
KURS PROGRAMOWANIA W JAVIE

UPORZĄDKOWANA LISTA JEDNOKIERUNKOWA Z LICZBAMI RZECZYWISTYMI

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

W pakiecie `narzedzia` zdefiniuj publiczną klasę `Lista`, która będzie reprezentować *uporządkowaną listę jednokierunkową* przechowującą w węzłach wartości liczbowe typu `double`. Klasa `Lista` powinna być tylko opakowaniem dla homogenicznej dynamicznej struktury danych opartej na węzłach niepublicznej klasy `Wezel`. Klasę `Lista` zdefiniuj w taki sposób, aby obiekty tych klas można było ze sobą porównywać (metoda `equals`) i klonować (metoda `clone`).

```
class Wezel implements Cloneable
{
    protected double wart;
    protected Wezel nast;
    // ...
    public Wezel clone () { /*...*/ } // zastosuj rekurencję
    public boolean equals (Object ob) { /*...*/ } // zastosuj rekurencję
    public String toString () { /*...*/ } // zastosuj rekurencję
}
public class Lista implements Cloneable
{
    protected Wezel poczatek;
    // ...
    public Lista clone () { /*...*/ }
    public boolean equals (Object ob) { /*...*/ } // sprawdź typ argumentu
    public String toString () { /*...*/ }
}
```

W klasach `Lista` i `Wezel` zaimplementuj metody do wstawiania, usuwania i wyszukiwania zadanej wartości.

Następnie napisz w klasie `ManipulacjaListami` program, który będzie testował działanie Twoich list. Program ma działać interaktywnie na konsoli: użytkownik wpisuje polecenie z parametrami a program polecenie to interpretuje i wykonuje. Do poleceń tych muszą należeć operacje słownikowe (wstawianie, usuwanie i wyszukiwanie) na wskazanej liście, wyświetlenie zawartości wszystkich list, oraz wyjście z programu. Nazwy poszczególnych list (i ich ilość) mają być przekazane do programu poprzez parametry wywołania. Oto przykładowe wywołanie tego programu i praca użytkownika:

```
> java ManipulacjaListami a b c
komenda: insert a 11
komenda: insert a 19
komenda: insert a 17
komenda: insert a 13
komenda: list a
{11, 13, 17, 19}
komenda: bla bla bla bla bla
?!?
komenda: clone a c
komenda: delete c 13
komenda: list c
{11, 17, 19}
```

```
komenda: insert b 29
komenda: insert b 23
komenda: insert b 13
komenda: delete b 7
komenda: delete b 13
komenda: search b 23
yes
komenda: search b 13
no
komenda: list
a: {11, 13, 17, 19}
b: {11, 17, 19}
c: {23, 29}
komenda: quit
```

Twój program powinien sprawdzać poprawność wpisywanych przez użytkownika komend i ich parametrów. Nieznane komendy i błędne parametry należy ignorować (odpowiedź “?!?” w powyższym przykładzie).