

KURS JĘZYKA JAVA

DRZEWO BST

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Zdefiniuj klasę sparametryzowaną do pamiętania zbioru dynamicznego w drzewie binarnych poszukiwań `BST<T>`. Klasa ta ma być opakowaniem dla homogenicznej struktury tworzonej wewnątrz na węzłach typu `Node`.

```
interface Dict <T extends Comparable<T>>
{
    // ... metody słownikowe
}
```

Twoja klasa powinna implementować operacje słownikowe (sprawdzać czy element o zadanej wartości istnieje `search()`, dodawać nowy element do zbioru `insert()`, usuwać element zadany `remove()`, wyznaczać element najmniejszy `min()` i największy `max()`) zdefiniowane w interfejsie `Dict<T>`.

```
class BST <T extends Comparable<T>> implements Dict<T>
{
    private class Node <T extends Comparable<T>>
    {
        Node<T> left, right, parent;
        T data;
        // ...
    }

    private Node<T> root;

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

Przy próbie włożenia do drzewa wartości `null` należy zgłosić wyjątek `NullPointerException`. Dopisz też metody podające ilość elementów w zbiorze `size()` i usuwającej wszystkie elementy z drzewa `clear()`.

Definicje wszystkich klas, interfejsów i wyjątków umieść w pakiecie `algorithms`.

Na koniec napisz program, który będzie rzetelnie testował działanie zdefiniowanego przez Ciebie drzewa BST. Testy przeprowadź na danych typu `Integer` oraz `String`.

Uwaga.

Program należy napisać, skompilować i uruchomić w zintegrowanym środowisku programistycznym *NetBeans* albo *IntelliJ*.