

Міністерство освіти і науки України
Національний університет «Львівська політехніка»



Лабораторна робота №1
з курсу:
«Об'єктно-орієнтоване програмування»

Виконала:
ст. гр. КН-110
Гелетій Софія
Прийняв:
ст. викладач
Гасько Р.Т.

Львів – 2018 р.

Лабораторна робота № 3

Завдання

Пройти завдання третього тижня на Prometheus на курсі JAVA101.

Надійшло | https://edx.prometheus.org.ua/courses/EPAM/JAVA101/2016_T2/courseware/e3bb4f9536e04bdca6b2d2001b0c3eee/0f57...

PROMETHEUS | автор Ігор Деркач: Основи програмування на Java | sonia_heletiy

Головна сторінка Курс Обговорення Прогрес

Заклади

- Вступ до курсу
- Тижень 1
- Тижень 2
- Тижень 3
- Тема 4. Класи
- Тема 5. Блоки
- Тема 6. Об'єкти
- Тема 7. Особливості використання класів
- Тест 3
- Практичні завдання 2
- Тижень 4
- Тижень 5

Тижень 3 > Тест 3 > Тест 3

Додати до закладок

Тест 3

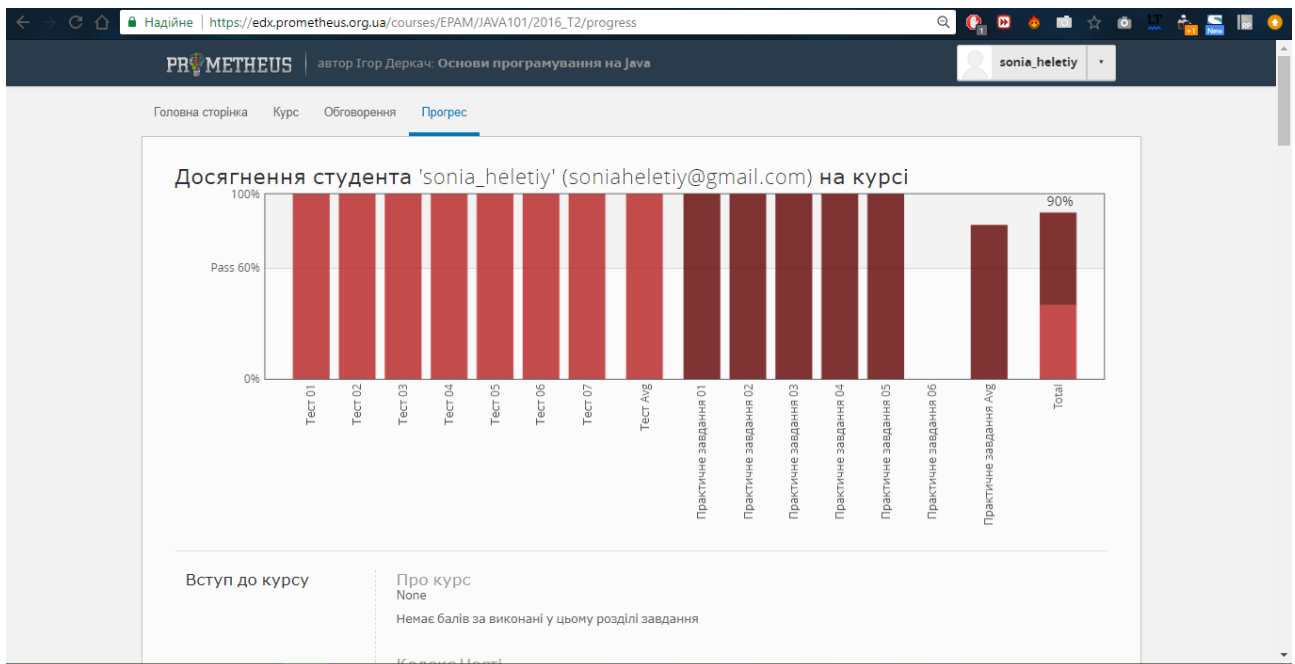
(5/5 балів)

