

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
Кафедра систем штучного інтелекту**

Лабораторна робота № 3
з дисципліни
«Об’єктно-орієнтоване програмування»

Виконала:
студентка групи КН-109
Пелешак Ю. М.
Викладач:
Гасько Р. Т.

Львів – 2018 р.

1. Створити клас, що описує зв'язний список. Клас повинен мати можливість додавання елементів та вилучення елементів. Додавання елементів відбувається в кінець списку, вилучення елементів відбувається за порядковим номером. Також створити методи, що дозволяють отримати розмір списку та елемент за його порядковим номером.

```
package com.tasks3.linkedlist;
```

```
public class LinkedList {  
    private Node begin;  
    private Node end;  
    private int num_ofEllem;
```

```
//конструктор без аргументів
```

```
    public LinkedList() {  
        begin = new Node();  
        end = begin;  
        this.num_ofEllem = 0;  
    }
```

```
// Додаємо елемент в кінець списку
```

```
    public void add(Integer data) {  
        end.setNext(new Node());  
        end = end.getNext();  
        end.setData(data);  
        this.num_ofEllem++;  
    }
```

```
// Отримуємо елемент по індексу; повертаємо null, якщо такий елемент недоступний ;
```

```
    public Integer get(int index) {  
        Node tbegin = begin;  
        for(int i = 0; i < index + 1 ; i++)  
        {  
            tbegin = tbegin.getNext();  
        }  
        return tbegin.getData();  
    }
```

```
// Вилучаємо елементу за індексом; повертаємо true у разі успіху або false в іншому випадку
```

```
    public boolean delete(int index) {  
        Node tbegin = begin;  
        if ((index < 0) || (index >= this.num_ofEllem))  
            return false;  
        for (int i = 0; i < index; i++) {  
            if (tbegin != null)  
                tbegin = tbegin.getNext();  
            else
```

```

        return false;
    }
    if(tbegin.getNext().getNext() != null)
    {
        tbegin.setNext(tbegin.getNext().getNext());
    }
    else
    {
        tbegin.setNext(null);
        end = tbegin;
    }
    num_ofEllem--;
    return true;
}

```

//Повертаємо розмір списку: якщо елементів в списку нема то повертаємо нуль

```

    public int size() {
    return this.num_ofEllem;
    }

```

```

}

```

Програму зараховано на Прометеусі.

The screenshot shows the Prometheus online IDE interface. The top bar displays the user 'JuliaPeleshchak/OOP_Ja...' and the task 'Завдання 1 | Практичні'. The address bar shows the URL 'https://edx.prometheus.org.ua/courses/EPAM/JAVA101/2016_T2/courseware/e3bb4f9536e04bdca6b2d2001b0c3eee/5c62f8376ea9403e8641dc6e0e1d81a2/'. The main editor area contains the following Java code:

```

35     return false;
36     for (int i = 0; i < index; i++) {
37         if (tbegin != null)
38             tbegin = tbegin.getNext();
39         else
40             return false;
41     }
42     if(tbegin.getNext().getNext() != null)
43     {
44         tbegin.setNext(tbegin.getNext().getNext());
45     }
46     else
47     {
48         tbegin.setNext(null);
49         end = tbegin;
50     }

```

Below the code editor, the status 'Правильно' (Correct) is displayed. Under the heading 'Результати тесту' (Test Results), there is a table with one row showing 'ВІРНО' (Correct) and a 'See full output' link. At the bottom, there are three buttons: 'ПЕРЕВІРКА' (Check), 'ЗБЕРЕГТИ' (Save), and 'ПОКАЗАТИ ВІДПОВІДЬ' (Show Answer).

2. Створити класи для зберігання колоди з 36 карт. Використовувати класи заготовки.

```

package com.tasks3.carddeck;
public class Deck {

