Informatik II D-BAUG Prüfung Lösung 1.2.2016 *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	9	$\sum$
Punkte										
Maximum	10	10	10	10	10	10	10	10	10	90

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 double recursive(double x){
        if (x < 8)
            return x * recursive( ((int) x) * 2);
        return x;
    }

    static void first(int a[], int b){
        a[0] = b;
    }

    static void second(int a[], int b[]) {
        a = b;
    }

    public static void main(String[] args){
        System.out.println(recursive(4.5)); // output: 36

        System.out.println(recursive(5.2)); // output: 52


        int a[] = {1,2};
        int b[] = {3,4};
        int c[] = {1,2};
        int d[] = {3,4};
        first(a,b[0]);
        System.out.println(a[0] + " " + b[0]); // output: 3 3

        second(c,d);
        System.out.println(c[0] + " " + d[0]); // output: 1 3
    }
}
```

(b) Ergänzen Sie den Code der folgenden Funktion `LargestDiff` so, dass der maximale Wert aller Differenzen $a_{i+1} - a_i$ von aufeinander folgenden Elemente a_i, a_{i+1} im Array `a[]` zurückgegeben wird. Sollte das Array weniger als zwei Elemente enthalten, soll 0 zurück gegeben werden.

Complement the code of the following function `LargestDiff` such that the maximal value of differences $a_{i+1} - a_i$ of two consecutive elements a_i, a_{i+1} of `a[]` is returned. If `a` has less then two elements, then 0 should be returned.

4 P

```
// pre: non-null array a
// post: return largest difference between successive
//       elements of a[]. Returns 0 if not applicable.
static double LargestDiff(double a[])
{
    double max = 0;
    for ( int i = 1; i < a.length; ++i )
    {
        double diff = a[i] - a[i-1];
        if ( diff > max )
            max = diff;
    }
    return max;
}
```

(c) Betrachten Sie folgende Variablendeklarationen

Consider the following variable declarations.

2 P

```
String x = new String("Info");
String y = new String("Info");
String z = y;
String u = "Infk";
```

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

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

`x == y`	`false`
`y != z`	`false`
`x.equals(y)`	`true`
`!z.equals(u)`	`true`

Ende Aufgabe 1

2 Zufallszahlen (Einarmiger Bandit) *(10 Punkte)*

Betrachten Sie folgenden Code, welcher für die Simulation eines einarmigen Banditen gedacht ist.

Consider the following code intended for the simulation of a slot machine.

```
class SlotMachine {
    // return a random integer drawn from the numbers {0, ..., 10}
    static int draw(){
        double rand = Math.random(); // returns a random value in [0,1)
        return (int) (rand * 11);
    }
    // return an array of three draws
    static int[] drawThree(){
        int[] draws = new int[3];
        for (int i = 0; i<3; ++i)
            draws[i]=draw();
        return draws;
    }
    // determine if the player has 3 times or 2 times the same value
    void play() {
        boolean playAgain = true;
        while(playAgain) {
            int[] draw = drawThree();
            if( draw[0]==draw[1] && draw[1]==draw[2] ) {
                printf("3 times the same value! You won the jackpot!");
                playAgain = false;
            } else if ( draw[0]==draw[1]||draw[1]==draw[2]||draw[0]==draw[2] ) {
                printf("2 times the same value! Congratulations!");
                playAgain = true;
            } else {
                printf("No luck. Insert coin and try again");
                playAgain = false;
            }
        }
    }
}
```

(a) Vervollständigen Sie die Methode `SlotMachine.draw` so, dass sie eine Zufallszahl zurückgibt, welche einer Gleichverteilung über der Menge der ganzen Zahlen $\{0, ..., 10\}$ entstammt.

Complement method `SlotMachine.draw` such that it returns a random number drawn from a uniform distribution over the set $\{0, ..., 10\}$ of integers.

3 P

(b) Vervollständigen Sie die Methode `SlotMachine.drawThree` so, dass das zurückgegebene Array drei zufällige Zahlen enthält.

Complement the method `SlotMachine.drawThree` such that the returned array contains three random draws.

2 P

(c) Vervollständigen Sie die Methode `SlotMachine.play` so, dass das Program herausfindet ob es sich um drei mal die gleiche Zahl (Fall 1), um zwei gleiche Zahlen und eine andere (Fall 2), oder um drei verschiedene Zahlen handelt. Nur falls der Spieler zwei gleiche Zahlen hat (Fall 2), darf er ein weiteres mal spielen.

Complement method `SlotMachine.play` such that the program determines if the player drew three times the same value (case 1), two times the same value and one other number (case 2), or three different values (case 3). Only in case the player obtained two times the same number (case 2), he or she can play again.

5 P

Ende Aufgabe 2

3 Mengen *(10 Punkte)*

Betrachten Sie folgende Implemenierung einer Menge. Eine Menge zeichnet sich dadurch aus, dass jedes Element (hier: eine Zeichenkette) genau einmal vorkommt. Freie Stellen im Array werden durch den `null`-Wert repräsentiert. Da wir uns nicht auf eine bestimmte Anzahl an Elementen festlegen wollen, möchten wir, dass sich die Grösse des Arrays dynamisch ändert.

The provided code implements a set. A set is characterized by the fact that each element (here: a string) can only be included once. A free entry in the array is represented by the `null`-value. Since we do not want to have a predetermined number of elements, we would like to grow the size of the array dynamically.

