Informatik II D-BAUG Prüfung Lösung 8.8.2016 *F. Friedrich*

Name, Vorname: .....

Legi-Nummer: .....

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ausführen 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:

a) Dauer der Prüfung: 120 Minuten.

Exam duration: 120 minutes.

b) Erlaubte Hilfsmittel: Wörterbuch (für gesprochene Sprachen).

Permitted examination aids: dictionary (for spoken languages).

c) Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.

Use a pen (black or blue), not a pencil. Please write legibly. We will only correct solutions that we can read.

d) Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben. Ungültige Lösungen deutlich durchstreichen. Korrekturen bei Multiple-Choice Aufgaben unmissverständlich anbringen.

All solutions must be written directly onto the exercise sheets in the provided boxes. Invalid solutions need to be crossed out clearly. Corrections to answers of multiple choice questions must be provided without any ambiguity.

e) Falls Sie sich durch irgendjemanden oder irgendetwas gestört fühlen, melden Sie dies sofort der Aufsichtsperson.

If you feel disturbed by anyone or anything, immediately let the supervisor of the exam know this.

f) 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.

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 preliminarily is only possible until 15 minutes before the exam ends.

g) Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen. Es darf zur gleichen Zeit immer nur eine Studentin oder ein Student zur Toilette.

If you need to go to the toilet, raise your hand and wait for a supervisor. Only one student can go to the toilet at a time.

h) Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.

Aufgabe	1	2	3	4	5	6	7	8	$\sum$
Punkte									
Maximum	10	10	10	10	10	10	10	10	80

1 Java *(10 Punkte)*

4 P

(a) Geben Sie die Ausgabe der `main`-Methode der folgenden Klasse `Main` direkt in den vorgesehenen Boxen im Code an.

Provide the output of the `main`-method of the class `Main` directly in the provided boxes.

```
public class Main {
  static int R(int a, int b) {
    if (a>0)
      return R(b%a, a);
    else
      return b;
  }

  static void I(int a[]){
    for (int i = 0; i < a.length; ++i)
      a[i] = a[i] + 1;
  }

  static int B(String a){
    int b=1; int s=0;
    for (int i = 0; i < a.length(); ++i){
      if (a.charAt(i)=='1') s += b;
      b*=2;
    }
    return s;
  }

  public static void main(String[] args){
       int X = 10;
       int Y = 20;
       System.out.println(X + Y + "=?=" + X + Y);
       // output:
       System.out.println(R(25,10));
       // output:
       int[] A = {1,2,3};
       I(A);
       for (int x: A) System.out.print(x + " ");
       // output:
       System.out.println();
       System.out.println(B("1001"));
       // output:
  }
}
```

(b) Ergänzen Sie den Code der folgenden Funktion `prod` so, dass das Produkt aller Werte im generalisierten Intervall $[a,b] := \{i \in \mathbb{Z} : a \leq i \leq b \vee b \leq i \leq a\}$ zurückgegeben wird. Tipp: Verwenden Sie Rekursion.

Complement the code of the following function `prod` such that the product of all values in the generalized interval $[a,b] := \{i \in \mathbb{Z} : a \leq i \leq b \vee b \leq i \leq a\}$ is returned. Hint: Use recursion.

4 P

```
//pre : two integers a and b
//post : return the product of all integers in the generalized interval
// [a ,b] = {integer x: a <= x <= b or b <= x <= a}
public static int prod( int a , int b){
  if (a==b) return [ a ];
  else if (a<b) return [ b * prod(a,b-1)
                         oder a * prod(a+1,b) ];
  else return [ a * prod(a-1,b)
                oder b * prod(a,b+1) ];
}
```

(c) Betrachten Sie folgende Klassen- und Variablendeklarationen.

Consider the following class and variable declarations.

2 P

```
class Person{
  public String name;
  public int age;
  public Person (String n, int a) {
    name = n; age = a;
  }
}
...

Person p1 = new Person("Mickey Mouse", 18);
Person p2 = new Person("Mickey Mouse", 18);
Person p3 = new Person("Donald Duck", 18);
Person p4 = p3;
```

Geben Sie an, ob folgende Ausdrücke wahr (`true`) oder falsch (`false`) sind.

Specify if the following expressions are `true` or `false`.

`p1 == p2`	`false`
`p2 == p3`	`false`
`p3 == p4`	`true`

Ende Aufgabe 1

2 Wettersimulation *(10 Punkte)*

Betrachten Sie folgenden Code, welcher für eine sehr einfache Simulation des Wetters gedacht ist. Eine kurze Beschreibung finden Sie auf der rechten Seite.

