МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №3

Выполнил

Литвинко В.А.

студент группы АС-50

Проверил

Крощенко А. А.

ст. преп. кафедры ИИТ

Брест 2020

Цель работы: научиться создавать и использовать классы в программах на языке программирования Java

ВАРИАНТ 5

Задание 1

5) Множество целых чисел ограниченной мощности – Предусмотреть возможность объединения двух множеств, вывода на печать элементов множества, а так же метод, определяющий, принадлежит ли указанное значение множеству. Класс должен содержать методы, позволяющие добавлять и удалять элемент в/из множества. Конструктор должен позволить создавать объекты с начальной инициализацией. Мощность множества задается при создании объекта. Реализацию множества осуществить на базе одномерного массива. Реализовать метод equals, выполняющий сравнение объектов данного типа.

Код программы:

```
package com.company;

class Set {
    private int maxPower;
    private int power = 0;
    private int[] elements;

    public Set(int maxPower) {
        this.maxPower = maxPower;
        elements = new int[maxPower];
    }

    public Set() {
        this.maxPower = 10;
        elements = new int[maxPower];
    }

    public int getMaxPower() {
        return maxPower;
    }

    public int getElement(int index) {
        return elements[index];
    }

    public int getPower() {
        return power;
    }

    public boolean checkExistElement(int element) {
        for (int i = 0; i < power; i++)
            if (elements[i] == element)
                return true;

        return false;
    }

    public void addElement(int element) {
```

```
        if (power == maxPower)
            return;

        for (int i = 0; i < power; i++)
            if (elements[i] == element)
                return;

        elements[power] = element;
        power++;
    }

    public void removeElement(int element) {
        for (int i = 0; i < power; i++)
            if (elements[i] == element) {
                for (int j = i; j < power - 1; j++)
                    elements[j] = elements[j + 1];

                power--;
                return;
            }
    }

    public boolean union(Set set) {
        int[] addedElements = new int[set.getMaxPower()];
        int addedElementsLength = 0;

        for (int i = 0; i < set.getPower(); i++) {
            if (!checkExistElement(set.getElement(i))) {
                addedElements[addedElementsLength] = set.getElement(i);
                addedElementsLength++;
            }
        }

        if (power + addedElementsLength > maxPower)
            return false;

        for (int i = 0; i < addedElementsLength; i++)
            addElement(addedElements[i]);

        return true;
    }

    public void print() {
        System.out.print("Set: ");

        for (int i = 0; i < power; i++) {
            System.out.print(elements[i]);
            if (i + 1 != power)
                System.out.print(",");
        }

        System.out.println();
    }

    public boolean equals(Set set) {
        if (set.getPower() != power)
            return false;

        for (int i = 0; i < power; i++)
            if (!set.checkExistElement(elements[i]))
                return false;

        return true;
    }
```

```
    public String toString() {
        StringBuilder str = new StringBuilder();

        str.append("Set: ");

        for (int i = 0; i < power; i++) {
            str.append(elements[i]);
            if (i + 1 != power)
                str.append(",");
        }

        return str.toString();
    }
}

public class Main {

    public static void main(String[] args) {
        Set set1 = new Set();
        for (int i = 0; i < 10; i++)
            set1.addElement(i);

        Set set2 = new Set(15);

        for (int i = 0; i < 10; i++)
            set2.addElement(i + 5);

        System.out.print("Set1 to string format: ");
        System.out.println(set1.toString());

        System.out.print("Set2 to string format: ");
        System.out.println(set2.toString());

        boolean result = set2.union(set1);

        if (!result)
            System.out.println("error union set2 <- set1.");
        else {
            System.out.print("Result union set2 <- set1: ");
            set2.print();
        }

        result = set1.union(set2);

        if (!result)
            System.out.println("error union set1 <- set2.");
        else {
            System.out.print("Result union set1 <- set2: ");
            set1.print();
        }

        set1.removeElement(5);
        set1.removeElement(7);
        set1.addElement(12);

        System.out.print("Set1 after remove 5, 7 and add 12: ");
        set1.print();

        System.out.println("Set1 exists 1? " + set1.checkExistElement(1));
        System.out.println("Set1 exists 11? " + set1.checkExistElement(11));

        System.out.println("Set1 equals Set2? " + set1.equals(set2));
```

```
        Set set3 = new Set(20);

        for (int i = 0; i < 5; i++)
            set3.addElement(i);

        set3.addElement(6);
        set3.addElement(8);
        set3.addElement(9);
        set3.addElement(12);

        System.out.print("Set3 to string format: ");
        System.out.println(set3.toString());

        System.out.println("Set1 equals Set3? " + set1.equals(set3));
    }
}
```

