

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ  
ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ  
ІНСТИТУТ» Кафедра «Програмна інженерія та інформаційні технології  
управління» Звіт з лабораторної роботи №5

З предмету «Алгоритмізація та програмування»

Виконав

Студент групи КН-36а

Рубан Ю.Д.

Перевірив:

Ст. в. Смолін П. О.

## Робота з масивами та рядками Java. Створення класів

### 1 Завдання на лабораторну роботу

#### 1.1 Індивідуальне завдання

Спроектувати та реалізувати два класи відповідно до індивідуального завдання. У першому з класів повинен бути описаний масив елементів другого класу. Класи повинні мати конструктори, приватні поля та відкриті методи, зокрема методи доступу (сеттери та геттери).

Слід окремо здійснити тестування кожного з класів, після чого окремими методами першого класу реалізувати основне завдання. У функції `main()` першого з класів створити необхідний об'єкт та викликати для нього методи, які реалізують основне завдання. Вивести результати у консольне вікно.

Варіант завдання, який слід реалізувати у програмі, визначається залежно від номеру студента у списку групи.

Виставка	Назва, прізвище скульптора	День	Кількість відвідувачів, коментар	Сумарна кількість відвідувачів, день з найбільшою кількістю відвідувачів, день з найбільшою кількістю слів у коментарі
----------	----------------------------	------	----------------------------------	--

#### 1.2 Ератосфенове решето

Заповнити масив із трьохсот цілих чисел послідовними додатними значеннями. Замінити всі значення, що не є простими числами, деяким від'ємним значенням. Для цього послідовно виключати всі числа - дільники інших чисел. Вивести на екран додатні значення, що залишилися, (прості числа).

#### 1.3 Сортування вибором (задача підвищеної складності)

Проініціалізувати одновимірний масив цілих чисел випадковими значеннями. Здійснити упорядкування масиву методом сортування вибором за таким алгоритмом:

- знаходимо індекс мінімального значення;
- здійснюємо обмін цього значення із значенням першої невідсортованої позиції (обмін не потрібний, якщо мінімальний елемент вже знаходиться на цій позиції);
- сортуюмо решту масиву, виключивши з розгляду вже відсортовані елементи.

Результат вивести на екран.

## 1.4 Знаходження чисел Фібоначчі

Реалізувати функцію обчислення чисел Фібоначчі (до 92-го числа включно) з використанням допоміжного масиву (статичного поля). Параметр функції – номер числа Фібоначчі. Пошук чисел Фібоначчі здійснюється за таким правилом:

$$F(1) = F(2) = 1; F(n) = F(n - 2) + F(n - 1)$$

Під час першого виклику функції масив заповнюється до необхідного числа. Під час наступних викликів число або повертається з масиву, або обчислюється з використанням останніх двох чисел, що зберігаються у масиві з подальшим заповненням масиву. Використовувати тип **long** для представлення чисел.

Здійснити тестування функції для різних значень номерів, що вводяться у довільному порядку.

## 1.5 Вирівнювання рядку

Прочитати аргумент командного рядка і додати в нього пробіли так, щоб довжина рядка дорівнювала заданому числу.

## 1.6 Аббревіатура

Увести з клавіатури рядок з декількох слів. Скласти рядок з перших літер слів з переведенням цих літер у верхній регістр.

## 1.7 Квадратне рівняння

Спроектувати клас для розв'язання квадратного рівняння. Слід передбачити аналіз усіх можливих комбінацій коефіцієнтів та відповідних результатів (два корені, один корінь для випадку виродження у лінійне рівняння, немає розв'язків, безмежна кількість коренів). Метод безпосереднього знаходження коренів (`solve()`) повинен повертати кількість коренів (-1 якщо безмежна кількість коренів). Геттери для  $x_1$  і  $x_2$  повинні повертати корені тільки, якщо було визначено вихідні дані та здійснено пошук коренів.

## Хід виконання роботи:

### Завдання 1:

Спроектовано та реалізовано два класи відповідно до індивідуального завдання. У першому з класів описаний масив елементів другого класу. Класи мають конструктори, приватні поля та відкриті методи, зокрема методи доступу (сеттери та геттери).

Здійснено тестування кожного з класів, після чого окремими методами першого класу реалізовано основне завдання. У функції main() першого з класів створено необхідний об'єкт та викликано для нього методи, які реалізують основне завдання. Виведено результати у консольне вікно.

