

Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

ОТЧЕТ

ЛАБОРАТОРНАЯ РАБОТА №3

Выполнил студент
группы АС-50:
Федорук Д. А.
Проверил:
Крощенко А. А.

Брест 2020

Вариант 9

Задание 1:

9) Множество вещественных чисел переменной мощности – Предусмотреть возможность пересечения 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. Реализацию множества осуществить на базе структуры ArrayList. Реализовать метод equals, выполняющий сравнение объектов данного типа.

```
package com.company;

import java.util.ArrayList;

class Set {
    private int power = 0;
    private ArrayList<Double> numbers;

    public Set() {
        numbers = new ArrayList<Double>();
    }

    public Double getElement(int index) {
        return numbers.get(index);
    }

    public int getPower() {
        return power;
    }

    public boolean checkExistElement(double element) {
        if(numbers.contains(element))
            return true;
        else
            return false;
    }

    public void addElement(double element) {
        numbers.add(element);
    }

    public void removeElement(double element) {
        numbers.remove(element);
    }

    public ArrayList intersection(Set set) {
        ArrayList<Double> addedNumbers = new ArrayList<>();

        for (double element: numbers)
        {
            if(set.numbers.contains(element))
                addedNumbers.add(element);
        }
        return addedNumbers;
    }
}
```

```

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

        for (double element: numbers
            ) {
            System.out.print(element+",");
        }
    }

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

        for (double element: numbers
            ) {
            if (!set.checkExistElement(element))
                return false;
        }
        return true;
    }

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

        str.append("Set: ");
        for (double element: numbers
            ) {
            str.append(element);
            str.append(",");
        }
        return str.toString();
    }
}

public class Main {

    public static void main(String[] args) {
        Set set1 = new Set();
        for(int i = 0; i < 5 ; i++){
            double random = Math.random()*100;
            set1.addElement(random);
        }
        set1.addElement(2.0);

        Set set2 = new Set();
        for(int i = 0; i < 5 ; i++){
            double random = Math.random()*100;
            set2.addElement(random);
        }
        set2.addElement(2.0);

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

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

        ArrayList<Double> result = set2.intersection(set1);
        System.out.print("Result intersection set2 <- set1:");

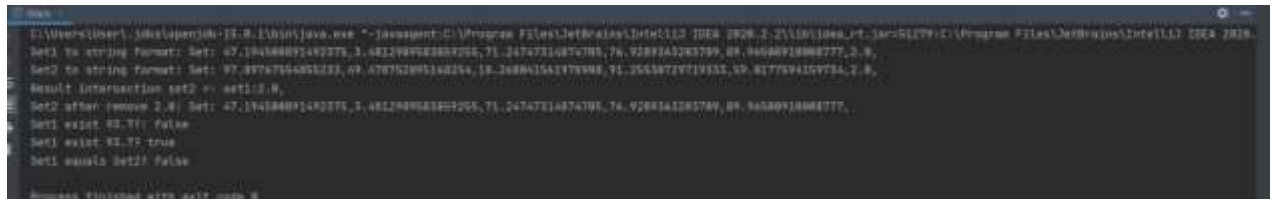
        for (double element: result
            ) {
            System.out.print(element+",");
        }
    }
}

```

```

set1.removeElement(2.0);
set2.removeElement(2.0);
System.out.print("\nSet2 after remove 2.0: ");
set1.print();
System.out.println("\nSet1 exist 93.7?: " +
set1.checkExistElement(93.7));
set1.addElement(93.7);
System.out.println("Set1 exist 93.7? " + set1.checkExistElement(93.7));
System.out.println("Set1 equals Set2? " + set1.equals(set2));
}
}

```



```

C:\Users\user\Idea\workspace>java -jar -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\jre\bin\idea_rt.jar=5279:C:\Program Files\JetBrains\IntelliJ IDEA\bin
Set1 to string format: Set: 47.18488891492275, 3.481298958385225, 71.24747314874705, 76.9289731283789, 89.9468891888777, 2.0
Set2 to string format: Set: 97.8976756485225, 89.678752895148254, 18.26884156197898, 71.2561872771953, 89.4177894159784, 2.0
Result: Intersection set2 <- set1: 2.0
Set2 after remove 2.0: Set: 47.18488891492275, 3.481298958385225, 71.24747314874705, 76.9289731283789, 89.9468891888777
Set1 exist 93.7?: false
Set1 exist 93.7?: true
Set1 equals Set2?: false
Process finished with exit code 0

```

Задание 2

9) Автоматизированная система склада

Написать программу для моделирования автоматизированного склада.