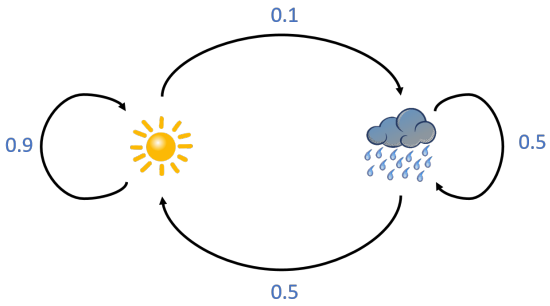
Consider the following code intended for a very simple simulation of the weather. A short description can be found on the right hand side.

```
class Weather{
  final static String[] Weather = {"sunny", "rainy"};

  // return a random integer drawn from probabilities prob[]
  // pre: prob != null, prob.length = 2
  static int Sample(double prob[]){
    double ran = Math.random(); // returns a random value in [0,1)
    [                                    ]
    [ if (ran < prob[0]) return 0;       ]
    [ else return 1;                     ]
    [                                    ]
  }

  // return the next weather randomly from the current weather
  // and the transition matrix m
  static int NextDaysWeather(int currentWeather, double[][] m) {
    return Sample( [ m[currentWeather]   ] );
  }

  public static void main(String[] args) {
    int m = 1000; // number days
    double[][] P = [ {{0.9, 0.1} , {0.5, 0.5}} ];
    int weather = 0; // start with a sunny day
    int counts[] = new int[2];
    for (int i = 0; i<m; ++i){
      weather = NextDaysWeather(weather, P));
      [ counts[weather]++ ];
    }
    for (int i=0; i<2; ++i){
      System.out.println( [ counts[i] ] + " days were " + Weather[i]);
    }
  }
}
```



Die Wettersimulation macht folgende Annahmen: es gibt nur sonniges und regnerisches Wetter. Wenn an einem Tag sonniges Wetter herrscht, dann ist am nächsten Tag das Wetter mit einer Wahrscheinlichkeit von 90% auch sonnig. Wenn regnerisches Wetter herrscht, so bleibt es mit einer Wahrscheinlichkeit von 50% regnerisch. Eine mögliche Ausgabe des Programmes wäre wie folgt:

The weather simulation makes the following assumptions: there are only sunny and rainy days. Given that a day is sunny, the probability that the next day is also sunny is 90%. If a day is rainy, the probability that the next day is also rainy is 50%. A possible output of the program could be as follows:

```
831 days were sunny
169 days were rainy
```

(a) Vervollständigen Sie die Methode `Weather.Sample` so, dass sie eine Zufallszahl zurückgibt, welche gemäss dem Wahrscheinlichkeitsvektor `prob` gezogen wird.

Complement method `Weather.Sample` such that it returns a random number drawn from the probabilities provided by the probability vector `prob`.

4 P

(b) Vervollständigen Sie die Methode `Weather.NextDaysWeather` so, dass sie das simulierte Wetter des nächsten Tages zurückgibt entsprechend oben erklärten Regeln.

Complement the method `Weather.NextDaysWeather` such that it returns the simulated weather of the next day according to the rules explained above.

2 P

(c) Vervollständigen Sie die Methode `Weather.main` so, dass das Program 1000 Tage gemäss dem oben beschriebenen Modell simuliert und zum Schluss die Anzahl sonniger und regnerischer Tage ausgibt.

Complement the method `Weather.main` such that the program simulates 1000 days according to the model described above and such that at the end the number of sonny and rainy days are output.

4 P

Ende Aufgabe 2

3 Hashtabellen *(10 Punkte)*

Ziel dieser Aufgabe ist das Speichern von geographischer Information (wie z.B. Längengrad und Breitengrad) von Städten. Angenommen, die geographische Information sei in einem Datentyp (einer Klasse) `Entry` repräsentiert. Für den schnellen Zugriff auf die Information zu jeder Stadt soll nun folgende Hashtabelle `Data` verwendet werden. Betrachten Sie folgenden Code und beantworten Sie die nachfolgenden Fragen.