Код програми:

Клас Main

```
import java.util.*;
public class Main {

    public static void main(String[] args)
    {
        ArrayList<Day> days =new ArrayList<Day>();
        Day d1=new Day(30,"день 1");
        days.add(d1);
        Day d2=new Day(25,"на след день получился большой комментарий");
        days.add(d2);
        Show s=new Show("выставка","автор",days);
        System.out.print("День с макс кол-вом посетителей "+
s.dayWithLargestAmountOfVisitors()+"\nСуммарное кол-во посетителей
"+s.SummOfVisitors()+" \nДень с самым большим комментарием
"+s.dayWithLargestComment());

    }
}
```

## Клас Day:

```
public class Day
{
    Day(int count,String c)
    {
        this.comment=c;
        this.countOfvisitors=count;
    }
    public void setComment(String c) {
        this.comment = c;
    }

    public void setCountOfvisitors(int count) {
        this.countOfvisitors = count;
    }

    public int getCountOfvisitors() {
        return this.countOfvisitors;
    }

    public String getComment() {
        return this.comment;
    }

    private int countOfvisitors;
    private String comment;
}
```

## Клас Show:

```
import java.util.*;
public class Show
{
    Show(){};
    Show(String t,String n,ArrayList<Day> d)
    {
        this.title = t;
        this.name = n;
        this.D=d;
    }
    ArrayList <Day> D = new ArrayList <Day>();
    int SummOfVisitors()
    {
        int res=0;
        for(int i=0;i<D.size();i++)
        {
            res+=D.get(i).getCountOfvisitors();
        }
        return res;
    }
    int dayWithLargestAmountOfVisitors()
    {
        int max=D.get(0).getCountOfvisitors();
        int res=1;
        for(int i=0;i<D.size();i++)
        {
            if (max < D.get(i).getCountOfvisitors())
            {
                max = D.get(i).getCountOfvisitors();
                res=i+1;
            }
        }
        return res;
    }
    int dayWithLargestComment()
    {
        int max=D.get(0).getComment().length();
        int res=0;
        for(int i=0;i<D.size();i++)
        {
            if (max < D.get(i).getComment().length())
            {
                max = D.get(i).getComment().length();
                res=i+1;
            }
        }
        return res;
    }
    void setName(String n){name = n;}
    String getName(){return name;}
    void setTitle(String t){title = t;}
    String getTitle(){return title;}
    private String title;
    private String name;
}
```

Результат виконання:

День с макс кол-вом посетителей 1

Суммарное кол-во посетителей 55

День с самым большим комментарием 2

## Завдання 2:

Заповнено масив із трьохсот цілих чисел послідовними додатними значеннями. Замінено всі значення, що не є простими числами, деяким від'ємним значенням. Виведено на екран додатні значення, що залишилися, (прості числа).

Код програми:

```
public class Main {  
    public static void main(String[] args) {  
        int[] a = new int [300];  
        for(int i =0;i<300;i++)  
        {  
            a[i] = i+1;  
        }  
        for(int i =2 ;i<300;i++)  
        {  
            for(int j = i*i-1;j<300;j+=i)  
            {  
                a[j] = -1;  
            }  
        }  
        for(int i =0 ;i<300;i++)  
        {  
            if(a[i]!=-1)  
                System.out.print(a[i]+" ");  
        }  
    }  
}
```

Результат виконання:

1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103  
107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199  
211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293

### Завдання 3:

Проініціалізовано одновимірний масив цілих чисел випадковими значеннями. Здійснено упорядкування масиву методом сортування вибором

Код програми

```
import java.util.*;
public class Main {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int[] a=new int[10];
        for(int i=0;i<10;i++)
        {
            a[i]=sc.nextInt();
        }
        int min;
        int indexOfMin;
        for(int i=0;i<10;i++)
        {
            min=a[i];
            indexOfMin=i;
            for (int j = i; j < 10; j++)
            {
                if (a[j] < min)
                {
                    min = a[j];
                    indexOfMin = j;
                }
            }
            int temp=a[indexOfMin];
            a[indexOfMin]=a[i];
            a[i]=temp;
        }
        for(int i=0;i<10;i++)
        {
            System.out.print(a[i]+" ");
        }
    }
}
```

Результат виконання

**5 3 0 1 4 6 2 8 9 7**

**0 1 2 3 4 5 6 7 8 9**



## Завдання 4

Реалізовано функцію обчислення чисел Фібоначчі (до 92-го числа включно) з використанням допоміжного масиву (статичного поля). Параметр функції – номер числа Фібоначчі.