На складе хранится различная продукция (Product)

Каждая продукция характеризуется следующей информацией:

- id;
- Наименование;
- UPC (штрих-код);
- Производитель;
- Цена;
- Срок хранения;
- Количество.

Программа должна иметь следующий функционал:

- Генерация списка продукции на складе;
- Предоставлять список товаров для заданного наименования;
- Предоставлять список товаров для заданного наименования, цена которых не превосходит заданную;
- Предоставлять список товаров, срок хранения которых истек.

```

package com.company;

import java.util.ArrayList;
import java.util.Scanner;

```

```
class Product{
    private int id;
    private String name;
    private String ups;
    private String manufacturer;
    private int price;
    private int shelfLife;
    private int number;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getUps() {
        return ups;
    }

    public void setUps(String ups) {
        this.ups = ups;
    }

    public String getManufacturer() {
        return manufacturer;
    }

    public void setManufacturer(String manufacturer) {
        this.manufacturer = manufacturer;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public int getShelfLife() {
        return shelfLife;
    }

    public void setShelfLife(int shelfLife) {
        this.shelfLife = shelfLife;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }
}
```

```
}
```

```
public class Main {

    public static void main(String[] args) {
        //список названий

        ArrayList<String> name = new ArrayList<>();
        name.add("Redmi 7");
        name.add("P 40 Pro");
        name.add("C9");
        name.add("Iphone 8");
        name.add("One");
        name.add("Galaxy A31");

        ArrayList<String> ups = new ArrayList<>();
        ups.add("23084571");
        ups.add("24574835");
        ups.add("01344543");
        ups.add("93497821");
        ups.add("34854922");
        ups.add("09489684");

        ArrayList<String> manufacturers = new ArrayList<>();
        manufacturers.add("Xiaomi");
        manufacturers.add("Huawey");
        manufacturers.add("Meizu");
        manufacturers.add("Apple");
        manufacturers.add("Motorolla");
        manufacturers.add("Samsung");

        ArrayList<Integer> prices = new ArrayList<>();
        prices.add(300);
        prices.add(450);
        prices.add(279);
        prices.add(720);
        prices.add(540);
        prices.add(700);

        ArrayList<Integer> shelfLives = new ArrayList<>();
        shelfLives.add(30);
        shelfLives.add(30);
        shelfLives.add(30);
        shelfLives.add(30);
        shelfLives.add(30);
        shelfLives.add(30);

        ArrayList<Integer> numbers = new ArrayList<>();
        numbers.add(30);
        numbers.add(15);
        numbers.add(8);
        numbers.add(45);
        numbers.add(2);
        numbers.add(32);

        // write your code here
        Product product1 = new Product();
        Product product2 = new Product();
        Product product3 = new Product();
        Product product4 = new Product();
        Product product5 = new Product();
    }
}
```

```

Product product6 = new Product();

product1.setId(1);
product1.setName(name.get(0));
product1.setUps(ups.get(0));
product1.setManufacturer(manufacturers.get(0));
product1.setPrice(prices.get(0));
product1.setShelfLife(shelfLives.get(0));
product1.setNumber(numbers.get(0));

product2.setId(2);
product2.setName(name.get(1));
product2.setUps(ups.get(1));
product2.setManufacturer(manufacturers.get(1));
product2.setPrice(prices.get(1));
product2.setShelfLife(shelfLives.get(1));
product2.setNumber(numbers.get(1));

product3.setId(3);
product3.setName(name.get(2));
product3.setUps(ups.get(2));
product3.setManufacturer(manufacturers.get(2));
product3.setPrice(prices.get(2));
product3.setShelfLife(shelfLives.get(2));
product3.setNumber(numbers.get(2));

product4.setId(4);
product4.setName(name.get(3));
product4.setUps(ups.get(3));
product4.setManufacturer(manufacturers.get(3));
product4.setPrice(prices.get(3));
product4.setShelfLife(shelfLives.get(3));
product4.setNumber(numbers.get(3));

product5.setId(5);
product5.setName(name.get(4));
product5.setUps(ups.get(4));
product5.setManufacturer(manufacturers.get(4));
product5.setPrice(prices.get(4));
product5.setShelfLife(shelfLives.get(4));
product5.setNumber(numbers.get(4));

product6.setId(6);
product6.setName(name.get(5));
product6.setUps(ups.get(5));
product6.setManufacturer(manufacturers.get(5));
product6.setPrice(prices.get(5));
product6.setShelfLife(shelfLives.get(5));
product6.setNumber(numbers.get(5));

System.out.println("Добро пожаловать");
System.out.println("Введите 1 - чтобы просмотреть всю продукцию
на складе");
System.out.println("Введите 2 - чтобы задать наименование и
вывести соответствующий список товаров");
System.out.println("Введите 3 - чтобы задать наименование и
максимальную цену, и вывести список подходящих товаров");
System.out.println("Введите 4 - чтобы вывести список товаров,
которые залежались на складе");
System.out.println("Введите 5 - чтобы завершить работу");
Scanner scanner = new Scanner(System.in);
int num = scanner.nextInt();
switch (num) {

