

Java

(studia wieczorowe)

zadanie nr 4
2 listopada 2005

Zdefiniuj klasę `DrzewoBST`, która będzie reprezentować drzewo poszukiwań binarnych przechowujące w węzłach pary *klucz-wartość* (klucz to napis, a wartość to liczba całkowita).

```
public class Para implements Comparable
{
    public final String klucz;
    public final int wartosc;

    public int Para (String kl, int wart) { /*...*/ }

    public int compareTo (Object ob) { /*...*/ }
}
```

Na drzewie tym będziemy wykonywać podstawowe operacje słownikowe: wstawianie, usuwanie i wyszukiwanie. Operacje te zdefiniuj w interfejsie słownikowym.

```
public interface Sownikowe
{
    void wstaw (Para p);
    Para usun (String klucz);
    Para szukaj (String klucz);
}
```

Klasa `DrzewoBST` powinna być tylko opakowaniem dla homogenicznej dynamicznej struktury danych opartej na węzłach zagnieżdżonej klasy `Wezel`. Klasa `DrzewoBST` powinna być wyposażona w inteligentną metodę klonującą.

```
public class DrzewoBST implements Sownikowe, Cloneable
{
    protected class Wezel implements Sownikowe, Cloneable
    {
        protected Wezel lewy;
        protected Wezel prawy;
        protected Para wartosc;
        // ...
    }

    protected Wezel korzen;

    // ...
}
```

Zdefiniuj rekurencyjną metodę drukującą zawartość drzewa w porządku *inorder* (wykorzystaj metodę `toString()` zdefiniowaną w klasie `DrzewoBST` i w klasie `Wezel` w wersji rekurencyjnej).

W osobnym pliku umieść program, który ze standardowego wejścia wczyta wartość n , która ma być liczbą całkowitą $1 \leq n \leq 1000$, potem kolejno n par (k, w) , gdzie k jest kluczem (łańcuchem znaków)

a w wartością (liczbą całkowitą). Umieść przeczytane pary (k, w) w drzewie binarnych poszukiwań, a na koniec wydrukuj zawartość całego drzewa na standardowe wyjście.

Paweł Rzechonek
Wrocław, 26 października 2005.