

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ
по лабораторной работе №3

Выполнил:
студент 3 курса
группы ПО-8
Дорошков А.Д.

Проверил:
Крощенко А.А.

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

Задание 1

Реализовать простой класс.

Требования к выполнению

- Реализовать пользовательский класс по варианту.
- Создать другой класс с методом main, в котором будут находиться примеры использования пользовательского класса.

Для каждого класса

- Создать поля классов
 - Создать методы классов
 - Добавьте необходимые get и set методы (по необходимости)
 - Укажите соответствующие модификаторы видимости
 - Добавьте конструкторы
 - Переопределить методы toString() и equals()
- Входные данные:

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.

Выходные данные:

```
Elements: 1 2 3 0 0
Elements: 4 5 3 0 0
Elements: 1 2 3 0 0
Elements: 1 2 3 4 5 0 0 0 0 0
[ 1 2 3 4 0 0 0 0 0 0 ]
true
true
false
```

Краткий текст программы:

```
public boolean contain(int value){
    for(int element : elements){
```

```

        if (element == value){
            return true;
        }
    }
    return false;
}

public void add(int element){

    if(!this.contain(element)){
        for (int i = 0; i < this.capacity; i++) {
            if(elements[i] == 0){
                elements[i] = element;
                break;
            }
        }
    }
}

public void remove(int element){

    if(this.contain(element)){
        for (int i = 0; i < this.capacity; i++) {
            if(elements[i] == element){
                elements[i] = 0;
                break;
            }
        }
    }
}

static public Set union(Set firstSet, Set secondSet) {
    Set result = new Set(firstSet.getCapacity() + secondSet.getCapacity());

    for (int i = 0; i < firstSet.capacity; i++) {
        result.add(firstSet.elements[i]);
    }
    for (int i = 0; i < secondSet.capacity; i++) {
        result.add(secondSet.elements[i]);
    }
    return result;
}

public void printElements() {
    System.out.print("Elements: ");
    for (int i = 0; i < this.capacity; i++) {
        System.out.print(this.elements[i] + " ");
    }
    System.out.println();
}

```

```

    }

    public boolean equals(Set set) {
        if (this.capacity != set.getCapacity()){
            return false;
        } else {
            for (int element: elements){
                if(!set.contains(element)){
                    return false;
                }
            }
            return true;
        }
    }
}

@Override
public String toString() {
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.append("[ ");
    for (int element: elements){
        stringBuilder.append(element);
        stringBuilder.append(' ');
    }
    stringBuilder.append(']');
    return stringBuilder.toString();
}

```

Задание 2

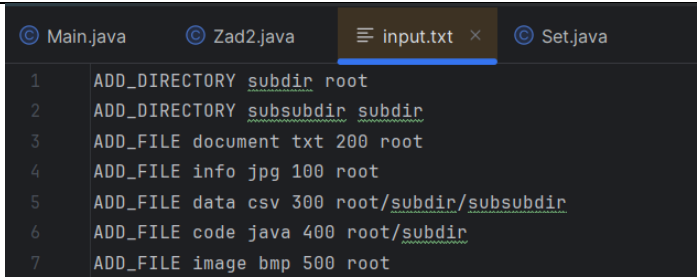
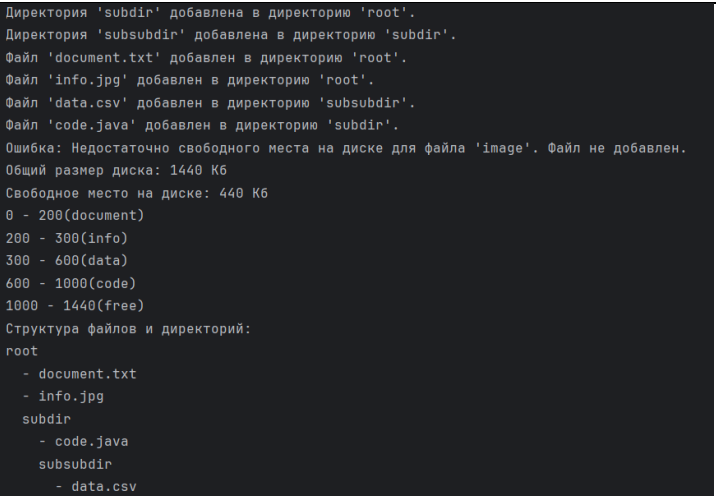
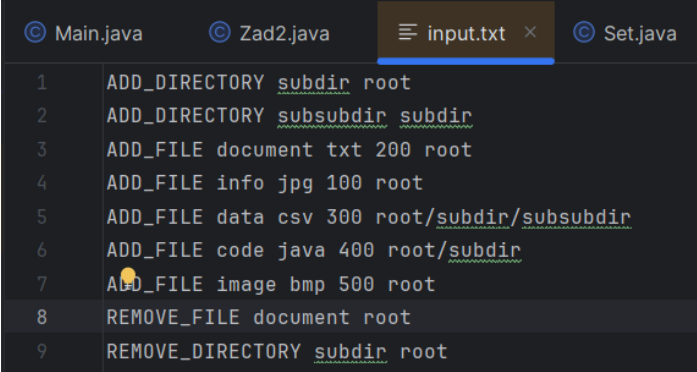
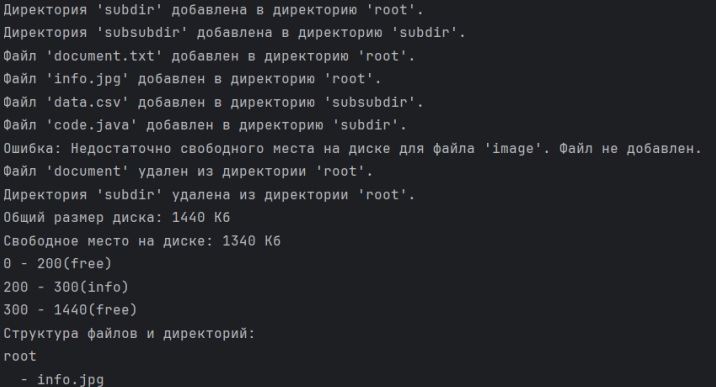
Моделирование файловой системы

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.

Результаты работы

Входные данные	Выходные данные
	
	

Текст основного класса:

```
public class Zad2 {
    public static void main(String[] args) {
        Disk disk = new Disk(1440);

        try {
            BufferedReader reader = new BufferedReader(new
            FileReader("C:\\Users\\Вадим\\IdeaProjects\\untitled2\\out\\input.txt")); {
                String line;
                while ((line = reader.readLine()) != null) {
                    String[] tokens = line.split("\\s+");

                    switch (tokens[0]) {
                        case "ADD_FILE":
                            File file = new File(tokens[1], tokens[2], Integer.parseInt(tokens[3]));
                            disk.addFile(file, tokens[4]);
                            break;

                        case "ADD_DIRECTORY":
                            Directory directory = new Directory(tokens[1]);
                            disk.addDirectory(directory, tokens[2]);
```

```

        break;

        case "REMOVE_FILE":
            disk.removeFile(tokens[1], tokens[2]);
            break;

        case "REMOVE_DIRECTORY":
            disk.removeDirectory(tokens[1], tokens[2]);
            break;
    }
}
} catch (IOException e) {
    e.printStackTrace();
}

disk.printDiskStatus();
}
}

```

Вывод: научился создавать и использовать классы в программах на языке программирования Java.