Programmiertechnik II
Klausur WS 2023/24
Angewandte Informatik Bachelor

Prof. Dr. Oliver Bittel

Name	
Matrikelnummer	

Aufgabe	Punkte	
1	12	
2	9	
3	12	
4	21	
5	17	
6	18	
7	23	
8	8	
Summe	**120**	

1. Beachten Sie die fettgedruckten Textteile.
2. Schreiben Sie die Lösungen in die Aufgabenblätter.
3. Viel Erfolg!

Aufgabe 1 (12 Punkte)

Die Klassen `ListNode` und `DataNode` und die Funktion `main` sind wie folgt definiert.

```
1  public class Aufgabe1 {
2
3      public static class ListNode {
4          public ListNode next;
5          public DataNode dataList;
6          public ListNode(ListNode p) {
7              this.next = p;
8              this.dataList = null;
9          }
10     }
11
12     public static class DataNode {
13         public DataNode next;
14         public int data;
15         public DataNode(DataNode p, int x) {
16             this.next = p;
17             this.data = x;
18         }
19     }
20
21     public static void main(String[] args) {
22         ListNode lst = new ListNode(null);
23         lst = new ListNode(lst);
24         lst = new ListNode(lst);
25
26         DataNode d1 = new DataNode(null, 5);
27         DataNode d2 = new DataNode(null, 7);
28         d2 = new DataNode(d2, 3);
29         lst.dataList = d1;
30         lst.next.next.dataList = d2;
31
32         for (ListNode l = lst; l != null; l = l.next) {
33             System.out.print("[");
34             for (DataNode p = l.dataList; p != null; p = p.next)
35                 System.out.print(p.data + ",␣");
36             System.out.println("]");
37         }
38     }
39 }
```

a) Geben Sie ein Speicherbelegungsbild (bitte neben dem Quellcode eintragen) für die Variable **`lst`** an, nachdem die Zeile 24 von `main` ausgeführt wurde. (4 Punkte)

b) Geben Sie ein Speicherbelegungsbild (bitte neben dem Quellcode eintragen) für die Variable **`lst`** an, nachdem die Zeile 30 von `main` ausgeführt wurde. (5 Punkte)

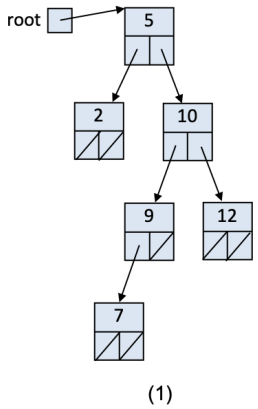
c) Was wird auf die Konsole ausgegeben, wenn die `for`-Schleife in Zeile 32 bis 37 durchlaufen wird? (3 Punkte)

Aufgabe 2 (9 Punkte)

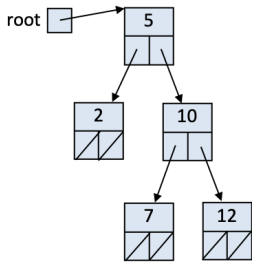
Die Knoten eines binären Suchbaums sind durch die Klasse `Node` definiert.

```
class Node {
   int data;
   Node left;
   Node right;
}
```

a) Schreiben Sie eine Folge von Java-Anweisungen (**keine Schleife und keine new-Aufrufe**), die den binären Suchbaum (1) in (2) überführt. (3 Punkte)

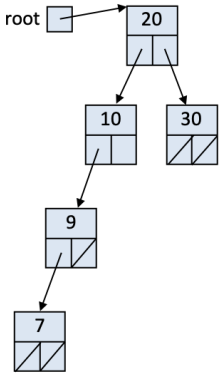


(1)

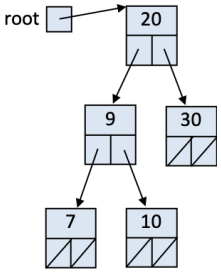


(2)

b) Schreiben Sie eine Folge von Java-Anweisungen (**keine Schleife und keine new-Aufrufe**), die den binären Suchbaum (3) in (4) überführt. (6 Punkte)



(3)



(4)

Aufgabe 3 (12 Punkte)

Das 11-elementige Feld a = {10, 12, 20, 8, 16, 14, 6, 4, 18, 2, 10} wird mit **Quicksort ohne 3-Median-Strategie** sortiert. Außerdem ist Quicksort so modifiziert, dass Teilfelder mit genau 2 Elementen mit einem einfachen Vertauschungsschritt sortiert werden.

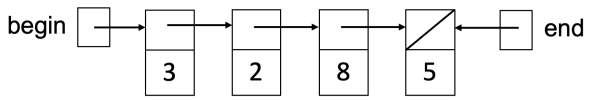
Tragen Sie in die abgebildete Tabelle ein, wie sich das Feld a beim Sortieren ändert (wie in der Vorlesung).

