
    new Dragon(
        "Fred the Red Dragon");
d.hasHealth();
```

Ausgabe / *Output*: Compiler Error:

```
Creature c =
    new Dragon(
        "Edward the Blue Dragon", 10);
c.attackPattern();
```

Ausgabe / *Output*: Compiler Error:

```
FlyingCreature c =
    new Dragon(
        "Bella the pink Dragon", 5);
Dragon d = c;
c.hasHealth();
```

Ausgabe / *Output*: Compiler Error:

Aufgabe 6: Matlab (4P)

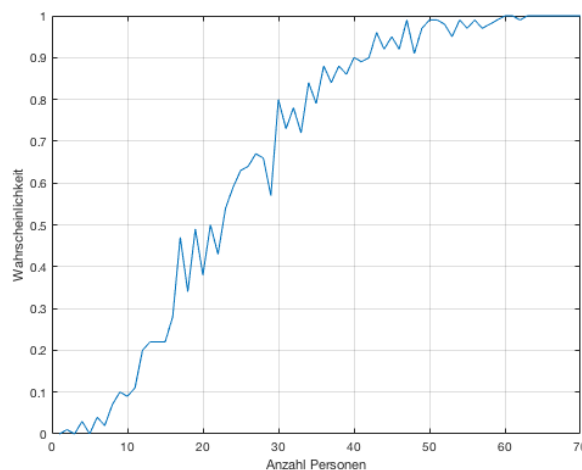
Der Matlab Code unten zeigt die Monte-Carlo Simulation des Geburtstagsparadoxons welches Sie von der Vorlesung kennen.

Ausserdem eine mögliche Ausgabe der Variable P_sim.

The Matlab code snippet shown below shows the Monte-Carlo simulation of the birthday paradoxon which you have seen in the lecture.

Also, a possible plot of variable P_sim.

```
1 daysInYear = 365;
2 maxPeople = 70;
3 experiments = 100;
4 P_sim = zeros(1,maxPeople);
5 for n = 0:maxPeople
6     collisions = 0;
7     for i = 1:experiments
8         BirthdayCount = zeros(1, daysInYear);
9         Birthdays = randi([1, daysInYear],1,n); % 1 x n Matrix
10        for birthday = Birthdays
11            BirthdayCount(birthday) = BirthdayCount(birthday) + 1;
12        end
13        maximum = max(BirthdayCount);
14        if maximum >= 2
15            collisions = collisions + 1;
16        end
17    end
18    P_sim(n) = collisions * 0.01;
19 end
```



- (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 birthday_paradoxon (line 18)

- (b) Da der Code nun funktioniert: Wie gross ist die Matrix BirthdayCount wenn die Programmausführung bei Zeile 13 steht?

As the code is working now: What is the size of the matrix BirthdayCount when the execution is at line 13?

/1P

Dimension: ×

- (c) Der Code liefert nur bei genau 100 Experimenten korrekte Prozentzahlen. Ersetzen Sie eine Zeile im Code so, dass auch für andere positive Anzahlen von Experimenten das korrekte Resultat berechnet wird. Schreiben Sie die Zeilennummer und den neuen Code in die untenstehenden Felder.

The provided code provides correct percentages only if we run 100 experiments. Replace one line in the code so that its output is correct also for other number of experiments. Write the line number and the new code in the fields below.

/2P

Zeile / *Line*: Code / *Code*: