# Министерство образования Республики Беларусь Учреждение образования

«Брестский государственный технический университет» Кафедра ИИТ

# Лабораторная работа №3

Выполнила: Студент 3 курса Группы ПО-5 Соколов А.А. Проверил:

Крощенко А.А.

Брест 2021

**Вариант - 9**

**Цель работы:** 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 находится 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()

9) Множество вещественных чисел переменной мощности – Предусмотреть возможность пересечения двух множеств, вывода на печать элементов множества, а также 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.

**Код программы:  
Start - Файл**File **package** kkk;

**public** **class** Start {

**public** **static** **void** main(String[] args){

Many\_numbers a1=**new** Many\_numbers();

Many\_numbers a2=**new** Many\_numbers(2.798f);

Many\_numbers a3=**new** Many\_numbers(5.7f);

System.***out***.print("Выводим первый - ");

System.***out***.println(a1.getNumbers());

System.***out***.print("Выводим второй - ");

System.***out***.println(a2.getNumbers());

System.***out***.println("Добавляем ко второму число 5.7");

a2.setNumbers(5.7f);

System.***out***.print("Выводим второй - ");

System.***out***.println(a2.getNumbers());

System.***out***.println(Many\_numbers.*intersection*(a1,a2));

System.***out***.println("Удаление по индексу(1) в первом: " + a1.getNumbers());

a1.remove(1);

System.***out***.print("Выводим первый - ");

System.***out***.println(a1.getNumbers());

System.***out***.println("Удаление 3.87 в первом, а также добавляем 5.7 в конец ");

a1.remove(3.87f);

a1.setNumbers(5.7f);

System.***out***.print("Выводим первый - ");

System.***out***.println(a1.getNumbers());

System.***out***.println(Many\_numbers.*intersection*(a1,a2));

System.***out***.print(a1.getNumbers() + " == " +a2.getNumbers() +" ");

System.***out***.println(a1.equals(a2));

System.***out***.print(a1.getNumbers() + " == " +a3.getNumbers() +" ");

System.***out***.println(a1.equals(a3));

System.***out***.println(a1);

}

}

**Many\_Numbers - Файл**

**package** kkk;

**import** java.util.ArrayList;

**public** **class** Many\_numbers {

**private** ArrayList<Float> Numbers=**new** ArrayList<>();

**public** **void** setNumbers(**float** a) { //Добавление числа в конец массива

**this**.Numbers.add(a);

}

**public** **void** setNumbers(**float** i, **float** j, **float** k) {

**this**.Numbers.add(i);

**this**.Numbers.add(j);

**this**.Numbers.add(k);

}

Many\_numbers(){//Конструктор без параметров

**this**.setNumbers(1.56f,2.798f,3.87f);

}

Many\_numbers(**float** a){ //Конструктор с параметром

**this**.setNumbers(a);

}

**public** ArrayList<Float> getNumbers() { //Просмотр всех элементов в массиве

**return** Numbers;

}

**public** **void** remove(**int** a){ //Удаление по номеру в массиве

a--;

**if**(Numbers.size()>a && a>=0) {

**float** b=Numbers.get(a);

Numbers.remove(b);

}

}

**public** **void** remove(**float** a){ //Удаление по значению

**if**(**this**.Numbers.contains((**float**)a)) {

**int** index = Numbers.indexOf((**float**)a);

Numbers.remove(index);

}

}

**public** **static** ArrayList<Float> intersection(Many\_numbers list1, Many\_numbers list2){ //Пересечение двух множеств

ArrayList<Float> list=**new** ArrayList<Float>();

**for** (**int** i=0;i<list1.Numbers.size();i++) {

**for**(**int** z=0;z<list2.Numbers.size();z++) {

**if**(list1.Numbers.get(i).equals(list2.Numbers.get(z))) {

list.add(list1.Numbers.get(i));

}

}

}

System.***out***.print("Пересечение двух множеств: A:"+ list1.getNumbers() + " и B:" + list2.getNumbers() +" = ");

**return** list;

}

@Override

**public** **boolean** equals(Object o) {

**if** (**this** == o)

**return** **true**;

**if** (o == **null** || **this**.getClass() != o.getClass())

**return** **false**;

Many\_numbers that = (Many\_numbers) o;

**if**(Numbers.size() != that.Numbers.size())

**return** **false**;

**int** i=0;

**for** (; i < Numbers.size(); i++) {

**if**(**this**.Numbers.get(i) != that.Numbers.get(i))

**return** **false**;

}

**if**(i == that.Numbers.size())

**return** **true**;

**return** **false**;

}

@Override

**public** String toString() {

**return** "Numbers: "+**this**.getNumbers() ;

}

}

## Результат:

## Задание 2

Разработать автоматизированную систему на основе некоторой структуры дан- ных, манипулирующей объектами пользовательского класса. Реализовать требуемые функции обработки данных Требования к выполнению

