

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

ОТЧЁТ
по лабораторной работе №3

Выполнил:
студент 3 курса
группы ПО-9
Тусюк Т.В.

Проверил:
Крощенко А.А.

Брест 2024

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 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()

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 позволять создавать объекты с начальной инициализацией. Реализацию множества осуществить на базе структуры **ArrayList**. Реализовать метод equals, выполняющий сравнение объектов данного типа.

Выполнение задания:

```
import java.util.ArrayList;

public class Task01 {
    private ArrayList<Double> numbers;

    public Task01() {
        this.numbers = new ArrayList<>();
    }

    public Task01(ArrayList<Double> numbers) {
        this.numbers = new ArrayList<>(numbers);
    }

    public void add(double number) {
        if (!numbers.contains(number)) {
            numbers.add(number);
        }
    }

    public void remove(double number) {
        numbers.remove(number);
    }
}
```

```

public boolean contains(double number) {
    return numbers.contains(number);
}

public Task01 intersection(Task01 otherSet) {
    ArrayList<Double> intersectionList = new ArrayList<>();
    for (Double num : numbers) {
        if (otherSet.contains(num)) {
            intersectionList.add(num);
        }
    }
    return new Task01(intersectionList);
}

public void print() {
    System.out.println(numbers);
}

@Override
public String toString() {
    return numbers.toString();
}

@Override
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (obj == null || getClass() != obj.getClass()) return false;
    Task01 that = (Task01) obj;
    return numbers.equals(that.numbers);
}

public static void main(String[] args) {
    Task01 set1 = new Task01();
    set1.add(1.5);
    set1.add(2.5);
    set1.add(3.5);
    System.out.println("Set 1: " + set1);

    Task01 set2 = new Task01();
    set2.add(2.5);
    set2.add(3.5);
    set2.add(4.5);
    System.out.println("Set 2: " + set2);

    Task01 intersectionSet = set1.intersection(set2);
    System.out.println("Intersection: " + intersectionSet);

    System.out.println("Does set 1 equal set 2? " + set1.equals(set2));
    System.out.println("Does set 1 equal itself? " + set1.equals(set1));
}
}

```

Результат:

```
Set 1: [1.5, 2.5, 3.5]
Set 2: [2.5, 3.5, 4.5]
Intersection: [2.5, 3.5]
Does set 1 equal set 2? false
Does set 1 equal itself? true
```

Задание 2:

Разработать автоматизированную систему на основе некоторой структуры данных, 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;
- URC (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.

Выполнение задания:

Warehouse.java

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class Warehouse {
    private List<Product> products;

    public Warehouse() {
        products = new ArrayList<>();
    }

    public void loadProductsFromFile(String filename) {
        try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                int id = Integer.parseInt(parts[0]);
                String name = parts[1];
                String upc = parts[2];
                String manufacturer = parts[3];
                double price = Double.parseDouble(parts[4]);
                int shelfLife = Integer.parseInt(parts[5]);
                int quantity = Integer.parseInt(parts[6]);
                Product product = new Product(id, name, upc, manufacturer, price,
shelfLife, quantity);
                products.add(product);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public List<Product> generateProductList() {
        return new ArrayList<>(products);
    }

    public List<Product> findProductsByName(String name) {
        List<Product> result = new ArrayList<>();
        for (Product product : products) {
            if (product.getName().equalsIgnoreCase(name)) {
                result.add(product);
            }
        }
        return result;
    }

    public List<Product> findProductsByNameAndPrice(String name, double maxPrice) {
        List<Product> result = new ArrayList<>();
        for (Product product : products) {
            if (product.getName().equalsIgnoreCase(name) && product.getPrice() <=
maxPrice) {
                result.add(product);
            }
        }
        return result;
    }
}
```

```

    }

    public List<Product> findExpiredProducts() {
        List<Product> result = new ArrayList<>();
        for (Product product : products) {
            if (product.getShelfLife() == 0) {
                result.add(product);
            }
        }
        return result;
    }
}

```

Product.java

```

public class Product {
    private int id;
    private String name;
    private String upc;
    private String manufacturer;
    private double price;
    private int shelfLife;
    private int quantity;

    public Product(int id, String name, String upc, String manufacturer, double
price, int shelfLife, int quantity) {
        this.id = id;
        this.name = name;
        this.upc = upc;
        this.manufacturer = manufacturer;
        this.price = price;
        this.shelfLife = shelfLife;
        this.quantity = quantity;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getUpc() {
        return upc;
    }

    public String getManufacturer() {
        return manufacturer;
    }

    public double getPrice() {
        return price;
    }

    public int getShelfLife() {
        return shelfLife;
    }

    public int getQuantity() {
        return quantity;
    }
}

```

Main.java

```
import java.util.List;
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Warehouse warehouse = new Warehouse();
        warehouse.loadProductsFromFile("F:\\Java\\SPP_Lab3\\src\\products.txt");

        List<Product> productList = warehouse.generateProductList();
        System.out.println("\tСписок всех продуктов");
        for (Product product : productList) {
            System.out.println(product.getName() +
                               ":" + product.getManufacturer());
        }

        Scanner scanner = new Scanner(System.in);

        System.out.println("\n\tВведите название товара для поиска:");
        String productName = scanner.nextLine();

        System.out.println("Введите максимальную цену:");
        double maxPrice = scanner.nextDouble();

        System.out.println("\nПродукты с наименованием '" + productName + "':");
        List<Product> namedProducts = warehouse.findProductsByName(productName);
        for (Product product : namedProducts) {
            System.out.println(product.getName() + " - " + product.getManufacturer()
                               + " - " + product.getQuantity() + " шт.");
        }

        System.out.println("\nПродукты с наименованием '" + productName + "' и ценой не более " + maxPrice + ":");
        List<Product> cheapNamedProducts =
warehouse.findProductsByNameAndPrice(productName, maxPrice);
        for (Product product : cheapNamedProducts) {
            System.out.println(product.getName() + " - " +
product.getManufacturer() + " - " + product.getPrice());
        }

        System.out.println("\nИстекший срок хранения продуктов:");
        List<Product> expiredProducts = warehouse.findExpiredProducts();
        for (Product product : expiredProducts) {
            System.out.println(product.getName() + "-" + product.getManufacturer());
        }
    }
}
```

Результат:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:D:\
    Список всех продуктов
Молоко:Бабушкина крынка
Хлеб:Хлебный комбинат "ХлебоХлеб"
Яблоки:Фермерское хозяйство "Зеленая поляна"
Мясо:Мясокомбинат "Красная говядина"
Макароны:Фабрика пасты "Итальянский вкус"
Сок:Соковая фабрика "Свежий напиток"
Картофель:Агрофирма "Золотой урожай"
Печенье:Кондитерская "Сладкий мир"
Колбаса:Мясоперерабатывающий комбинат "Деликатес"
Сыр:Молочная ферма "Сырная радость"
Молоко:Остромечьево
Молоко:Савушкин продукт

    Введите название товара для поиска:
Молоко
Введите максимальную цену:
3

Продукты с наименованием 'Молоко':
Молоко - Бабушкина крынка-100 шт.
Молоко - Остромечьево-50 шт.
Молоко - Савушкин продукт-20 шт.

Продукты с наименованием 'Молоко' и ценой не более 3.0:
Молоко - Бабушкина крынка - 2.5
Молоко - Остромечьево - 2.0
Молоко - Савушкин продукт - 1.5

Истекший срок хранения продуктов:
Колбаса-Мясоперерабатывающий комбинат "Деликатес"
Сыр-Молочная ферма "Сырная радость"
Молоко-Савушкин продукт
```

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