```

```
private Card cards[];  
private int size;  
private Card used[];
```

//конструктор без аргументів

```
public Deck()  
{  
    int index = 0;  
    this.size = 36;  
    cards = new Card[this.size];  
    used = new Card[this.size];  
    for(int i = 0; i < 4; i++)  
    {  
        for(int j = 0; j < 9; j++)  
        {  
            this.cards[index++] = new Card(Rank.values[j], Suit.values[i]);  
        }  
    }  
}
```

```
public void shuffle() {  
  
    for(int i = 0; i < this.size; i++) {  
        int random_num = (int)(Math.random()*this.size);  
  
        Card tmp = this.cards[i];  
        this.cards[i] = this.cards[random_num];  
        this.cards[random_num] = tmp;  
    }  
}
```

```
public void order() {  
    int index = 0;  
    for(int i = 0; i < 4; i++)  
    {  
        for(int j = 0; j < 9; j++)  
        {  
            boolean create = true;  
            for(int k = size; k < 36; k++)  
            {  
                if((used[k].getRank() == Rank.values[i])&&(used[k].getSuit() ==  
Suit.values[j]))  
                    create = false;  
            }  
            if(create) {  
                this.cards[index].setRank(Rank.values[j]);  
                this.cards[index].setSuit(Suit.values[i]);  
                index++;  
            }  
        }  
    }  
}
```

```

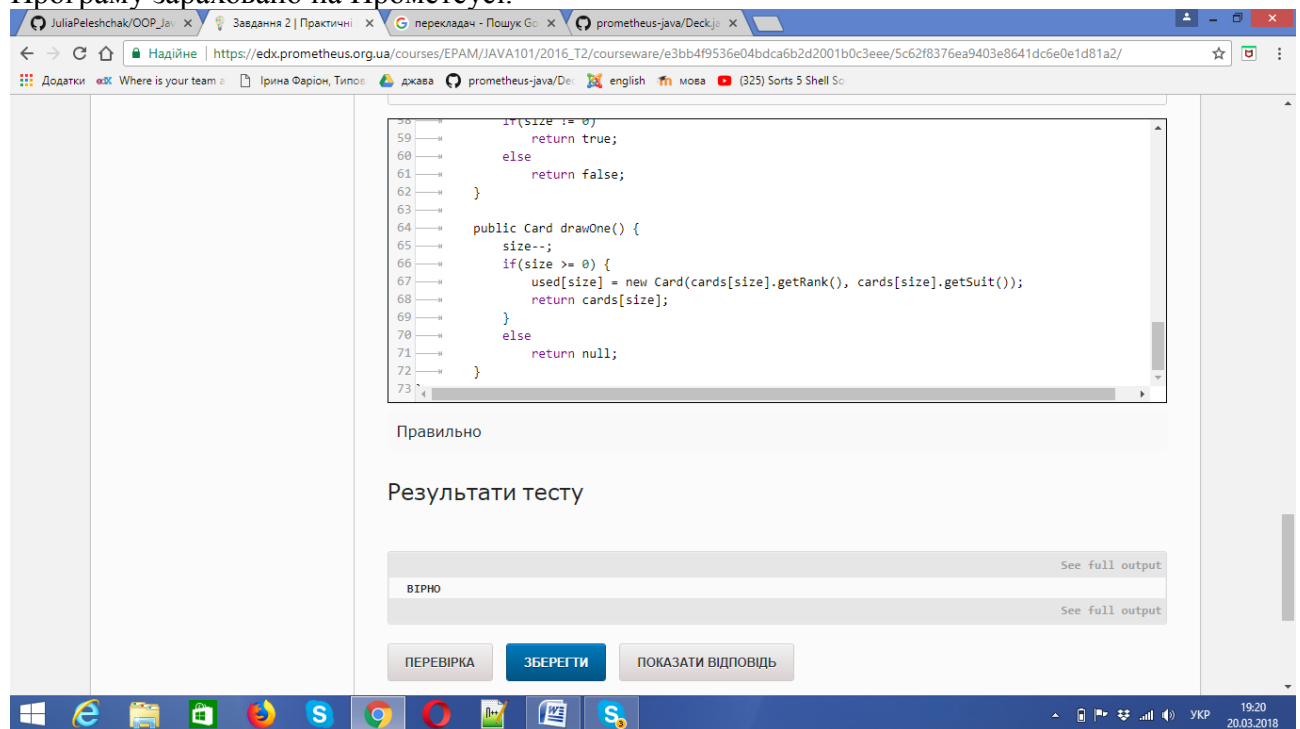
    }
    }
}

public boolean hasNext() {
    if(size != 0)
        return true;
    else
        return false;
}

public Card drawOne() {
    size--;
    if(size >= 0) {
        used[size] = new Card(cards[size].getRank(), cards[size].getSuit());
        return cards[size];
    }
    else
        return null;
}
}

```

Програму зараховано на Прометеусі.



3. Використовуючи рекурсію, вивести на екран задане по порядковому номеру число Фібоначі.

```
package com.tasks3.fibonacci;
```

```
public class Fibonacci
{

```

```
public static long getNumber(int position)
{
```

//якщо позиція менша нуля

```
if (position <= 0)
{
    return -1;
}
```

//якщо позиція =1 або =2 виводимо 1(число фібоначчі на першій і другій позиції = 1)

```
if (position == 1 || position == 2)
{
    return 1;
}
```

```
long[] nums = new long[position];
nums[0] = nums[1] = 1;
```

//якщо число знаходиться на позиції >2

```
for (int i = 2; i < position; i++)
{
    nums[i] = nums[i - 1] + nums[i - 2];
}

return nums[position - 1];
}
```

Програму зараховано на Прометеусі.

The screenshot shows a web browser window with the URL https://edx.prometheus.org.ua/courses/EPAM/JAVA101/2016_T2/courseware/e3bb4f9536e04bdca6b2d2001b0c3eee/5c62f8376ea9403e8641dc6e0e1d81a2/. The browser tabs include "JuliaPeleshchak/OOP_Ja...", "Завдання 3 | Практичні...", "перекладач - Пошук Go...", and "prometheus-java/Deckj...". The main content area displays a Java code editor with the following code:

```
4
5 public class Fibonacci
6 {
7     public static long getNumber(int position)
8     {
9         if (position <= 0)
10        {
11            return -1;
12        }
13
14        if (position == 1 || position == 2)
15        {
16            return 1;
17        }
18
19        long[] nums = new long[position];
```

Below the code editor, the text "Правильно" (Correct) is displayed. Under the heading "Результати тесту" (Test Results), there is a table with two rows, both showing "ВІРНО" (Correct) in the first column and a "See full output" link in the second column.

Результат	Деталі
ВІРНО	See full output
ВІРНО	See full output

At the bottom of the interface, there are three buttons: "ПЕРЕВІРКА" (Check), "ЗБЕРЕГТИ" (Save), and "ПОКАЗАТИ ВІДПОВІДЬ" (Show Answer). The Windows taskbar at the bottom shows the time as 19:24 on 20.03.2018.

Висновок: під час виконання лабораторної роботи №3 я удосконалила свої знання щодо використання зв'язного списку, рекурсії. Краще засвоїла використання класів.