Вывод:

```
Set1 to string format: Set: 0,1,2,3,4,5,6,7,8,9
Set2 to string format: Set: 5,6,7,8,9,10,11,12,13,14
Result union set2 <- set1: Set: 5,6,7,8,9,10,11,12,13,14,0,1,2,3,4
error union set1 <- set2.
Set1 after remove 5, 7 and add 12: Set: 0,1,2,3,4,6,8,9,12
Set1 exists 1? true
Set1 exists 11? false
Set1 equals Set2? false
Set3 to string format: Set: 0,1,2,3,4,6,8,9,12
Set1 equals Set3? true

Process finished with exit code 0
```

Задание 2

5) Составить программу, которая моделирует заполнение гибкого диска (1440 Кб). В процессе работы файлы могут записываться на диск и удаляться с него.
С каждым файлом (File) ассоциированы следующие данные:
• Размер
• Расширение
• Имя файла
• Как файлы могут трактоваться и директории, которые в свою очередь содержат другие файлы и папки.
Если при удалении образовался свободный участок, то вновь записываемый файл помещается на этом свободном участке, либо, если он не помещается на этом участке, то его следует разместить после последнего записанного файла. Если файл превосходит длину самого большого участка, выдается аварийное сообщение. Рекомендуется создать список свободных участков и список занятых участков памяти на диске.

Код программы:

```
package com.company;

import com.company.exceptions.FileEmptyException;
import com.company.exceptions.FileExistsException;
import com.company.exceptions.NoDirectoryException;
import com.company.exceptions.NoPlaceException;

import java.io.FileNotFoundException;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Storage storage = new Storage();

        boolean key = true;
        byte code;
        Scanner scanner = new Scanner(System.in);

        while(key)
        {
            for(int i=0; i<15; i++)
                System.out.println();

            System.out.println("1: вывести список файлов и директорий");
            System.out.println("2: вывести список свободных сегментов");
            System.out.println("3: вывести список занятых сегментов");
            System.out.println("4: добавить файл");
            System.out.println("5: добавить папку");
            System.out.println("6: удалить файл");
            System.out.println("7: удалить папку");
            System.out.println("8: выйти");

            code = Byte.parseByte(scanner.nextLine());

            if(code==1)
                storage.printFiles();
            else if(code==2)
                storage.printFreeSegments();
            else if(code==3)
                storage.printOccupiedSegments();
            else if(code==4)
            {
                String path, directory;
                System.out.println("Введите путь к файлу:");
                path = scanner.nextLine();
                System.out.println("Введите путь к файлу на гибком диске:");
                directory = scanner.nextLine();

                try
                {
                    storage.writeFile(path, directory);
                }
                catch(FileEmptyException e)
                {
                    System.out.println("Ошибка: файл пуст");
                }
                catch(FileExistsException e)
                {
                    System.out.println("Ошибка: файл существует на гибком
диске");
                }
                catch(NoDirectoryException e)
                {
```

```
                    System.out.println("Ошибка: указанной директории не
существует на гибком диске");
                }
                catch(NoPlaceException e)
                {
                    System.out.println("Ошибка: на гибком диске нет места");
                }
                catch(FileNotFoundException e)
                {
                    System.out.println("Ошибка: файл не найден");
                }
                catch(Exception e)
                {
                    System.out.println("Ошибка");
                    e.printStackTrace();
                }

            }
            else if(code==5)
            {
                String path, name;
                System.out.println("Введите путь к директории на гибком
диске:");
                path = scanner.nextLine();
                System.out.println("Введите имя папки:");
                name = scanner.nextLine();

                try
                {
                    storage.createDirectory(path, name);
                }
                catch(FileExistsException e)
                {
                    System.out.println("Ошибка: папка уже существует на
гибком диске");
                }
                catch(Exception e)
                {
                    System.out.println("Ошибка");
                    e.printStackTrace();
                }
            }
            else if(code==6)
            {
                System.out.println("Введите путь к файлу на гибком диске:");
                String path = scanner.nextLine();

                try
                {
                    storage.removeFile(path);
                }
                catch(FileNotFoundException e)
                {
                    System.out.println("Ошибка: файл не существует на гибком
диске");
                }
                catch(Exception e)
                {
                    System.out.println("Ошибка");
                    e.printStackTrace();
                }
            }
            else if(code==7)
            {
```

```
                System.out.println("Введите путь к папке на гибком диске:");
                String path = scanner.nextLine();

                try
                {
                    storage.removeDirectory(path);
                }
                catch(Exception e)
                {
                    System.out.println("Ошибка");
                    e.printStackTrace();
                }
            }
            else if(code==8)
                key=false;

            System.out.println("Нажмите Enter:");
            scanner.nextLine();
        }
    }
}
```

