

Binära sökträd

Tema: Binära sökträd, interfacet Comparable.

- U 1. Visa hur ett binärt sökträd ser ut efter insättning av element med följande nycklar: 2, 5, 1, 7, 9, 6, 3, 0, 8, 4.
- U 2. I vilken ordning besöks noderna i trädet från uppgift U 1 om man traverserar det preorder, inorder respektive postorder. Vilken traverseringordning ska man använda om man vill behandla noderna i växande ordning?
- U 3. Vilken är den minsta möjliga resp. största möjliga höjden i ett binärt sökträd med n stycken noder?
- U 4. Kan man använda ett binärt sökträd som hjälp för att sortera ett stort antal element?
- U 5. Antag att vi har följande klasser för att representera ett binärt sökträd:

```
public class BinarySearchTree<E extends Comparable<? super E>> {
    private Node<E> root;

    // metoder för insättning, borttagning,...

    private static class Node<E extends Comparable<? super E>> {
        private E element;
        private Node<E> left;
        private Node<E> right;

        private Node(E element) {
            this.element = element;
        }
    }
}
```

Lägg till en metod `toString()` i klassen `BinarySearchTree`. Metoden ska returnera en sträng med innehållet i trädet i växande ordning. Elementen i trädet ska i strängen separeras med ett newline-tecken (`'\n'`).

- U 6. Lägg till följande metod i klassen `BinarySearchTree`:

```
public void printPart(E min, E max);
```

Metoden ska skriva ut innehållet i de noder som är större än eller lika med `min` och mindre än eller lika med `max` i växande ordning. Använd rekursiv teknik och undvik att besöka fler noder än nödvändigt.

- U 7. Vad innebär det att ett binärt sökträd är balanserat? Varför vill man balansera binära sökträd?

Implementera interfacet Comparable

- U 8. Antag att vi ska lagra böcker i ett binärt sökträd av typen `BinarySearchTree` från uppgift U 5. En bok beskrivs av följande klass:

```
public class Book {
    private String isbn;
```

```
private String title;
private String author;

public Book(String isbn, String title, String author) {
    this.isbn = isbn;
    this.title = title;
    this.author = author;
}
}
```

- a) Vad händer om vi försöker deklarera och skapa ett binärt sökträd enligt följande:

```
BinarySearchTree<Book> books = new BinarySearchTree<Book>();
```

- b) En bok identifieras av sitt ISBN-nummer. Ändra i klassen Book så att den implementerar interfacet Comparable.