

Java (laboratorium)

zadanie nr 4

Zadanie będzie polegało na zdefiniowaniu klasy reprezentującej *listę jednokierunkową*. Lista ta ma przechowywać elementy będące parą *nazwa-wartość*.

```
public class Para
{
    public final String nazwa ;
    public final double wartosc ;
    // ...
    public Para (String nazwa, double wartosc) throws NullPointerException
    { /* ... */ }
    public String toString ()
    { /* ... */ }
}
```

Sama lista to ciąg połączonych ze sobą *węzłów*.

```
public class Węzeł
{
    protected Węzeł następny ;
    // ...
    public Węzeł (Para para) throws NullPointerException
    { /* ... */ }
    public String toString ()
    { /* ... */ }
}
```

Klasa *Lista* ma być klasą, która nadzoruje operacjami na liście. Powinna ona być klasą abstrakcyjną, z dwiema abstrakcyjnymi metodami — jedna do *wstawiania* elementów a druga do ich *usuwania* z listy.

```
public abstract class Lista
{
    protected Węzeł lista = null ;
    protected int ilość = 0 ;
    // ...
    public abstract void wstaw (Para para) throws NullPointerException ;
    public abstract Para usuń () throws IllegalStateException ;
    public String toString ()
    { /* ... */ }
}
```

Wszystkie wymienione klasy mają zdefiniowaną metodę `toString()`, aby można było oglądać zawartość listy (czyli wszystkich znajdujących się na niej elementów). Definicje tych klas umieść w pakiecie *narzedzia*.

Kolejnym krokiem będzie zdefiniowanie wyspecjalizowanych wersji takiej listy: *kolejki* i *stosu*. Kolejka to lista, do której możemy wstawiać elementy tylko na koniec, a usuwać tylko z początku listy. Stos natomiast jest taką odmianą listy, w której operacje wstawiania i usuwania mogą być wykonywane tylko z przodu listy.

Na koniec napisz programy testujące zdefiniowane przez Ciebie struktury danych. Testowanie ma się odbywać w sposób interaktywny, a więc program ma realizować wydawane na konsoli polecenia. Powinniśmy mieć możliwość wydawania następujących rozkazów:

- sprawdzenie liczby elementów na liście, na przykład:

```
> rozmiar  
2  
>
```

- wstawienie elementu do listy, na przykład:

```
> wstaw x 127.5  
>
```

- usunięcie elementu z listy, na przykład:

```
> usun  
(y: 63.75)  
>
```

- wypisanie wszystkich elementów listy, na przykład:

```
> wypisz  
{(x: 127.5), (z: 31.875)}  
>
```

- wyjście z programu, na przykład:

```
> koniec
```

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, 19 października 2004.