Geben Sie außerdem die **Aufrufstruktur** von Quicksort an. Beachten Sie, dass in der Aufrufstruktur auch Aufrufe für leere Teilfelder vorkommen können.

10	12	20	8	16	14	6	4	18	2	10

Aufgabe 4 (21 Punkte)

Es soll eine Klasse für linear verkettete Listen mit int-Zahlen **ohne Hilfskopfknoten** realisiert werden. Es wird eine Referenz auf den Anfang `begin` und eine Referenz auf das Ende `end` der Liste gespeichert.



Gegeben ist eine rudimentäre Klasse `List`.

```
public class List {
   static private class Node {
      private Node next;
      private int data;
      private Node(Node p, int x) {this.data = x; this.next = p;}
   }
   private Node begin, end;

   public List() {
      begin = end = null;
   }

   // ...
}
```

a) Schreiben Sie eine Methode `contains(x)`, die `true` zurückliefert, falls `x` in dieser Liste enthalten ist. (4 Punkte)

b) Schreiben Sie eine Methode `add(x)` mit Rückgabetyp `void`. `add(x)` hängt das Element `x` an das Ende dieser Liste an. Implementieren Sie `add` **ohne Schleife**, indem die Instanzvariable `end` verwendet wird. (4 Punkte)

c) Schreiben Sie einen Konstruktor `List(List l)` der einer tiefe Kopie der Liste l erstellt. (4 Punkte)

d) Schreiben Sie eine Methode `startsWith(l)`, die prüft, ob die Liste `l` mit dem Anfang dieser Liste übereinstimmt. In folgendem Beispiel stimmt die Liste `t` mit dem Anfang der Liste `s` überein: (6 Punkte)

```
List s = new List();
s.add(1); s.add(2); s.add(3); s.add(5 ); ...   // s = 1,2,3,5,2,3,7
List t = new List();
t.add(1); t.add(2); t.add(3);        // t = 1,2,3

boolean b = s.startsWith(t);         // true
```

e) Wie muß die Methode `add` aus b) verändert werden, so dass verkettete Aufrufe möglich sind? (3 Punkte)

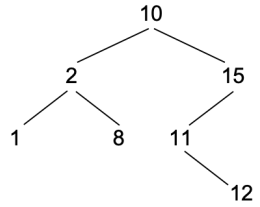
```
l.add(5).add(3).add(7).add(1);
```

Aufgabe 5 (17 Punkte)

Eine Klasse `BinarySearchTree` für binäre Suchbäume für int-Zahlen ist rudimentär definiert.

```
public class BinarySearchTree {
   private static class Node {
      private int data;
      private Node left;
      private Node right;
   }
   private Node root = null;
   // ...
}
```

Die folgende Abbildung zeigt einen binären Suchbaum mit Integer-Zahlen der Höhe 3.



a) Geben Sie den binären Suchbaum an, nachdem die Zahl 10 im abgebildeten Baum gelöscht wurde. (2 Punkte)

b) Schreiben Sie eine Methode `height()`, die die Höhe eines binären Suchbaums zurückliefert. Die Höhe eines leeren Baums ist als -1 definiert. Vervollständigen Sie folgenden Code. **Schleifen sind nicht erlaubt!** (6 Punkte)

```
public int height() {
   return heightR(root);
}

private int heightR(Node p) {









}
```

c) Schreiben Sie eine Methode `collect(Predicate<T> pred)`, die alle Daten im Baum, die das Prädikat `pred` erfüllen, in eine Liste speichert und zurückliefert. Vervollständigen Sie folgenden Code. **Schleifen sind nicht erlaubt!** (7 Punkte)

```
public List<Integer> collect(Predicate<Integer> pred) {









}

private void collectR(Node p, Predicate<Integer> pred, List<Integer> l)
{









}
```

d) Wie muss die collect-Methode aus c) aufgerufen werden, um alle geraden Zahlen im Baum `t` aufzusammeln? Ergänzen Sie das folgende Hauptprogramm (2 Punkte)

```
public static void main() {
   BinarySearchTree t = new BinarySearchTree();
   t.add(10);
   t.add(15);
   // ...

   List<Integer> list = ...


}
```

Aufgabe 6 (18 Punkte)