```
public class Menge {

    /* Array for storing the elements */
    protected String [] data = new String[1];

    /* Returns true if the element is in the array, false otherwise. */
    public boolean contains(String element) {
        for(int i=0; i<data.length; ++i) {
            if(data[i] != null && data[i].equals(element)) {
                return true;
            }
        }
        return false;
    }


    /* Insert new element into array. If no space is left, increase
       the array size dynamically. */
    public void insert(String element) {
        if(!contains(element)) {
            for(int i=0; i<data.length; ++i) {
                if (data[i] == null) {
                    data[i] = element;
                    return;
                }
            }

            String[] newData = increase();
            newData[data.length] = element;
            data = newData;
        }
    }
```

```
    /* Double array size and copy elements from old array to new
       array. Return the newly created array. */
    public String[] increase() {
        String[] newData = new String[data.length * 2];
        for(int i = 0; i<data.length; ++i) {
            newData[i] = data[i];
        }
        return newData;
    }
}
```

(a) Vervollständigen Sie die Methode `contains(...)`, so dass sie `true` zurück gibt wenn sich das Element bereits in der Menge befindet. Falls das Element nicht vorhanden ist, soll die Methode `false` zurück geben.

Complete the method `contains(...)` *such that it returns* `true` *if the element is already in the set. If the element is not contained in the set, the method shall return* `false`. 2 P

(b) Vervollständigen Sie die Methode `insert(...)`, so dass sie ein neues Element in die Menge einfügt. Falls das gesamte Array gefüllt ist, wird es dynamisch vergössert, und das neue Element wird an die erste freie Stelle geschrieben.

Complete the method `insert(...)` *such that it inserts a new element into the set. If there is no free space left, the array size will grow dynamically, and the new element will be written to the first free entry.* 4 P

(c) Vervollständigen Sie die Methode `increase()`, so dass sie ein neues Array mit der doppelten Anzahl an Einträgen erstellt welches alle Werte aus dem alten Array enthält.

Complete the method `increase(...)` *such that it creates an array with twice the capacity and which contains all the elements of the old array.* 4 P

Ende Aufgabe 3

4 Objektorientierte Programmierung *(10 Punkte)*

Diese Aufgabe befasst sich mit verschiedenen Aspekten der objektorientierten Programmierung

This task deals with different aspects of object oriented programming.

```
class A {
}

class B extends A {
}

class C extends B {
}

class D extends B {
}

class E extends A {
}

public class Main {
    public static void main(String[] args) {

        D w = new C();   [ ] Ja/Yes [X] Nein/No
        B x = new D();   [X] Ja/Yes [ ] Nein/No
        E y = new A();   [ ] Ja/Yes [X] Nein/No
        A z = new D();   [X] Ja/Yes [ ] Nein/No

