Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна робота №3

з курсу:

“ОБ’ЄКТНО ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ”

Виконав:
ст. гр. КН-110
Шевчук Ігор
Прийняв:
Гасько Р.Т.

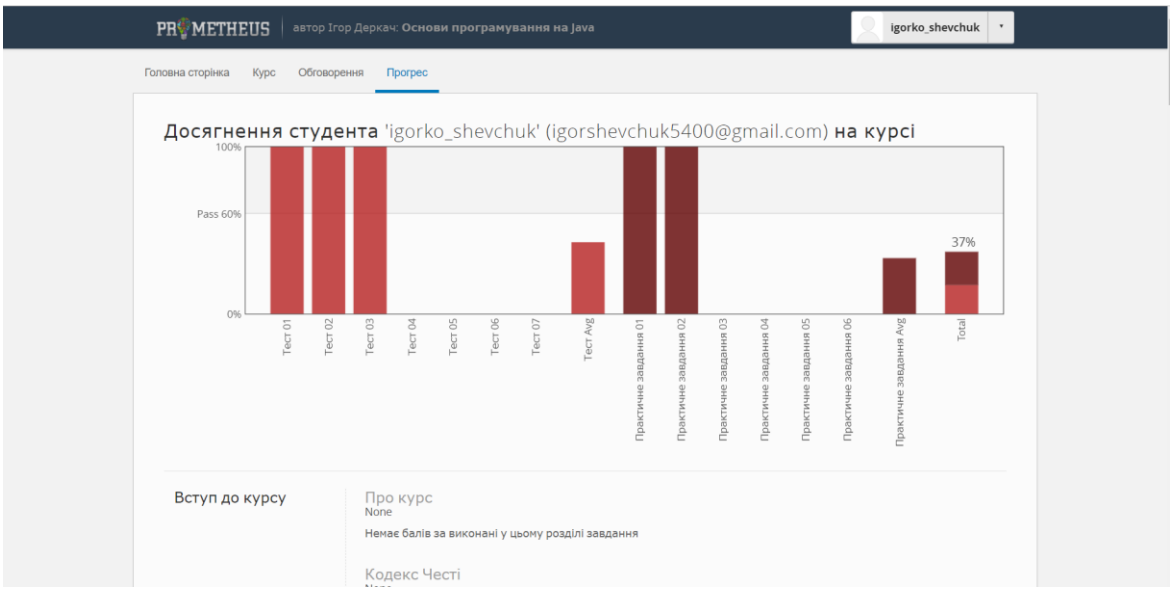
Львів – 2018 р.