The objective of this task is to store geographic information (such as latitude and longitude) of cities. Let the geographic information be represented in some data type (a class) `Entry`. Now, for the efficient access to the information of a city the following hash table `Data` shall be used. Consider the following code and answer the questions below.

```
1  public class Data{
2    String[] key;  // Keys
3    Entry[] data;  // Data
4    int size;
5
6    public Data() {
7      data = new Entry[8]; key = new String[8];
8      size = 0;
9    }
10   // reasonable hash value mapping a name to integer
11   int Hash(String name) {...}; // implementation omitted for brevity
12   // Find data set associated to k using linear probing.
13   // Return index of data set.
14   int Probe(String k){
15     int index = Hash(k);
16     while(key[index] != null && !k.equals(key[index]))
17       index = (index + 1) % key.length;
18     return index;
19   }
20   // Return data set associated to k using linear probing.
21   // Return null if no data is associated to k.
22   public Entry Get(String k){
23     return data[Probe(k)];
24   }
25   // Grow the hash table by a factor of 2
26   void Grow() {
27     Entry[] oldData = data;
28     String[] oldKey = key;
29     data = new Entry[data.length*2];
30     key = new String[key.length*2];
31     size = 0;
32     for (int i = 0; i<oldData.length; ++i)
33       if (oldKey[i] != null)
34         Set(oldKey[i], oldData[i]);
```

```
35    }
36    // Set data at index of k to f. Replace data if present.
37    public void Set(String k, Entry f){
38      int index = Probe(k);
39      data[index] = f;
40      if ( key[index] == null ){
41        key[index]= k;
42        size++;
43        if (size == key.length)
44          Grow();
45      }
46    }
47  }
```

(a) Wir gehen davon aus, dass die Funktion `Data.Hash` "genügend gut" implementiert ist. Was bedeutet das: welche Eigenschaften muss sie im obigen Kontext haben?

We assume that the function `Data.Hash` is implemented "reasonable well". What does this mean: what are the requirements for this function in the context of the code above?

2 P

$0 \leq H(s) < key.length$, Möglichst $s \neq t \Rightarrow H(s) \neq H(t)$

(b) Vervollständigen Sie die Implementation von `Data.Probe` und `Data.Get`. Es soll lineares Sondieren implementiert werden.

Complement the code of `Data.Probe` and `Data.Get`. Linear probing shall be implemented.

3 P

(c) Vervollständigen Sie die Implementation von `Data.Set` und `Data.Grow`.

Complement the code of `Data.Set` and `Data.Grow`.

2 P

(d) Bei Laufzeitmessungen mit einem grossen Datensatz stellen Sie fest, dass ziemlich viele Kollisionen mit obigem Code entstehen, obwohl die Hashfunktion gut implementiert ist. Woran könnte das liegen und welche Codezeile(n) (Zeilennummer angeben) würden Sie ändern, um das Problem zu beheben?

Using runtime measurements you identify that a lot of collisions occur although the hash function is reasonably well implemented. What could be the reason and which line(s) of code (provide the line number) would you change in order to address this problem?

3 P

Zu hoher Füllgrad. Füllgrad begrenzen in Zeilen 42+43

Ende Aufgabe 3

4 Objektorientierte Programmierung *(10 Punkte)*

In dieser Aufgabe geht es um den objektorientierten Ansatz zum Erstellen generischer Frameworks.
Es soll ein Framework zur Generierung von Zufallszahlen mit gewisser Verteilungsannahme erstellt werden. Das Framework enthält dafür: 1.) Pseudozufallszahlen-Generatoren, welche nur uniform verteilte Zufallszahlen im Intervall $[0, 1)$ erzeugen können. 2.) Verteilungsgeneratoren, welche aus uniformverteilten Zufallszahlen solche mit einer gewissen Verteilung erzeugen können.
Exemplarisches Ziel ist, dass folgender Code funktioniert.

This task is about the object oriented paradigm to design generic frameworks.
A framework for the generation of random numbers drawn from a certain distribution shall be designed. For this, the framework contains: 1.) Pseudo-Random Number Generators that can generate random values uniformly distributed in the interval $[0, 1)$ and 2.) Distribution generators that can transform uniformly distributed random numbers into random numbers with a certain distribution.
Exemplary goal is that the following code works.

```
public class Main{

  // output sample mean and sample variance of random numbers
  // sampled from distribution d.
  public static void Analyze(Distribution d) {
    double mean = 0;
    double ssq = 0;
    int n;
    int m = 1000;
    // provisional means algorithm
    for (n=1; n<m; ++n){
      double value = d.Sample();
      double oldMean = mean;
      mean = oldMean + (value - oldMean) / n;
      ssq = ssq + (value - oldMean) * (value - mean);
    }
    // output of mean and empirical variance
    System.out.println(mean + " +/- " + ssq/(m-1));
  }

  public static void main(String[] args){
    // random number generator
    Generator g = new Builtin();
    // uniform distribution using g for sampling
    Distribution u = new Uniform(g);
    // distribution of a dice
    Distribution d = new Dice(g);
    Analyze(u);
    Analyze(d);
  }
}
```

(a) Vervollständigen Sie folgenden Code so, dass der Code auf der linken Seite compiliert und richtig funktioniert.