* Задание посвящено написанию классов, решающих определенную задачу автоматизации;
* Данные для программы загружаются из файла (формат произволен). Файл создать и написать вручную

## 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.

**Код программы:**

## Start - Файл

**package** kkk;

**public** **class** Start {

**public** **static** **void** main(String[] args){

Product a1=**new** Product();

a1.Download("Product1.txt");

System.***out***.print(a1);

System.***out***.println("Ищем в списке товаров <Фисташки>");

a1.Search\_Name("Фисташки");

System.***out***.println("Ищем в списке товаров <Вафли>");

a1.Search\_Name("Вафли");

System.***out***.println("Поиск товаров ниже указанной стоимости(3 руб)");

a1.Search\_price\_down(3f);

System.***out***.println("Проверка товаров которых срок годности закончился. ");

a1.Data();

}

}

**Product - Файл**

**package** kkk;

**import** java.io.File;

**import** java.io.IOException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Date;

**import** java.util.Scanner;

**public** **class** Product {

**private**

ArrayList<Integer> Id=**new** ArrayList<>(10); //Id

ArrayList<Long> UPC=**new** ArrayList<>(10); //Штрих-код

ArrayList<String> Name=**new** ArrayList<>(10); //Название товара

ArrayList<Float> Price=**new** ArrayList<>(10); //Цена

ArrayList<String> Manufactures=**new** ArrayList<>(10); //Производитель

ArrayList<Integer> Amount=**new** ArrayList<>(10); //Количество

ArrayList<String> Date=**new** ArrayList<>(10); //Дата истечение срока хранения

// SimpleDateFormat format = new SimpleDateFormat("dd.MM.yyyy");

**public**

Product() {

}

**void** Download(String directory){ //Чтение файла с данными о продуктах

**try** {

File file\_first=**new** File(directory);

Scanner scanner= **new** Scanner(file\_first, "UTF8");

**while**(scanner.hasNextLine()) {

String line=scanner.nextLine();

String[] mass\_first =line.split(" ");

**this**.Id.add(Integer.*parseInt*(mass\_first[0].trim()));

**this**.UPC.add(Long.*parseLong*(mass\_first[1]));

**this**.Name.add(mass\_first[2]);

**this**.Price.add(Float.*valueOf*(mass\_first[3]));

**this**.Manufactures.add(mass\_first[4]);

**this**.Amount.add(Integer.*parseInt*(mass\_first[5].trim()));

**this**.Date.add(mass\_first[6]);

}

scanner.close();

}**catch**(IOException e){

System.***out***.println("Error: " +e);

}

}

**void** Search\_Name(String name) { //Поиск по товару

**int** a=0;

**for**(**int** i=0;i<**this**.Id.size();i++) {

**if**(name.equals(**this**.Name.get(i))) {

**this**.table();

**this**.vivod(i);

a++;

}

}

**if**(a==0) {

System.***out***.println("Такого товара ("+name+") не существует");

}

}

**void** Search\_price\_down(**float** price) { //Поиск товаров ниже указанной цены

**int** a=0;

**for**(**int** i=0;i<**this**.Id.size();i++) {

**if**(price>=**this**.Price.get(i)) {

**if**(a==0) {

**this**.table();

}

**this**.vivod(i);

a++;

}

}

**if**(a==0) {

System.***out***.println("Таких товаров ниже указанной стоимости ("+price+") не существует");

}

}

**void** Data() { //Проверка товаров у которых закончился срок годности

SimpleDateFormat format = **new** SimpleDateFormat("dd.MM.yyyy");

Date dateOne= **new** Date();

System.***out***.printf("%1$s %2$td %2$tB %2$tY\n", "Сегодняшняя дата:", dateOne);

Date dateTwo = **null**;

**int** a=0;

**for**(**int** i=0;i<**this**.Id.size();i++) {

**try** {

dateTwo = format.parse(**this**.Date.get(i));

} **catch** (Exception e) {

e.printStackTrace();

}

**if**(dateOne.getTime()>=dateTwo.getTime()) {

**if**(a==0) {

**this**.table();

}

**this**.vivod(i);

a++;

}

}

}

@Override

**public** String toString() {

String f="";

**this**.table();

**for**(**int** i=0;i<**this**.Id.size();i++) {

**this**.vivod(i);

}

**return** f;

}

**void** table() {

System.***out***.printf("%-5s%-15s%-13s%-7s%-22s%-12s%-29s%n","ID","Штрих-код","Наименование","Цена","Производитель","Количество","Дата истечение срока хранения");

}

**void** vivod(**int** i){

System.***out***.printf("%-5s%-15s%-13s%-7s%-22s%-12s%-29s%n",**this**.Id.get(i),**this**.UPC.get(i),**this**.Name.get(i),**this**.Price.get(i),**this**.Manufactures.get(i),**this**.Amount.get(i),**this**.Date.get(i));

}

## }

## Результат:

**Вывод:** в ходе работы научилась создавать и использовать классы в программах на языке программирования Java.