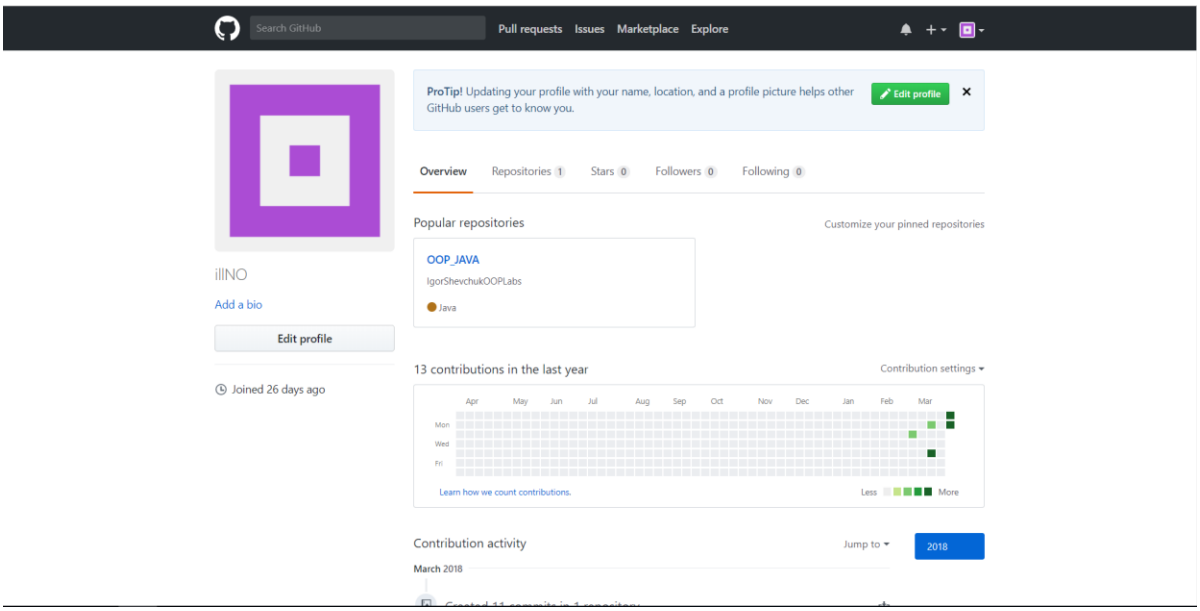
Лабораторна робота №3

Завдання: Опрацювати 3 тиждень на prometheus.org

Виконання роботи

GitHub link - https://github.com/illNO






```
package com.tasks3.linkedlist;


public class LinkedList {
    private Node tail;
    private Node head;
    private int size = 0;

    public LinkedList() {

    }

    public void add(Integer data) {
        Node newNode = new Node();
        newNode.setData(data);
```

```
    if (size == 0)
        head = newNode;
    else
        tail.setNext(newNode);

    tail = newNode;
    size++;
}

public Integer get(int index) {
    return findNodeByIndex(index).getData();
}

public boolean delete(int index) {
    if (findNodeByIndex(index) != null ) {

        if (index != 0)
            findNodeByIndex(index - 1).setNext(findNodeByIndex(index+1));
        else
            head = head.getNext();

        size--;
        return true;
    }
```

```
    return false;
}

public int size() {
    return size;
}

private Node findNodeByIndex(int index) {
    if (index < size && index >= 0) {
        Node curNode = head;
        int curIndex = 0;
        while (curIndex < index){
            if (curNode.getNext() != null) {
                curNode = curNode.getNext();
                curIndex++;
            }

        }
        return curNode;
    }
    else
        return null;
}

public String toString() {
```

```
        StringBuilder resultString = new StringBuilder("[");

        for (int i = 0; i < size; i++) {
            resultString.append(get(i));

            if (i < size-1) resultString.append(", ");
        }
        resultString.append("]");
        return resultString.toString();
    }
}


package com.tasks3.carddeck;

import java.util.*;

public class Deck {

    public Card[] deck = new Card[36];
    private int size = 0;

    public void shuffle() {
        Random rng = new Random(System.currentTimeMillis());
```

```
      for (int i = 0; i < this.deck.length; i++)
      {
         int rRank = rng.nextInt(Rank.values.length);
         int rSuit = rng.nextInt(Suit.values.length);

         this.deck[i] = new Card(Rank.values[rRank], Suit.values[rSuit]);
         System.out.println(this.deck[i].getRank().getName() + " " +
this.deck[i].getSuit().getName());
      }

      this.size = 36;

      //Collections.shuffle(deck);
   }

   public void order() {
      for (int suit = 0, place = 0; suit < 4; suit++)
      {
         for (int rank = 0; rank < 9; rank++, place++)
         {
            this.deck[place] = new Card(Rank.values[rank], Suit.values[suit]);
            System.out.println(this.deck[place].getRank().getName() + " " +
this.deck[place].getSuit().getName());
         }
      }
```

```
        this.size = 36;
    }

    public boolean hasNext() {
        return (this.size > 0);
    }

    public Card drawOne() {
        if (size <= 0)

            return null;

        Card rCard = this.deck[this.size - 1];
        this.deck[this.size - 1] = null;
        this.size--;
        return rCard;
    }
}

package com.tasks3.fibonacci;

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

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

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

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

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

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