```

```

        case 1:
        {
            System.out.println("Id: "+product1.getId()+" , name:
"+product1.getName()+" , ups: "+product1.getUps()+" , manufacturer: "+
product1.getManufacturer()+" , price: "+product1.getPrice()+" , shelf life:
"+product1.getShelfLife()+" , number: "+product1.getNumber());
            System.out.println("Id: "+product2.getId()+" , name:
"+product2.getName()+" , ups: "+product2.getUps()+" , manufacturer: "+
product2.getManufacturer()+" , price: "+product2.getPrice()+" , shelf life:
"+product2.getShelfLife()+" , number: "+product2.getNumber());
            System.out.println("Id: "+product3.getId()+" , name:
"+product3.getName()+" , ups: "+product3.getUps()+" , manufacturer: "+
product3.getManufacturer()+" , price: "+product3.getPrice()+" , shelf life:
"+product3.getShelfLife()+" , number: "+product3.getNumber());
            System.out.println("Id: "+product4.getId()+" , name:
"+product4.getName()+" , ups: "+product4.getUps()+" , manufacturer: "+
product4.getManufacturer()+" , price: "+product4.getPrice()+" , shelf life:
"+product4.getShelfLife()+" , number: "+product4.getNumber());
            System.out.println("Id: "+product5.getId()+" , name:
"+product5.getName()+" , ups: "+product5.getUps()+" , manufacturer: "+
product5.getManufacturer()+" , price: "+product5.getPrice()+" , shelf life:
"+product5.getShelfLife()+" , number: "+product5.getNumber());
            System.out.println("Id: "+product6.getId()+" , name:
"+product6.getName()+" , ups: "+product6.getUps()+" , manufacturer: "+
product6.getManufacturer()+" , price: "+product6.getPrice()+" , shelf life:
"+product6.getShelfLife()+" , number: "+product6.getNumber());

            break;
        }
        case 2:{
            Scanner scanner2 = new Scanner(System.in);
            String specName = scanner2.nextLine();
            if(product1.getName().equals(specName))
                System.out.println("Id: "+product1.getId()+" , name:
"+product1.getName()+" , ups: "+product1.getUps()+" , manufacturer: "+
product1.getManufacturer()+" , price: "+product1.getPrice()+" , shelf life:
"+product1.getShelfLife()+" , number: "+product1.getNumber());
            if(product2.getName().equals(specName))
                System.out.println("Id: "+product2.getId()+" , name:
"+product2.getName()+" , ups: "+product2.getUps()+" , manufacturer: "+
product2.getManufacturer()+" , price: "+product2.getPrice()+" , shelf life:
"+product2.getShelfLife()+" , number: "+product2.getNumber());
            if(product3.getName().equals(specName))
                System.out.println("Id: "+product3.getId()+" , name:
"+product3.getName()+" , ups: "+product3.getUps()+" , manufacturer: "+
product3.getManufacturer()+" , price: "+product3.getPrice()+" , shelf life:
"+product3.getShelfLife()+" , number: "+product3.getNumber());
            if(product4.getName().equals(specName))
                System.out.println("Id: "+product4.getId()+" , name:
"+product4.getName()+" , ups: "+product4.getUps()+" , manufacturer: "+
product4.getManufacturer()+" , price: "+product4.getPrice()+" , shelf life:
"+product4.getShelfLife()+" , number: "+product4.getNumber());
            if(product5.getName().equals(specName))
                System.out.println("Id: "+product5.getId()+" , name:
"+product5.getName()+" , ups: "+product5.getUps()+" , manufacturer: "+
product5.getManufacturer()+" , price: "+product5.getPrice()+" , shelf life:
"+product5.getShelfLife()+" , number: "+product5.getNumber());
            if(product6.getName().equals(specName))
                System.out.println("Id: "+product6.getId()+" , name:
"+product6.getName()+" , ups: "+product6.getUps()+" , manufacturer: "+
product6.getManufacturer()+" , price: "+product6.getPrice()+" , shelf life:
"+product6.getShelfLife()+" , number: "+product6.getNumber());

            break;
        }
    }