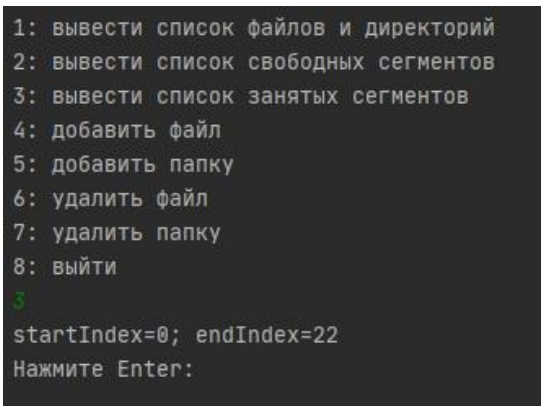
Вывод:

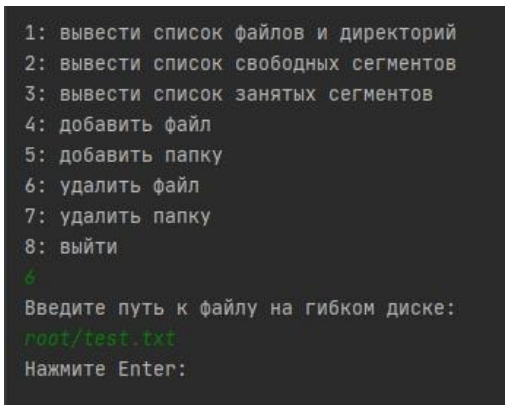
```
1: вывести список файлов и директорий
2: вывести список свободных сегментов
3: вывести список занятых сегментов
4: добавить файл
5: добавить папку
6: удалить файл
7: удалить папку
8: выйти
4
Введите путь к файлу:
T:/test.txt
Введите путь к файлу на гибком диске:
root
Нажмите Enter:
```

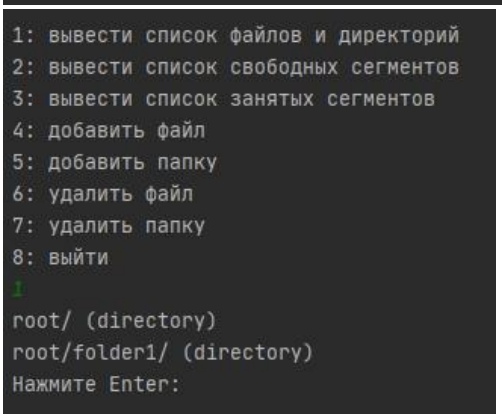
```
1: вывести список файлов и директорий
2: вывести список свободных сегментов
3: вывести список занятых сегментов
4: добавить файл
5: добавить папку
6: удалить файл
7: удалить папку
8: выйти
5
Введите путь к директории на гибком диске:
root
Введите имя папки:
folder1
Нажмите Enter:
```

```
1: вывести список файлов и директорий
2: вывести список свободных сегментов
3: вывести список занятых сегментов
4: добавить файл
5: добавить папку
6: удалить файл
7: удалить папку
8: выйти
1
root/ (directory)
root/test.txt; size=23; startIndex=0
root/folder1/ (directory)
Нажмите Enter:
```

```
1: вывести список файлов и директорий
2: вывести список свободных сегментов
3: вывести список занятых сегментов
4: добавить файл
5: добавить папку
6: удалить файл
7: удалить папку
8: выйти
2
startIndex=23; endIndex=1439999
Нажмите Enter:
```







Вывод: приобрел базовые навыки работы с классами в программах на языке программирования Java