Complement the following code such that the code on the left hand side compiles and works correctly.

10 P

```
abstract class Generator {
  abstract double Generate();
}

abstract class Distribution {
  abstract double Sample();
}

class Builtin extends Generator
{
  double Generate(){
    return Math.random();
  }
}

class Uniform extends Distribution
{
  Uniform(Generator g){
    rng = g;
  }

  double Sample(){
    return rng.Generate();
  }
}

class Dice extends Distribution
{
  Dice(Generator g){
    rng = g;
  }

  double Sample(){
    return (int)(1+6*rng.Generate());
  }
}
```

Ende Aufgabe 4

5 Ringbuffer *(10 Punkte)*

Die Datenstruktur der Warteschlange (FIFO) wird üblicherweise mit einer verketteten Liste implementiert. Das hat unter anderem den Vorteil, dass darin beliebig viele Elemente aufgenommen werden können. Für jeden aufgenommenen Wert muss jedoch ein Listenelement alloziert werden.

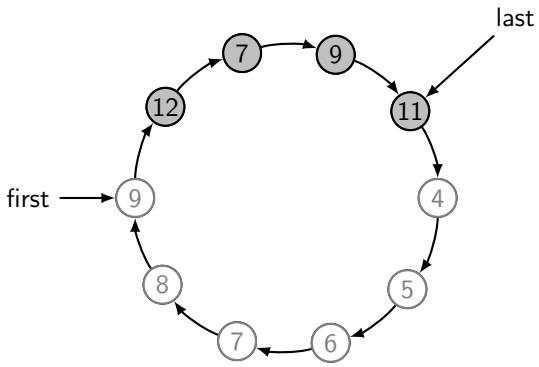
The queue data structure (FIFO) is usually implemented using a linked list. One of the advantages is that the queue can hold an arbitrary number of elements. For each entered value a list element has to be allocated, though.

Einmal allozierte, aber in der Queue nicht mehr genutzte Listenelemente sollen nun beim Einfügen wiederverwendet werden. Dazu kombinieren wir das Konzept eines Ringbuffers mit dem Konzept der Queue: die Warteschlange wird, wie üblich, mit einer einfach verketteten Liste implementiert. Wir ergänzen sie aber zu einer Ringstruktur. Wir nehmen an, dass einmal allozierte Listenelemente nie wieder freigegeben werden.

List elements that have been once allocated but are no longer used by the queue shall now be reused for new values. In order to achieve this, we combine the concept of a ring-buffer with that of a queue: the queue is implemented as usual with a singly linked list but the list is connected to a ring structure. We assume that list elements once allocated will never be deallocated again.

Beispielszenario: / *Example scenario*

1. Erstelle neue Warteschlange / *Generate new queue*
2. $\underbrace{\texttt{Put}(1); \texttt{Put}(2); \ldots; \texttt{Put}(9)}$ neun Aufrufe / *nine calls*
3. $\underbrace{\texttt{x} = \texttt{Get}(); \texttt{x} = \texttt{Get}(); \ldots; \texttt{x} = \texttt{Get}()}$ neun Mal / *nine times*
4. `Put(12); Put(7); Put(9); Put(11);`



Betrachten Sie obiges Szenario, die dazugehörige Abbildung und den Code auf der nächsten Seite. Überlegen Sie sich zuerst, unter welchen Bedingungen der Buffer leer oder voll ist. Mit "voll" sei bezeichnet, dass das Einfügen eines zusätzlichen Wertes zu einer Allokation eines neuen Listenelementes führt. "Leer" ist der Buffer dann, wenn `Get` und `Put` gleich oft fehlerfrei aufgerufen wurden. Tipp: `first` zeigt immer auf ein ungenutztes Element ("Wächter").