Під час першого виклику функції масив заповнюється до необхідного числа. Під час наступних викликів число або повертається з масиву, або обчислюється з використанням останніх двох чисел, що зберігаються у масиві з подальшим заповненням масиву. Використано тип **long** для представлення чисел.

### Код програми

```
import java.util.*;
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        System.out.println(fib(n));
        n=sc.nextInt();
        System.out.println(fib(n));
        n=sc.nextInt();
        System.out.print(fib(n));
    }

    public static long f[] = new long[92];
    public static int last=2;
    public static long fib(int n)
    {
        f[0]=1;
        f[1]=1;
        if(last<n)
        {
            for(int i = last;i<n;i++)
            {
                f[i]=f[i-1]+f[i-2];
            }
            last = n;
            return f[n-1];
        }
        else
        {
            return f[n-1];
        }
    }
}
```

### Результат виконання

23 28657

13 233

25 75025

## Завдання 5

Прочитано аргумент командного рядка і додано в нього пробіли так, щоб довжина рядка дорівнювала заданому числу.

Код програми

```
import java.util.*;
public class Main {
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String s=sc.nextLine();
        int n = sc.nextInt();
        while(n>s.length())
        {
            s+=" ";
        }
        System.out.print(s);
    }
}
```

## Завдання 6

Уведено з клавіатури рядок з декількох слів. Складено рядок з перших літер слів з переведенням цих літер у верхній регістр.

Код програми

```
package com.company;
import java.util.*;
public class Main {
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String s=sc.nextLine();
        String[] splt=s.split(" ");
        String res="";
        for(int i=0;i<splt.length;i++)
        {
            res+=splt[i].substring(0,1);
        }
        System.out.print(res.toUpperCase());
    }
}
```

Результат виконання

forgive or organize lairs

FOOL

## Завдання 7

Спроектовано клас для розв'язання квадратного рівняння. Передбачено аналіз усіх можливих комбінацій коефіцієнтів та відповідних результатів (два корені, один корінь для випадку виродження у лінійне рівняння, немає розв'язків, безмежна кількість коренів). Метод безпосереднього знаходження коренів (solve()) повинен повертати кількість коренів (-1 якщо безмежна кількість коренів). Геттери для  $x_1$  і  $x_2$  повинні повертати корені тільки, якщо було визначено вихідні дані та здійснено пошук коренів.

Код програми

Клас Main

```
import java.util.*;
public class Main {
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        double c = sc.nextDouble();
        Solve roots=new Solve(a,b,c);
        switch (roots.calc())
        {
            case -1: System.out.print("error");break;
            case 1: System.out.print(roots.getX1());break;
            case 2: System.out.print(roots.getX1()+" "+roots.getX2());break;
        }
    }
}
```

## Клас Solve

```
public class Solve
{
    Solve(double m,double n,double p)
    {
        this.a=m;
        this.b=n;
        this.c=p;
    }
    int calc()
    {
        D=b*b - 4*a*c;
        if(D<0){return -1;}
        if(D==0)
        {
            if(a==0)
            {
                if(b==0)
                {
                    return -1;
                }
                x1=-c/b;
                isSolvedWithOneRoot=true;
                return 1;
            }
            x1=-b/(2*a);
            isSolvedWithOneRoot=true;
            return 1;
        }
        else
        {
            if(a==0)
            {
                if(b==0)
                {
                    return -1;
                }
                x1=-c/b;
                isSolvedWithOneRoot=true;
                return 1;
            }
            x1=(-b+Math.sqrt(D))/(2*a);
            x2=(-b-Math.sqrt(D))/(2*a);
            isSolvedWithTwoRoots=true;
            return 2;
        }
    }

    double getX1()
    {
        if(isSolvedWithOneRoot==true||isSolvedWithTwoRoots==true)
            return x1;
        else
            return -23423234;
    }
    double getX2()
    {
        if(isSolvedWithTwoRoots==true)
            return x2;
        else
            return -23424234;
    }
    private double a;
    private double b;
    private double c;
    private double x1;
    private double x2;
    private boolean isSolvedWithOneRoot=false;
    private boolean isSolvedWithTwoRoots=false;
    private double D;
}
```

## **Висновки**

У даній лабораторній роботі я навчився працювати з масивами та рядками, дізнався про алгоритм пошуку простих чисел та навчився створювати класи і працювати з ними