    }
}
```

---

4 P

(a) Geben Sie für den obigen Programmcode an ob die Zuweisungen in der main Funktion korrekt (Ja/Yes) oder ungültig sind (Nein/No).

Decide for the code above if the assignments in the main function are correct (Ja/Yes) or invalid (Nein/No).

---

2 P

(b) Erlären Sie das Schlüsselwort 'static' vor einer Methodendeklaration:

Explain the keyword 'static' in front of a method declaration:

Methode die ohne Objekt ausgeführt werden kann. Kann nur auf statische Felder zugreifen.

(c) Erlären Sie den Begriff 'Kapselung': *Explain the term 'encapsulation / information hiding':* 2 P

Verbergen interner Daten und Strukturen vor dem Zugriff von aussen.

(d) Neben 'Kapselung', gibt es noch zwei weitere fundamentale Konzepte der objektorientierten Programmierung. Nennen Sie diese. *In addition to 'encapsulation / information hiding', two other fundamental concepts of object-oriented programming exist. Name both.* 2 P

Vererbung, Polymorphie

Ende Aufgabe 4

5 Verkettete Listen *(10 Punkte)*

Betrachten Sie folgenden Repräsentation einer einfach verketteten Liste.

Consider the following representation of a singly linked list.

```
class ListElement {
    private int data;
    private ListElement next;

    public ListElement(int newData) {
        data = newData;
        next = null;
    }
    public void setNext(ListElement nextElem) { next = nextElem;}
    public ListElement getNext() { return next; }
    public int getData() { return data; }
}

class List {
    public ListElement first = null, last = null;

    public void append(ListElement e){
        if (last == null)
            first = e;
        else
            last.setNext(e);
        last = e;
        e.setNext(null);
    }

    // post: all elements >= 0 from list moved to positive list
    //       all elements <0 from list moved to negative list
    //       this list is empty after the operation
    public void splitBySign(List positive, List negative){
        ListElement curr = first;
        while (curr != null){
            List insertList;
            ListElement next = curr.getNext();
            if (curr.getData()<0)
                insertList = negative;
            else
                insertList = positive;
            insertList.append(curr);
            curr = next;
        }
        first = null; last = null;
    }
```

```
}
```

(a) Vervollständigen Sie die Funktion `splitBySign` so, dass sie die Elemente der bestehende Liste auf die zwei gegebenen Listen `positive` und `negative` aufteilt. Elemente kleiner Null kommen in die `negative` Liste, alle anderen in die `positive` Liste. Die ursprüngliche Liste wird geleert. Vermeiden Sie Elementkopien.

Complete the function `splitBySign` such that it splits a list to the provided lists `positive` and `negative`. Elements smaller than zero belong to list `negative`, all others to list `positive`. The original list is being emptied. Avoid copies of elements.

4 P

(b) Vervollständigen Sie nun folgende Methode von `List` so, dass nun nur der positive Teil einer Liste an eine andere Liste übertragen wird, der negative verbleibt in gleicher Reihenfolge in der gegebenen Liste.

Complete the following method of `List` such that only the positive part of a list is moved to another list, the negative part stays in the given list in its original order.

4 P

```
// post: all elements >= 0 from list moved to positive list
//       all other elements stay in this list
public void movePositive(List positive){
    ListElement prev = null;
    ListElement curr = first;
    while (curr != null){
        ListElement next = curr.getNext();
        if (curr.getData()>=0){
            positive.append(curr);
            if (prev != null)
                prev.setNext(next);
            else
                first = next;
        }
        else
            prev = curr;
        curr = next;
    }
    if (prev != last)
        last = prev;
}
```

(c) Was ist die Komplexität im schlechtesten Fall für das Einfügen eines Elements am Begin einer einfach verketteten Liste der Länge n?

What is the worst-case complexity of inserting an element at the start of a singly linked sorted list with length n?

2 P

$O(1)$ accesses

Ende Aufgabe 5

6 Bäume *(10 Punkte)*

Gegeben sei folgende Repräsentation eines binären Baumes und der nachfolgende Code. (Beachten sie das es sich NICHT um einen SearchTree handelt - daher die Werte in jeder beliebigen Reihenfolge im Baum vorkommen koennen)

Consider the following representation of a binary tree and the subsequent code. (Watch out that it is not a SearchTree - meaning values are not in any order inside the Tree!)