Consider the scenario above, the corresponding figure and the code on the next page. First think about conditions under which the buffer is empty or full. Here, "full" means that the insertion of a new value leads to an allocation of a new list element. "Empty" means that `Get` *and* `Put` *have been called without error equally often. Hint:* `first` *always points to an unused element ("Sentinel").*

---

4 P (a) Ergänzen Sie die Prozeduren `IsEmpty` und `IsFull` entsprechend.

Complete the code of procedures `IsEmpty` *and* `IsFull` *accordingly*

---

6 P (b) Vervollständigen Sie weiterhin unten stehende Implementation der Prozeduren `Put` und `Get`.

Complete the implementation of procedures `Put` *und* `Get` *below*

```
class Node{
  Node next;
  int value;
  Node (Node n, int v){
    next = n; value =v;
  }
}

// The Ring implements a First-In First-Out (FIFO) data structure
public class Ring {
  Node first, last;

  public Ring(){
    first = last = new Node(null, 0); // sentinel
    last.next = first;
  }

  public boolean IsFull(){
    return last.next == first;
  }
  public boolean IsEmpty(){
    return first == last;
  }

  // post: insert an element at the end of the fifo
  public void Put(int value){
    if (IsFull()){
      last.next = new Node(first,0);
    }
    last = last.next;
    last.value = value;
  }

  // pre: non-empty ring
  // post: logically remove and return value of the
  //       element at the front of the fifo
  public int Get(){

    first = first.next;
    return first.value;

  }
}
```

Ende Aufgabe 5

6 Bäume *(10 Punkte)*

In dieser Aufgabe beschäftigen wir uns mit Morse-Bäumen. Ein Morse-Baum kann zum schnelle Dekodieren einer Zeichenkette aus dem Morse-Alphabet (bestehend aus "." und "-") in einen lesbaren Buchstaben verwendet werden. Im Folgenden sehen Sie die Implementation eines Morse-Baumes. Beantworten Sie die Fragen unten. Ein Morse-Code heisst wohlgeformt, wenn er nur aus den Zeichen "-" und "." besteht. Ein Morse Code String heisst wohlgeformt, wenn er nur aus Morse Codes besteht, welche durch Leerzeichen getrennt sind.

In this task we consider Morse-Trees. A Morse-Tree can be used to quickly decode a string from the morse alphabet (consisting of characters "." and "-") into a readable character. Consider the following implementation of a morse tree. Answer the questions below. A morse code is well-formed if it only consists of the characters "-" and ".". A morse code string is well formed, if it only consists of well formed morse codes separated by blank characters.

```
class TreeNode{
  char key = 0;
  TreeNode left = null;
  TreeNode right = null;
}

public class MorseTree {
  TreeNode root;

  MorseTree(){
    root = new TreeNode();
  }
  // pre: c != 0, code != null, well-formed morse code
  public void Enter(String code, char key){
    TreeNode prev = root;
    for (int i=0; i<code.length(); ++i){
      if (code.charAt(i) == '-'){ // long
        if (prev.right == null) prev.right = new TreeNode();
        prev = prev.right;
      }
      else{ // short
        if (prev.left == null) prev.left = new TreeNode();
        prev = prev.left;
      }
    }
    prev.key = key;
  }
  // pre code != null, well formed morse code string
  String Decode(String code){
    TreeNode prev = root;
    String result = "";
    for (int i= 0; i<code.length();++i){
      while(i<code.length() && code.charAt(i) != ' ' && prev != null){
        if (code.charAt(i) == '-'){ // long
          prev = prev.right;
```

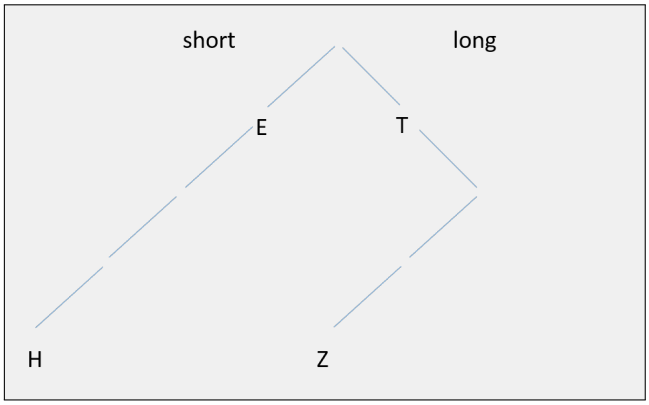
```
            }
            else { // short
                prev = prev.left;
            }
            ++i;
        }
        // end of encoded character
        if (prev == null || prev.key == 0) return "ERROR";
        result += prev.key;
        prev = root;
    }
    return result;
  }
}
```

(a) Skizzieren Sie den Baum welcher durch folgende Anweisungen aufgebaut wird. Es soll erkennbar sein, welche Buchstaben im Baum gespeichert sind.

Sketch the tree that is created by the following statements. It must be visible how the characters are stored in the tree.