```



```

        case 3:{
            Scanner scanner2 = new Scanner(System.in);
            String specName = scanner2.nextLine();
            Scanner scanner3 = new Scanner(System.in);
            int specPrice = scanner3.nextInt();

            if(product1.getName().equals(specName) && product1.getPrice() <= specPrice)
                System.out.println("Id: "+product1.getId()+", name: "+product1.getName()+", ups: "+product1.getUps()+", manufacturer: "+product1.getManufacturer()+", price: "+product1.getPrice()+", shelf life: "+product1.getShelfLife()+", number: "+product1.getNumber());

            if(product2.getName().equals(specName) && product2.getPrice() <= specPrice)
                System.out.println("Id: "+product2.getId()+", name: "+product2.getName()+", ups: "+product2.getUps()+", manufacturer: "+product2.getManufacturer()+", price: "+product2.getPrice()+", shelf life: "+product2.getShelfLife()+", number: "+product2.getNumber());

            if(product3.getName().equals(specName) && product3.getPrice() <= specPrice)
                System.out.println("Id: "+product3.getId()+", name: "+product3.getName()+", ups: "+product3.getUps()+", manufacturer: "+product3.getManufacturer()+", price: "+product3.getPrice()+", shelf life: "+product3.getShelfLife()+", number: "+product3.getNumber());

            if(product4.getName().equals(specName) && product4.getPrice() <= specPrice)
                System.out.println("Id: "+product4.getId()+", name: "+product4.getName()+", ups: "+product4.getUps()+", manufacturer: "+product4.getManufacturer()+", price: "+product4.getPrice()+", shelf life: "+product4.getShelfLife()+", number: "+product4.getNumber());

            if(product5.getName().equals(specName) && product5.getPrice() <= specPrice)
                System.out.println("Id: "+product5.getId()+", name: "+product5.getName()+", ups: "+product5.getUps()+", manufacturer: "+product5.getManufacturer()+", price: "+product5.getPrice()+", shelf life: "+product5.getShelfLife()+", number: "+product5.getNumber());

            if(product6.getName().equals(specName) && product6.getPrice() <= specPrice)
                System.out.println("Id: "+product6.getId()+", name: "+product6.getName()+", ups: "+product6.getUps()+", manufacturer: "+product6.getManufacturer()+", price: "+product6.getPrice()+", shelf life: "+product6.getShelfLife()+", number: "+product6.getNumber());
                break;
        }
        case 4:{
            if(product1.getShelfLife() > 45)
                System.out.println("Id: "+product1.getId()+", name: "+product1.getName()+", ups: "+product1.getUps()+", manufacturer: "+product1.getManufacturer()+", price: "+product1.getPrice()+", shelf life: "+product1.getShelfLife()+", number: "+product1.getNumber());
            if(product2.getShelfLife() > 45)
                System.out.println("Id: "+product2.getId()+", name: "+product2.getName()+", ups: "+product2.getUps()+", manufacturer: "+product2.getManufacturer()+", price: "+product2.getPrice()+", shelf life: "+product2.getShelfLife()+", number: "+product2.getNumber());
            if(product3.getShelfLife() > 45)
                System.out.println("Id: "+product3.getId()+", name: "+product3.getName()+", ups: "+product3.getUps()+", manufacturer: "+product3.getManufacturer()+", price: "+product3.getPrice()+", shelf life: "+product3.getShelfLife()+", number: "+product3.getNumber());
            if(product4.getShelfLife() > 45)
                System.out.println("Id: "+product4.getId()+", name: "+product4.getName()+", ups: "+product4.getUps()+", manufacturer: "+product4.getManufacturer()+", price: "+product4.getPrice()+", shelf life: "+product4.getShelfLife()+", number: "+product4.getNumber());

```

```

        if(product5.getShelfLife()>45)
            System.out.println("Id: "+product5.getId()+", name:
"+product5.getName()+", ups: "+product5.getUps()+", manufacturer: "+
product5.getManufacturer()+", price: "+product5.getPrice()+", shelf life:
"+product5.getShelfLife()+", number: "+product5.getNumber());
        if(product6.getShelfLife()>45)
            System.out.println("Id: "+product6.getId()+", name:
"+product6.getName()+", ups: "+product6.getUps()+", manufacturer: "+
product6.getManufacturer()+", price: "+product6.getPrice()+", shelf life:
"+product6.getShelfLife()+", number: "+product6.getNumber());
        break;
    }
    case 5:{
        break;
    }
}
}
}

