

Java (laboratorium)

zadanie nr 5

Zdefiniuj klasę reprezentującą element typu *nazwa-wartość* w taki sposób, aby elementy te można było ze sobą porównywać przy pomocy metody `compareTo` oraz klonować przy pomocy metody `clone()`.

```
public class Para implements Cloneable, Comparable
{
    public final String nazwa ;    // klucz
    public final double wartosc ; // wartość
    // ...
    public Para (String nazwa, double wartosc) throws NullPointerException
    { /* ... */ }
    public String toString ()
    { /* ... */ }
    public Object clone ()
    { /* ... */ }
    public int compareTo (Object ob)
    { /* ... */ }
}
```

Następnie zdefiniuj interfejs `Słownik`, który będzie definiował trzy podstawowe operacje słownikowe: wstawianie (elementu typu `Para`), usuwanie (względem klucza) i wyszukiwanie (względem klucza). Przy wyszukiwaniu funkcja powinna zwracać wartość związaną z zadaniem kluczem, a gdy odpowiedniego elementu nie ma powinna zgłosić stosowny wyjątek. Interfejs ten powinien rozszerzać interfejs `Cloneable`.

Dalej, należy zdefiniować klasę obsługującą dwukierunkową listę uporządkowaną `SortLista`, złożoną z ciągu węzłów typu `Wezeł`, z których każdy przechowuje jedną parę typu *klucz-wartość* (w naszym przypadku będą to obiekty klas `Para`). Zdefiniowana lista ma implementować interfejs `Słownik`. Klasa `SortLista` ma być klasą, która nadzoruje operacje na liście, a zasadnicza część implementacji operacji słownikowych powinna być przerzucona na klasę `Wezeł`.

Na koniec napisz programy testujące zdefiniowaną przez Ciebie strukturę listową. Pierwszy test ma sprawdzać poprawność klonowania (sklonuj listę, dokonaj różnych zmian w obu listach, na końcu porównaj ich końcowe zawartości). Drugi, to test wydajnościowy: wstaw do listy 1 000 000 losowo wygenerowanych par i zmierz programowo czasy 10 czasów pośrednich (kolejno po 100 000, 200 000, ... 1 000 000 elementów).

Program należy opatrzyć stosownymi komentarzami dokumentacyjnymi i na końcu wygenerować dokumentację poleceniem `javadoc` — należy wygenerować jedną dokumentację dla wszystkich zdefiniowanych w programie klas narzędziowych (mają to być klasy publiczne, a więc zdefiniowane w osobnych plikach) i oddzielnie dla programów testujących (tu powinna się znaleźć instrukcja obsługi dla użytkownika).

Paweł Rzechonek
Wrocław, 4 listopada 2004.