PROGRAMMING II – ABSTRACT DATA TYPES

Unassessed Coursework 4: *Heaps and AVL trees*

The aims of this coursework is to practice:

- the implementation of Heaps and AVL trees.
- using heaps and hash tables, by writing examples of access procedures, high-level procedures and small problem applications.

1. Consider the static implementation of a (generic) ADT MaxHeap illustrated diagrammatically in slide 6 of Unit 8. Implement the following access procedure:

```
public void add(T newItem) throws HeapException
```

//post: adds the newItem in the heap, preserving the ordering of the heap.
//post: throws exception if the addition is violated.

2. Consider a class `MaxHeap<T>` that provides a static implementation of a MaxHeap. Implement the following auxiliary procedure `heapRebuild` of the class `MaxHeap<T>` .

```
private void heapRebuild(T[ ] heap, int root, size)
```

//pre "root" is the index of the semi-heap's root that has to be placed at
//pre: the proper position. Size is the index of the next available space in `heap`.

3. Assume the existence of a class `MaxHeap<T>` that implements an ADT Heap of elements of type T. Implement an ADT Priority queue defined by the interface `PQInterface<T>` given below. For simplicity, ignore the cases of exception.

```
public interface PQInterface<T>{
      public boolean pqIsEmpty();
      public void pqInsert(T newElem)
      //pre: none
      //post: Inserts newElem in the priority queue at its proper position.
      public T pqDelete( )
      //pre: none
      //post: Retrieves and deletes the element with highest priority.
}
```

4. Consider the class diagram for an AVL tree given in Slides 14 and 15 of Unit 6 (on AVL trees). Assume `getHeightDifference`() to be an access procedure of the class `TreeNode<K>`.

(a) Give the implementation of the class TreeNode<K>

(b) Give the implementation of the access procedure `insert(V value)` of the class `AVLTree<K>` . Assume only one generic type and that the elements are order according to their value. Give also the implementation of all supporting auxiliary methods.