1. Що виведе на екран наступний фрагмент коду?

```
class IntsTest {
    int i;
}

class Main {
    public static void main(String args[]) {
        IntsTest test = new IntsTest();
        System.out.println(test.i);
    }
}
```

0 ✓



1. Створіть 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 створіть методи, що дозволяють отримати розмір списку та елемент за його порядковим номером.

```

public class LinkedList { private Node head; private Node tail; private int
size; public LinkedList() { tail = null; head = null; size = 0; }
public void add(Integer data) {
    Node newNode = new Node();
    newNode.setData(data);
    if (tail == null) {
        tail = newNode;
        head = newNode;
        size++;
    }
    else {
        tail.setNext(newNode);
        tail = newNode;
        size++;
    }
}
public Integer get(int index) {
    Node findNode = head;
    if (findNode == null || index > (size - 1) || index < 0) {
        return null;
    }
    while (findNode != null) {
        for (int i = 0; i < index; i++) {
            findNode = findNode.getNext();
        }
        return findNode.getData();
    }
    return findNode.getData();
}

public boolean delete(int index) {

```

```

Node curNode = head;
Node prevNode = head;
while (curNode != null) {
    if (curNode == null || index > (size - 1) || index < 0) {
        return false;}
    if (head == tail) {
        tail = null;
        head = null;
        size--;
        return true;
    }
    if (index == 0) {
        head = head.getNext();
        size--;
        return true;}

    else {
        while (curNode != null) {
            curNode = curNode.getNext();
            for (int i = 0; i < (index-1); i++) {
                prevNode = curNode;
                curNode = curNode.getNext();
            }
            prevNode.setNext(curNode.getNext());
            size--;
            return true;
        }
    }
}
return true;
}

```

```

public int size() {
    return size;
}
}

```

2. Створіть класи для зберігання колоди з 36 карт

```

public class Deck {
    public int topCard = 36;

    public Card[] cards = new Card[36];

    public Deck() {

        for (int i = 0; i < 36; i++) {
            cards[i] = new Card(Rank.values[i % 9], Suit.values[i % 4]);
        }
    }

    public void shuffle() {
        Card[] left = new Card[18];
        Card[] right = new Card[18];
        Card[] shift = new Card[1];

        for (int n = 0; n < 61; n++) {
            for (int i = 0; i < 18; i++) {
                left[i] = cards[i];
                right[i] = cards[i + 18];
            }

            for (int i = 0, j = 0, k = 0; i < 36; i++) {
                if ((i % 2) == 1) {

```

```

        cards[i] = right[j];
        j++;
//        System.out.println(i + " " + j);
    } else {
        cards[i] = left[k];
        k++;
//        System.out.println(i + " " + k);

    }
}

shift[0] = cards[0];
cards[0] = cards[35];
cards[35] = shift[0];
}
}

public void order() {
    int k = 0;
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 9; j++) {
            cards[k] = new Card(Rank.values[j], Suit.values[i]);
            k++;
        }
    }
}

public boolean hasNext() {
    if (topCard > 0) {
        return true;
    } else {
        return false;
    }
}
}

```

```

    public Card drawOne() {
        if (hasNext()) {
            topCard--;
            return cards[topCard];
        } else {
            return null;
        }
    }
}

```

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

```

public class Fibonacci {

    public long getNumber(int position){

        if (position<1) return -1;

        if (position<=2) return 1;

        return getNumber(position-1)+getNumber(position-2);

    }

    public static void main(String[] args) {

        Fibonacci fib = new Fibonacci();

        System.out.println("f(9)="+fib.getNumber(9));

    }

}

```

Висновок: я навчилася писати перші програми на Java, практично закріпила знання на прикладі.