In der Klasse `RestaurantBewertung` ist eine Map `isLikedBy` definiert, die für jedes Restaurant alle Personen speichert, die das Restaurant gut finden. Da nur Namen relevant sind, sind Personen und Restaurants vom Typ String.

```
public class RestaurantBewertung {
   private Map<String, Set<String> > isLikedBy = new TreeMap<>();
   // ...
}
```

a) Definieren Sie eine Methode `addLike(res,pers)`, die speichert, dass Person `pers` das Restaurant `res` gut findet. Beispiel: (4 Punkte)

```
RestaurantBewertung resBew = new RestaurantBewertung();
resBew.addLike("Defne", "Peter");         // Peter mag Defne
resBew.addLike("Defne", "Petra");         // Petra ebenso
resBew.addLike("Suppengrun", "Petra");    // Petra mag auch Suppengrun
resBew.addLike("Suppengrun", "Markus");   // Markus ebenso
```

b) Schreiben Sie eine Methode `getLikes(res)`, die die Menge aller Personen zurückliefert, die das Restaurant `res` mögen. Falls `res` in der Restaurantbewertung nicht vorkommt, wird `null` zurückgeliefert. (2 Punkte)

c) Schreiben Sie eine Methode `getRestaurants(pers)`, die die Menge alle Restaurants zurückliefert, die Person `pers` gut findet. (6 Punkte)

d) Schreiben Sie eine Methode `top10()`, die die 10 Restaurants mit den meisten Likes ausgibt. Es soll jeweils der Name des Restaurants und Anzahl der Likes sortiert nach der Anzahl der Likes ausgegeben werden. Sollte es in der Restaurantbewertung weniger als 10 Restaurants geben, dann werden alle Restaurants ausgegeben. (6 Punkte)

Aufgabe 7 (23 Punkte)

Messwerte bestehen aus Sensortyp (”CO2”, ”Temp”, ...), Postleitzahl (PLZ) des Standorts des Sensors und Messwert.

```
public record Messwert(
      String typ,    // "CO2", "Temp", ...
      String plz,    // Standort als PLZ
      double wert)   // Messwert
{
   public static void main(String[] args) {
      List<Messwert> mLst = new LinkedList<>();
      mLst.add(new Messwert("CO2", "78364", 418.1));
      mLst.add(new Messwert("Temp", "78464", 21.5));
      mLst.add(new Messwert("CO2", "78362", 412.2));
      mLst.add(new Messwert("CO2", "78462", 410.8));
      mLst.add(new Messwert("CO2", "78464", 419.4));
      mLst.add(new Messwert("Temp", "78567", 22.3));
   }
}
```

Lösen Sie folgende Teilaufgaben.

a) Geben Sie den Typ des folgenden Lambda-Ausdrucks an. (2 Punkte)

```
incr = (Messwert m, Double p) -> new Messwert(m.typ(), m.plz(), m.wert()*p);
```

b) Was gibt folgende Anweisung aus. (2 Punkte)

```
Messwert m = new Messwert("Temp", "78464", 20.0);
System.out.println(incr.apply(m, 1.5));
```

c) Erhöhen Sie in der Liste `mLst` (siehe `main`) alle Messwerte mit Hilfe von **`replaceAll`** (siehe Interface `List`) und **`incr`** aus a) um 10%. (3 Punkte)

d) Prüfen Sie mit Hilfe von **Stromoperationen**, ob in der Liste `mLst` alle Temperaturwerte (Sensortyp = "Temp") in einem Intervall [a,b] liegen und geben Sie das Ergebnis der Prüfung aus. (4 Punkte)

e) Bestimmen Sie in der Liste `mLst` mit Hilfe von **Stromoperationen** den Standort (PLZ) mit der größten gemessenen CO2-Konzentration und geben Sie den Standort aus. (4 Punkte)

f) Was wird durch folgende Anweisung auf die Konsole ausgegeben? (4 Punkte)

```
System.out.println(mLst.stream()
            .map(m->m.plz).reduce("plz", (s1,s2)-> s1+":"+s2));
```

g) Was wird durch folgende Anweisungen auf die Konsole ausgegeben? (4 Punkte)

```
Map<String, List<String>> map = mLst.stream().collect(
   Collectors.groupingBy(
      Messwert::typ,
      TreeMap::new,
      Collectors.mapping(Messwert::plz,Collectors.toList())
   )
);
System.out.println(map);
```

Aufgabe 8 (8 Punkte)

Gegeben sei folgende statische Methode `f`.

```
static <T> boolean f(TreeSet<T> t, T e) {
   for (T x : t)
      if (x.equals(e))
         return true;
   return false;
}
```

a) Was leistet die Funktion `f(t,e)`? (2 Punkte)

b) Schätzen Sie die Laufzeit T(n) der Funktion `f(t,e)` ab, falls eine Menge `t` mit n Elementen übergeben wird (O-Notation). (2 Punkte)

c) Durch welche wesentlich effizientere Anweisung ließe sich der Aufruf `f(t,e)` ersetzen? (2 Punkte)

d) Welche Laufzeit T(n) hat Ihre Lösung aus c)? (2 Punkte)