```
public class TreeNode {
  public int value;
  public TreeNode left;
  public TreeNode right;

  public TreeNode(int value) {
    this.value = value;
  }
}

public class Tree {
   TreeNode root;

  public static boolean TreeMax() {
    return max(root);
  }

//pre: a node of the Tree
//post: returns the maximum of itself and its subtrees (its not a Search Tree!)
   protected static int max(TreeNode node) {
        if ( node == null ){
            return Integer.MIN_VALUE;
        } else {
            int left = max(node.left);
            int right = max(node.right);
            if ( left > right && left > node.value )
                return left;
            else {
                if ( right > node.value )
                    return right;
                } else {
                    return node.value;
                }
            }}}
```

(a) Vervollständigen Sie die Methode `max` so, dass sie das maximale Element im Baum zurückgibt. Tipp: verwenden Sie Rekursion.

Complete method `max` such that it returns the maximal element in the tree. Hint: use recursion.

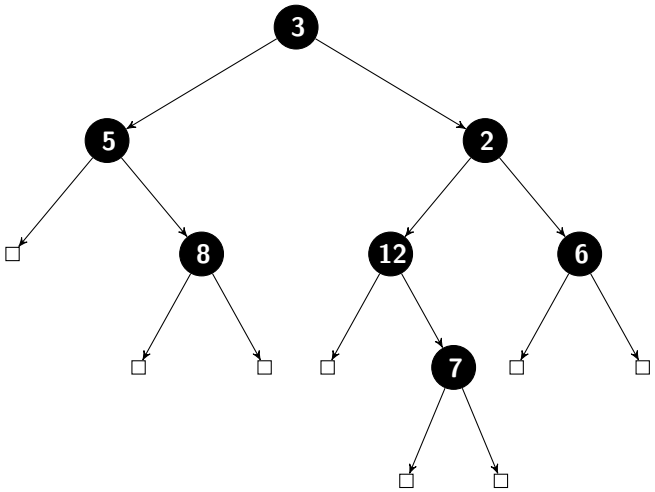
8 P

(b) Folgende Methode `print(TreeNode node)` wird mit der Wurzel des unten dargestellten Baumes aufgerufen. Welchen String gibt die Funktion zurück? (Die Leerzeichen in Ihrem String sind für die Korrektur irrelevant.)

The following method `print(TreeNode node)` is called on the root node on the tree below. Provide the string returned by the method. (Spaces in your string are not relevant for the grading.)

2 P