4 P

```
MorseTree m = new MorseTree();
m.Enter(".",'E');
m.Enter("-",'T');
m.Enter("....",'H');
m.Enter("--..",'Z');
```



(b) Vervollständigen Sie den code der Methode `MorseTree.Decode` so dass sie zu einem Morse-Code den Text zurückgibt. Beispiel:

Complement the code of `MorseTree.Decode` such that it returns a human readable text from a morse code. Example:

6 P

```
MorseTree m = new MorseTree();
m.Enter(".-",'A');
... // other characters left out for brevity
m.Enter("--..",'Z');
System.out.println(m.Decode("... --- ..."));    // Output SOS
System.out.println(m.Decode(".. -. ..-. ---"));  // Ouput INFO
```

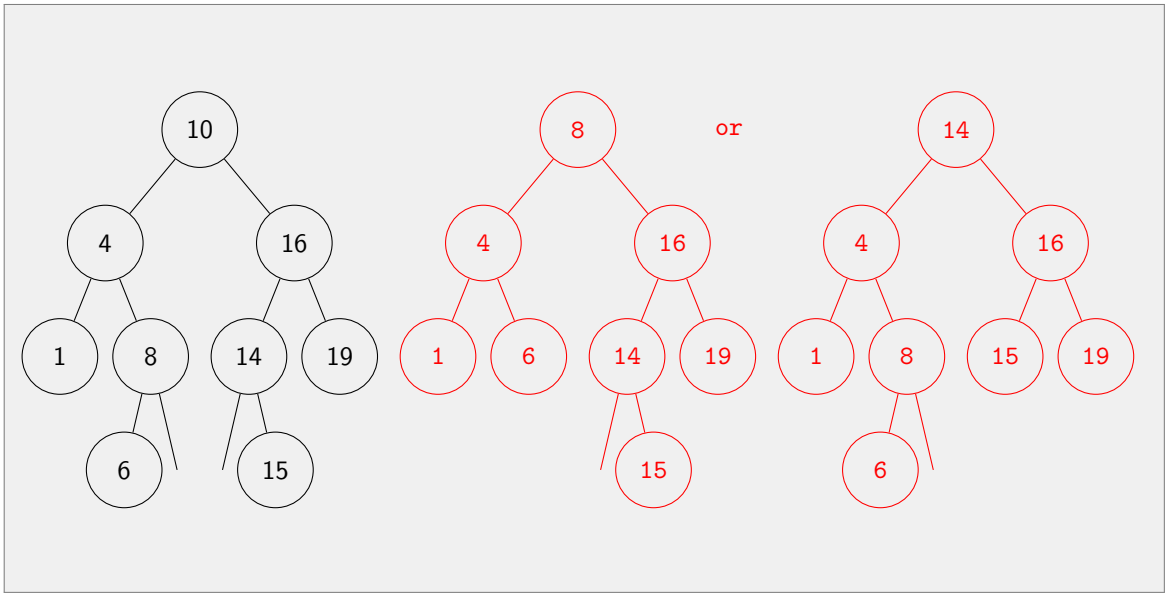
Ende Aufgabe 6

7 Datenstrukturen und Algorithmen *(10 Punkte)*

2 P

(a) Zeichnen Sie neben folgendem **binären Suchbaum** einen binären Suchbaum ein, welcher sich nach dem (nicht balancierenden) Algorithmus der Vorlesung ergibt, wenn man die Zahl 10 löscht.