```

```

C:\Users\user\IdeaProjects\IT-8\T1\src\java.exe -jar %workspace%\Program Files\JetBrains\IntelliJ IDEA 2020.2\bin\idea_rt.jar -S6494712\Program Files\
Идеи\workspace
Задача: написать
Задача 1 - создать программу, которая выводит на экран
Задача 2 - создать классы, соответствующие описанию задачи
Задача 3 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 4 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 5 - создать классы, соответствующие описанию задачи

Id: 1, name: Beko T, ups: 138A071, manufacturer: Beko, price: 180, shelf life: 18, number: 10
Id: 2, name: P od Piv, ups: 24V1000, manufacturer: Beko, price: 250, shelf life: 18, number: 15
Id: 3, name: D, ups: 1314100, manufacturer: Beko, price: 270, shelf life: 18, number: 8
Id: 4, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 45
Id: 5, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 8
Id: 6, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 10

Process finished with exit code 0

```

```

C:\Users\user\IdeaProjects\IT-8\T1\src\java.exe -jar %workspace%\Program Files\JetBrains\IntelliJ IDEA 2020.2\bin\idea_rt.jar -S6494712\Program Fi
Задача: написать
Задача 1 - создать программу, которая выводит на экран
Задача 2 - создать классы, соответствующие описанию задачи
Задача 3 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 4 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 5 - создать классы, соответствующие описанию задачи

Id: 1, name: Beko T, ups: 138A071, manufacturer: Beko, price: 180, shelf life: 18, number: 10
Id: 2, name: P od Piv, ups: 24V1000, manufacturer: Beko, price: 250, shelf life: 18, number: 15
Id: 3, name: D, ups: 1314100, manufacturer: Beko, price: 270, shelf life: 18, number: 8
Id: 4, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 45
Id: 5, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 8
Id: 6, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 10

Process finished with exit code 0

```

```

C:\Users\user\IdeaProjects\IT-8\T1\src\java.exe -jar %workspace%\Program Files\JetBrains\IntelliJ IDEA 2020.2\bin\idea_rt.jar -S6494712\Program Fi
Задача: написать
Задача 1 - создать программу, которая выводит на экран
Задача 2 - создать классы, соответствующие описанию задачи
Задача 3 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 4 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 5 - создать классы, соответствующие описанию задачи

Id: 1, name: Beko T, ups: 138A071, manufacturer: Beko, price: 180, shelf life: 18, number: 10
Id: 2, name: P od Piv, ups: 24V1000, manufacturer: Beko, price: 250, shelf life: 18, number: 15
Id: 3, name: D, ups: 1314100, manufacturer: Beko, price: 270, shelf life: 18, number: 8
Id: 4, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 45
Id: 5, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 8
Id: 6, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 10

Process finished with exit code 0

```

```

C:\Users\user\IdeaProjects\IT-8\T1\src\java.exe -jar %workspace%\Program Files\JetBrains\IntelliJ IDEA 2020.2\bin\idea_rt.jar -S6494712\Program Fi
Задача: написать
Задача 1 - создать программу, которая выводит на экран
Задача 2 - создать классы, соответствующие описанию задачи
Задача 3 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 4 - создать классы, соответствующие описанию задачи, и вывести список товаров
Задача 5 - создать классы, соответствующие описанию задачи

Id: 1, name: Beko T, ups: 138A071, manufacturer: Beko, price: 180, shelf life: 18, number: 10
Id: 2, name: P od Piv, ups: 24V1000, manufacturer: Beko, price: 250, shelf life: 18, number: 15
Id: 3, name: D, ups: 1314100, manufacturer: Beko, price: 270, shelf life: 18, number: 8
Id: 4, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 45
Id: 5, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 8
Id: 6, name: Beko S, ups: 1314100, manufacturer: Beko, price: 180, shelf life: 18, number: 10

Process finished with exit code 0

```

```
C:\Users\user1>.\bin\server6-25.8.1\bin\java.exe ^-javaagent:C:\Program Files\JetBrains\IntelliJ IDE 2020.2\bin\idea_rt.jar=54697:C:\Program Files\JetBrains\IntelliJ IDE 2020.2\bin\java.exe^
Добро пожаловать
Введите 1 - чтобы посмотреть все процедуры на складе
Введите 2 - чтобы задать наименование и единицы существующей строки товара
Введите 3 - чтобы задать наименование и координатные цены, и единицы строки складских товаров
Введите 4 - чтобы вывести список товаров, которые затребованы на складе
Введите 5 - чтобы завершить работу

Process finished with exit code 0
|
```