```
public static String print(TreeNode node) {
        if (node != null) {
          String left = print(node.left);
          String right = print(node.right);
          String result = left + " " + node.key + " " + right;
          return result;
        } else return "";
    }
```



5 8 3 12 7 2 6

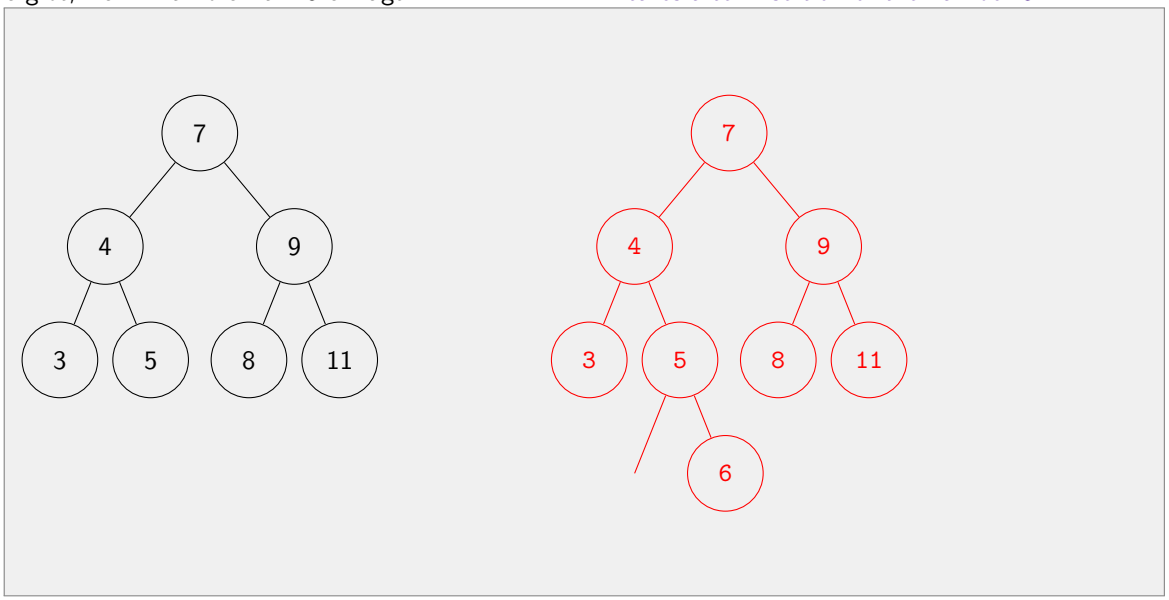
Ende Aufgabe 6

7 Datenstrukturen und Algorithmen *(10 Punkte)*

2 P

(a) Zeichnen Sie neben folgendem **binären Suchbaum** den binären Suchbaum ein, welcher sich nach dem (nicht balancierenden) Algorithmus der Vorlesung ergibt, wenn man die Zahl 6 einfügt.

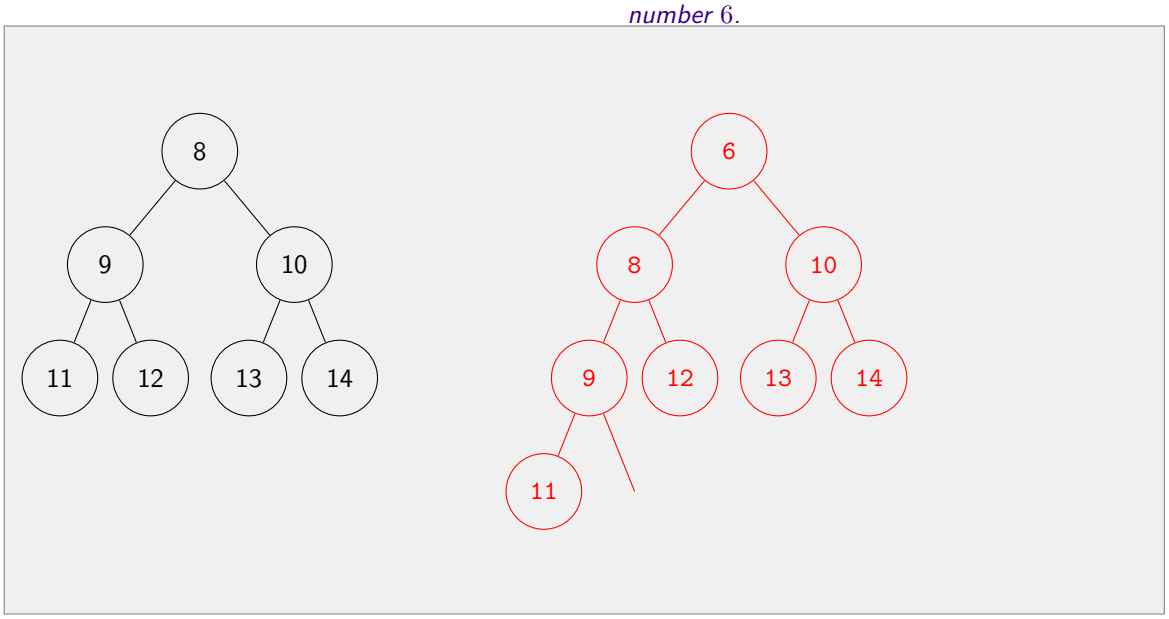
Draw next to the following **binary search tree** *the binary search tree that results from the (non balancing) algorithm shown in the lectures after insertion of the number* 6.



3 P

(b) Zeichnen Sie neben folgendem **Min-Heap** den Min-Heap ein, welcher sich nach dem Algorithmus der Vorlesung ergibt, wenn man die Zahl 6 einfügt.

Draw next to the following **Min-Heap** *the Min-Heap that results from the algorithm shown in the lectures after insertion of the number* 6.



(c) Beschreiben Sie stichwortartig einen Algorithmus für die schnelle inkrementelle Berechnung ("online Algorithmus") des Mittelwerts.

Describe in note form an algorithm for the fast incremental computation ("online algorithm") of the mean.

2 P

Verwendung der Formel $\mu_{n+1} = (\mu_n * n + x)/(n+1)$. Update von μ in jedem Schritt.

(d) Beschreiben Sie stichwortartig einen Algorithmus für die schnelle inkrementelle Berechnung ("online Algorithmus") des Medians.

Describe in note form an algorithm for the fast incremental computation ("online algorithm") of the median.

3 P

Verwendung zweier Heaps: min- und max-heap für oberen und unteren Teil der Daten. Nimm neuen Wert hinzu: wenn kleiner als max vom max-heap, dann in max heap, sonst in min-heap. Balancieren der heaps: wenn ein heap 2 Elemente mehr enthält als der andere, dann schiebe das min oder ma in den anderen Heap. Median ergibt sich als min, max oder Mittelwert, je nach Anzahl Elemente.

Ende Aufgabe 7

8 Algorithmen *(10 Punkte)*

5 P

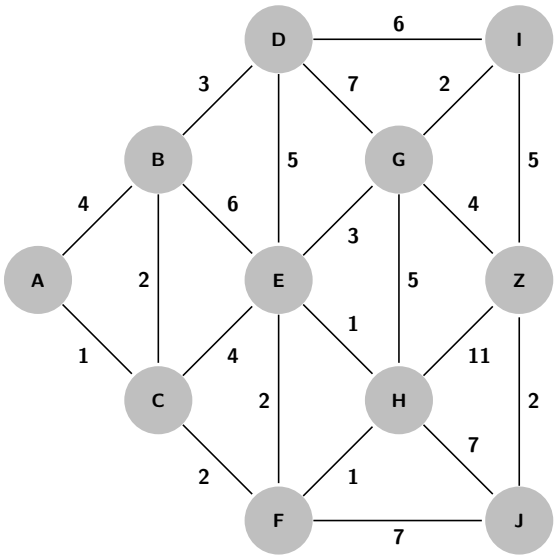
(a) Gegeben sei ein einfaches Polygon P ohne Selbstüberschneidung. P sei spezifiziert durch eine Liste von Eckpunkten aus $\mathbb{R}^2$. Darüber hinaus sei ein Punkt $q \in \mathbb{R}^2$ gegeben. Beschreiben Sie in Stichworten einen Algorithmus zur Bestimmung, ob der Punkt q im Inneren des Polygons P liegt.