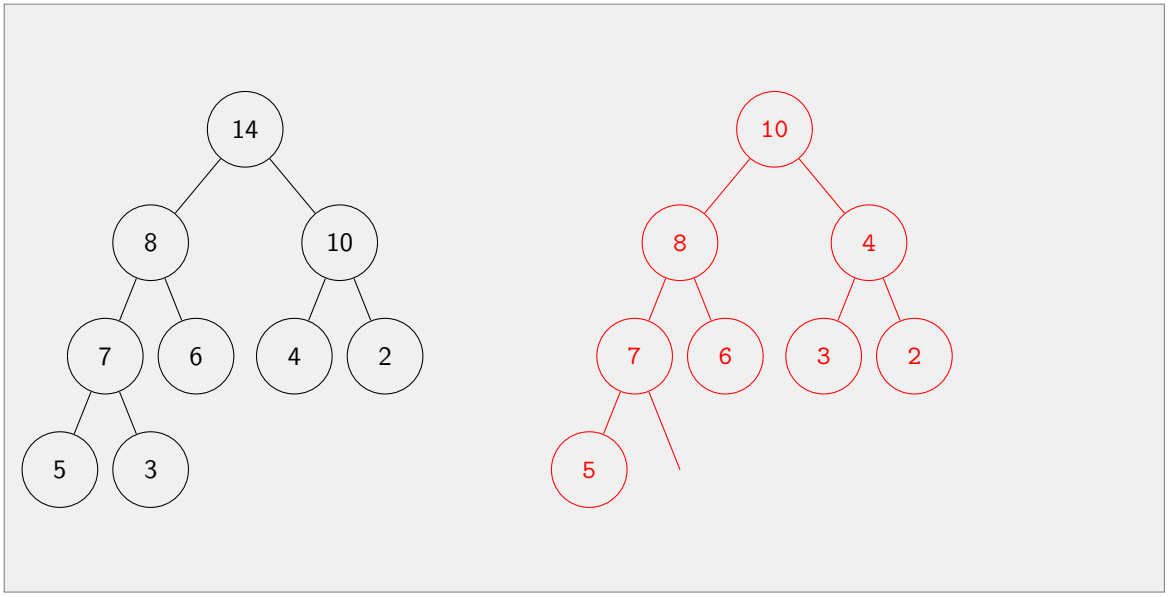
Draw next to the following **binary search tree** *the binary search tree that results from the (non balancing) algorithm shown in the lectures after deletion of the number* 10.



3 P

(b) Zeichnen Sie neben folgendem **Max-Heap** den Max-Heap ein, welcher sich nach dem Algorithmus der Vorlesung ergibt, wenn man die Wurzel extrahiert.

Draw next to the following **Max-Heap** *the Max-Heap that results from the algorithm shown in the lectures after extraction of the root.*



(c) Ergänzen Sie den folgenden Code der Klasse `OnlineMedian` für die schnelle inkrementelle Berechnung ("online Algorithmus") des Medians. Sie können davon ausgehen, dass die Klasse `ArrayHeap` einen (Min- oder Max-) Heap korrekt implementiert

Complement the following code of the class `OnlineMedian` for the fast incremental computation ("online algorithm") of the median. You can assue that the class `ArrayHeap` implements the (Min- or Max-) Heap correctly.

5 P

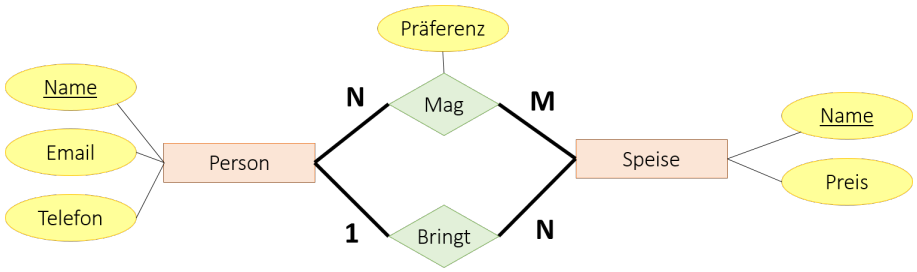
```
class ArrayHeap {
  // minHeap == true ==> implement a min-Heap, otherwise a max-Heap
  ArrayHeap (boolean minHeap) {...}
  // number of elements stored in the heap
  public int NumberElements(){...}
  // insert element to the heap
  public void Insert(double value){...}
  // returns the value of the root
  public double GetRoot() {...}
  //  extract the root and return its value
  public double ExtractRoot() {...}
}
class OnlineMedian {
  ArrayHeap minHeap = new ArrayHeap(true);
  ArrayHeap maxHeap = new ArrayHeap(false);
  int n = 0;
  // post: insert a new value to this data structure
  public void Insert(double value){
    if (minHeap.NumberElements()==0  || value > minHeap.GetRoot())
      minHeap.Insert(value)                                        ;
    else
      maxHeap.Insert(value)                                        ;
    ++n;
    if (maxHeap.NumberElements() < n/2)
      maxHeap.Insert(minHeap.ExtractRoot())                        ;
    else if(maxHeap.NumberElements() > n/2)
      minHeap.Insert(maxHeap.ExtractRoot())                        ;
  }
  // pre:  at least one value has been entered previously
  // post: Return the median of all values that have been entered previously
  public double Get(){
    if (n%2 == 0)
      return (minHeap.GetRoot()+maxHeap.GetRoot())/2               ;
    else
      return minHeap.GetRoot();
  }
}
```

