

---

---

# THE JAVA PROGRAMMING LANGUAGE

## THE BINARY SEARCH TREE

University of Wrocław  
Institute of Computer Science

*Paweł Rzechonek*

---

---

### Subexercise 1

Define a generic class to maintain a dynamic set in the binary search tree `BST<T>`. This class should be a wrapper for a homogeneous structure formed of objects of type `BSTNode`. Your class should implement the dictionary operations: `search()` check whether the item exists in the tree, `insert()` add a new value into the tree, and `delete()` remove a given value from the tree. of a given element defined Declare the operation in the interface `Dictionary<T>`. Note, that each operation on a binary search tree (especially searching) require comparisons between nodes.

```
class BST<T extends Comparable<T>> implements Dictionary<T>
{
    private class BSTNode <T extends Comparable<T>>
    {
        BSTNode<T> left, right, parent;
        T data;
        // ... dictionary operations
    }

    private BSTNode<T> root;

    // ... dictionary operations

    public String toString () { /*...*/ }
}
```

When you will try to insert into the tree the value `null` your program should throw the exception `NullPointerException`. Append to the class `BST` a method giving the number of items in the tree `size()` and a method removing all items from the tree `clean()`.

Define all your classes, interfaces, and exceptions in package `algorithms`.

### Subexercise 2

Append to the class representing `BST` a method `iterator()`, which will create and return an iterator associated with the tree. The iterator should implement an interface `Iterator<T>`. Your iterator should be sensitive to any changes in the tree — if a tree during the iterator

walk will be modified, the next use of the iterator should result in throwing an exception `IllegalStateException`.

```
class BSTIterator<T extends Comparable<T>> implements Iterator<T>
{
    // ... iterator operations
}
```

Define all your classes, interfaces, and exceptions in package `algorithms`.

### Subexercise 3

Finally, write a program that will reliably test all the operations defined in the BST and in the associated iterator. Test the BST on the data type `Integer` and `String`.

### Requirement

Create, build, and run your application in *NetBeans*, or *Eclipse*, or *IntelliJ* IDE.

### Hint

Some information about BST can be obtained on the website:

[http://en.wikipedia.org/wiki/Binary\\_search\\_tree](http://en.wikipedia.org/wiki/Binary_search_tree)