

KURS PROGRAMOWANIA W JAVIE

LISTA

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

W pakiecie **narzędzia** zdefiniuj klasę listę (jedno- albo dwukierunkową) jako kolekcję z danymi typu **Integer**. Klasa **Lista** ma opakowywać homogeniczną strukturę listową zbudowaną na węzłach. Klasę **Węzeł** zdefiniuj jako chronioną klasę wewnętrzną w liście.

```
package narzędzia;

public class Lista
{
    protected class Węzeł
    { /* ... */ }
    protected Węzeł początek;
    //...

    public void wstaw (int poz, int x) throws IndexOutOfBoundsException
    { /* ... */ }
    public int czytaj (int poz) throws IndexOutOfBoundsException
    { /* ... */ }
    public void usuń (int poz) throws IndexOutOfBoundsException
    { /* ... */ }
    public int długość ()
    { /* ... */ }

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

W klasie **Lista** zdefiniuj metody do modyfikowania listy (**wstaw(int,int)** i **usuń(int)**), odczytywania poszczególnych elementów listy (**czytaj(int)**), badania długości listy (**długość()**) oraz prezentowania zawartości listy (nadpisana metoda **toString()**).

Poza pakietem **narzędzia** napisz program testujący zdefiniowaną przez Ciebie listę. Program ma działać interaktywnie i reagować na polecenia wpisywane z klawiatury.

Uwaga.

Program należy skompilować i uruchomić z wiersza poleceń!

Przykład.

Oto przykład, jak można zorganizować sobie program interaktywny z wykorzystaniem wyjątków:

```
public static void main (String[] args) throws IOException
{
    BufferedReader we = new BufferedReader(new InputStreamReader(System.in));
    PrintWriter wy = new PrintWriter(new OutputStreamWriter(System.out),true);
    PrintWriter bl = new PrintWriter(new OutputStreamWriter(System.err),true);
    Lista lista = new Lista(); // lista z danymi
    bl.println("hello");
    bl.print("> "); bl.flush();
    for (String instr=we.readLine(); instr!=null; instr=we.readLine())
        try
        {
            instr = instr.trim();
            if (instr.length()==0||instr.startsWith("//")) // linia pusta lub komentarz
                throw new NullPointerException();
            String[] token = instr.split("\\p{Blank}+");
            if (token[0].equals("insert")||token[0].equals("i"))
            {
                if (token.length<3||token.length>3&&!token[3].startsWith("//"))
                    throw new IllegalArgumentException();
                int pozycja = Integer.parseInt(token[1]);
                int wartość = Integer.parseInt(token[2]);
                if (pozycja<0||pozycja>lista.długość())
                    throw new IndexOutOfBoundsException(token[1]);
                lista.wstaw(pozycja,wartość);
                wy.format("wstawienie: lista[%s] <- %s\n",...); // wymaga uzupełnienia
            }
            else if (token[0].equals("delete")||token[0].equals("d"))
                { /*...*/ }
            else if (token[0].equals("read")||token[0].equals("r"))
                { /*...*/ }
            else if (token[0].equals("length")||token[0].equals("l"))
                { /*...*/ }
            else if (token[0].equals("print")||token[0].equals("p"))
                { /*...*/ }
            else throw new NoSuchMethodException(token[0]);
        }
        catch (NoSuchMethodException ex)
            { bl.format("nieznane polecenie (%s)\n",ex.getMessage()); }
        catch (IndexOutOfBoundsException ex)
            { bl.format("niepoprawna pozycja w liście (%s)\n",ex.getMessage()); }
        catch (NumberFormatException ex)
            { bl.format("argument powinien być liczbą całkowitą\n"); }
        catch (IllegalArgumentException ex)
            { bl.format("zbyt dużo albo zbyt mało argumentów\n"); }
        catch (NullPointerException ex)
            { bl.println("linia pusta lub komentarz za znakami //"); }
        finally
            { bl.print("> "); bl.flush(); }
    bl.println("bye");
}
```