Ende Aufgabe 7

8 Datenbanken *(10 Punkte)*

Sie planen eine Party nach dem Abschluss der Prüfungen. Sie nehmen die Partygäste in einer Datenbank auf. Dabei speichern Sie zu jeder Person den Namen, Telefonnummer und Email-Addresse, wer welche Speise mit welcher Präferenz mag und wer welche Speisen mitbringt. Sie achten darauf, dass keine Speise mehrfach mitgebracht wird und schätzen darüber hinaus den Preis jeder Speise, die mitgebracht wird. Sie zeichen das ER-Diagramm und erhalten folgendes:

You are planning a party for when the exam period is over. You store the party guests into a data base. For each person you store name, telephone number, email address, who likes (Mag) what kind of food (Speise) with preference and who brings (Bringen) what kinds of food. You make sure that no food will be brought more than once. You estimate the price of each dish that is brought along. You draw the ER-Model and come up with the following:



---

4 P

(a) Geben Sie unten das Relationale Modell zu obigem ER-Diagramm an. Sie können entweder die formale Notation wählen oder Tabellen zeichnen. Fassen Sie ggfs. korrekt zusammen, wie in der Vorlesung gezeigt.

Provide the relational model to the ER-Diagram above. You can either use a formal notation or draw tables. Combine tables correctly as presented in the lectures.

```
Personen:{Name, Telefon, Email}
Mögen:{Person, Speise, Präferenz}
Speise:{Name, Preis}
Bringen:{Person, Speise}

Speise und Bringen zusammenfassen ergibt

Personen:{Name, Telefon, Email}
Mögen:{Person, Speise, Präferenz}
Speise: {Name, Preis, mitgebrachtVon}
```

---

2 P

(b) Formulieren Sie folgende Anfrage in SQL: Welche Paare von Personen mögen dieselben Speisen (für jedes Paar genügt eine Übereinstimmung bei den beliebten Speisen).

Formulate the following query in SQL: Pairs of people that like the same dishes (for each pair a single matching dish suffices).

```
SELECT p1.name, p2.name
FROM Personen p1, Personen p2, Mögen m1, Mögen m2
WHERE m1.person == p1.name AND m2.person == p2.name AND m1.Speise == m2.Speise
AND p1.name < p2.name
```

Betrachten Sie folgendes Datenbankschema, welches Niederschlagsmessungen an Messstationen und Messtagen und tageweise Wetterbeobachtungen enthält.

Consider the following data base scheme that contains rain fall data (Niederschlag) measured at station (StationID) and day (Tag) and the weather situation (Lage) for each day (Tag).

***Stationen** (Stations)*

StationID	Name
1	Gipfel
2	Alm
3	Tal

***Messungen** (Measurements)*

StationID	Tag	Niederschlag
1	1	0
3	1	1
1	2	10
2	2	20
3	2	80
2	5	0
3	5	100

***Wetter** (Weather)*

Tag	Lage
1	sonnig
2	heiter
3	bewölkt
4	bewölkt
5	bewölkt
6	heiter
7	sonnig

(c) Geben Sie zu den folgenden Anfragen jeweils das Resultat an. Tabellen können Sie notieren, indem Sie die Überschriften mit Doppelpunkt, Zeilen mit Semikolon und Spalten mit Komma trennen. Beispiel (Tabelle Stationen):
`StationID,Name: 1,Gipfel; 2,Alm; 3,Tal.`

For each of the following queries provide the result. Tables are denoted by separating the headers with a colon, rows by semicolons and columns by commas. Example (Table Stationen):
`StationID,Name: 1,Gipfel; 2,Alm; 3,Tal.`

4 P

(A)
```
SELECT s.Name FROM Stationen s, Messungen m
WHERE m.Niederschlag = 0 AND m.stationID = s.stationID
```

```
Name: Gipfel; Alm
```

(B)
```
SELECT SUM(m.Niederschlag) FROM Messungen m, Wetter w
WHERE m.Tag = w.Tag AND w.Lage="sonnig"
```

```
SUM(m.Niederschlag): 1
```

(C)
```
SELECT s.name, SUM( m.Niederschlag ) FROM Stationen s, Messungen m
WHERE s.stationID = m.stationID GROUP BY s.stationID, s.Name
```

```
Name, SUM(m.Niederschlag): Gipfel,10 ; Alm,20 ; Tal,181
```

(H) $\pi_{Name,Tag,Niederschlag}(Stationen \bowtie Messungen \bowtie \sigma_{Lage='sonnig'}(Wetter))$

```
Name, Tag, Niederschlag: Gipfel,1,0 ; Tal,1,1
```

Ende Aufgabe 8