Sie können Rechenungenauigkeiten der Fliesskommarechnung in Ihrer Erklärung ignorieren.

Consider a simple polygon P without self intersection. Let P be specified as a list of vertices in $\mathbb{R}^2$. Moreover consider a point $q \in \mathbb{R}^2$. Describe in note form an algorithm to determine if the point q is located inside P.
You are free to ignore inaccuracies of floating point computation in your description.

Auswahl einer beliebigen Geraden, z.B. Horizontale. Schnitt dieser Horizontalen mit allen Segmenten des Polygons. Abzählen aller Schnitte, welche sich auf einem der beiden Halbstrahlen von q aus befinden, bei der Horizontalen also alle Punkte s mit $s_x < q_x$. Anzahl ungerade: innen.

Betrachten Sie folgenden Graphen, welcher alle möglichen Wege zwischen "Städten" A, B, C (etc.) mit deren Längen darstellt.

Consider the following graph that shows all possible ways between "cities" A, B, C (etc.) together with their lengths.



(b) Geben Sie einen kürzesten Weg von A nach Z und dessen Länge an.

Provide a shortest Path from A to Z and its length.

1 P

A,C,E,G,Z: 12

(c) Geben Sie zu jedem Punkt des Graphen die kürzeste Weglänge zu A an.

Tipp: notieren Sie sich die Schritte des Algorithmus von Dijkstra in obigen Graphen

To each point of the graph specify the shortest path length to A.

Tip: in the graph above, note down the steps from Dijkstra's algorithm.

4 P

A:0, C:1, B:3, D:6, E:5, F:3, G:8, H:4, J:10, I:10, J:10, Z:12

Ende Aufgabe 8

9 Datenbanken *(10 Punkte)*

Im Auftrag des Olympischen Komitees erstellen sie eine Datenbank in der die Teilnahme von Sportlers an den Spielen festgehalten wird. Dafür haben Sie folgendes Schema entwickelt.

The Olympic Committee has asked you to design a database which tracks the participation of athletes in the games. For this purpose you designed the following schema.

Sportler (Athlete)

Name	**Alter**	**Grösse**
Tom	20	172
Sophie	24	174
Jeff	28	169
Luke	19	178
Lea	21	169

Teilnehmen (Participate)

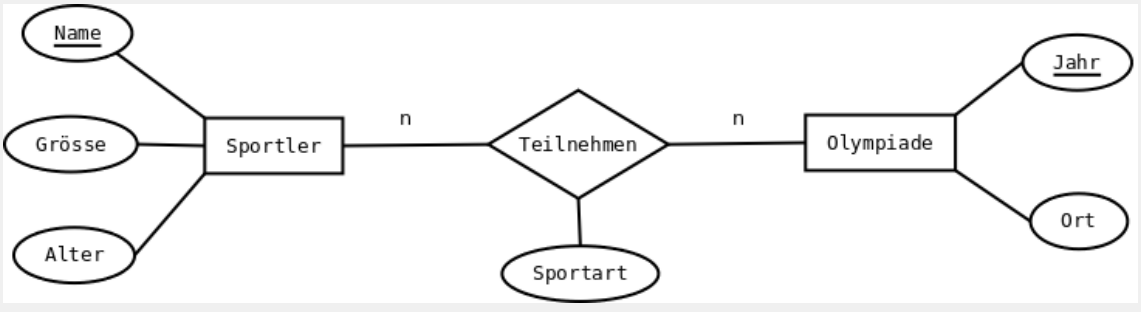
Name	**Jahr**	**Sportart**
Tom	1992	Tennis
Sophie	1992	Triathlon
Tom	1996	Tennis
Jeff	1996	Tennis
Jeff	2000	Badminton
Luke	2000	Kunstturnen
Jeff	2008	Tischtennis
Lea	2008	Triathlon

Olympiade (Olympics)

Jahr	**Ort**
1992	Barcelona
1996	Atlanta
2000	Sydney
2008	Peking

---

4 P (a) Geben Sie das entsprechende ER Model an. *Provide the corresponding ER model.*



---

2 P (b) Formulieren Sie folgende Anfrage in SQL:
Wieviele Sportler sind grösser als 170?

Formulate the following query in SQL:
How many athletes are taller than 170?

```
SELECT COUNT(*) FROM Sportler WHERE Grösse > 170
```

(c) Gegeben sind die vier Abfragen (a)-(d). Schreiben Sie die entsprechenden Buchstaben zu den nachfolgenden Antworten. Es gibt Antwortmöglichkeiten ohne entsprechende Abfrage.

Given the four SQL queries (a)-(d), write the corresponding characters next to the results. Some results do not have a mapping query. 4 P

a)
```
SELECT s.Name
FROM Sportler s
WHERE s.Alter < 22
AND s.Grösse > 170
ORDER BY s.Alter DESC
```

b)
```
SELECT s.Grösse
FROM Sportler s, Teilnehmen t
WHERE t.Sportart = "Triathlon"
AND s.Name = t.Name
ORDER BY s.Alter DESC
```

c)
```
SELECT s.Name
FROM Sportler s, Teilnehmen t, Olympiade o
WHERE o.Ort = 'Barcelona'
AND o.Jahr = t.Jahr
AND t.Name = s.Name
ORDER BY s.Grösse ASC
```

d)
```
SELECT DISTINCT s.Name
FROM Teilnehmen t1, Teilnehmen t2, Sportler s
WHERE t1.Name = t2.Name
AND t1.Jahr != t2.Jahr
AND s.Name = t1.Name
ORDER BY s.Grösse ASC
```

A	Tom, Luke	*Tom, Luke*
	169, 174	*169, 174*
B	174, 169	*174, 169*
	Tom, Luke, Lea	*Tom, Luke, Lea*
	Jeff	*Jeff*
	Barcelona	*Barcelona*
	Tom, Sophie, Jeff	*Tom, Sophie, Jeff*
D	Jeff, Tom	*Jeff, Tom*
C	Tom, Sophie	*Tom, Sophie*
	172, 174, 178	*172, 174, 178*
	Tom, 20, Luke, 19	*Tom, 20, Luke, 19*
	Sophie, Lea	*Sophie, Lea*

Ende Aufgabe 9