

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №5-1
по курсу: «Языки и методы программирования»
«Монады в языке Java»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Приобретение навыков использования монад Optional и Stream в программах на языке Java.

2. Персональный вариант

Таблица, отображающая названия товаров в их цены и количества имеющихся на складе единиц товара, с операциями:

1. порождение потока названий товаров, стоимость запасов которых превышает указанную сумму денег;
2. поиск товара, количество единиц которого на складе превышает суммарное количество единиц всех остальных товаров.

Проверить работу первой операции нужно путём группировки названий товаров по первой букве названия.

3. Решение

3.1. Код

```
import java.util.*;
import java.util.stream.Stream;

class Product {
    Integer cost, count;
    String name;
    Product (String name, int count, int cost) {
        this.name = name;
        this.count = count;
        this.cost = cost;
    }

    @Override
    public String toString() {
        return "Product: " + name + " (cost=" + cost.toString()
            + ", count=" + count.toString() + ")";
    }
}

class ProductTable {
    HashMap<String, Product> Table;
    int total;
    ProductTable() {
        Table = new HashMap<>();
        total = 0;
    }
}
```

```

    }
    void add(Product p) {
        Table.put(p.name, p);
        total += p.count;
    }

    void add(String name, int cost, int count) {
        this.add(new Product(name, count, cost));
    }

    public Stream<String> nameStream(int v) {
        ArrayList<String> result = new ArrayList<>();

        Table.entrySet().stream()
            .filter(x -> x.getValue().count *
                ↪ x.getValue().cost > v)
            .forEach(x -> result.add(x.getKey()));

        return result.stream();
    }

    public Optional<Product> getProduct() {
        Optional<Product> result = Optional.empty();

        Optional<Map.Entry<String, Product>> tmp =
            ↪ Table.entrySet()
                .stream()
                .filter(x -> x.getValue().count * 2 > total)
                .findFirst();

        if (tmp.isPresent()) {
            result = Optional.ofNullable(tmp.get().getValue());
        }
        return result;
    }
}

class NameComparator implements Comparator<String> {
    public int compare(String a, String b) {
        return a.charAt(0) - b.charAt(0);
    }
}

public class Test {
    public static void main(String[] args) {

```

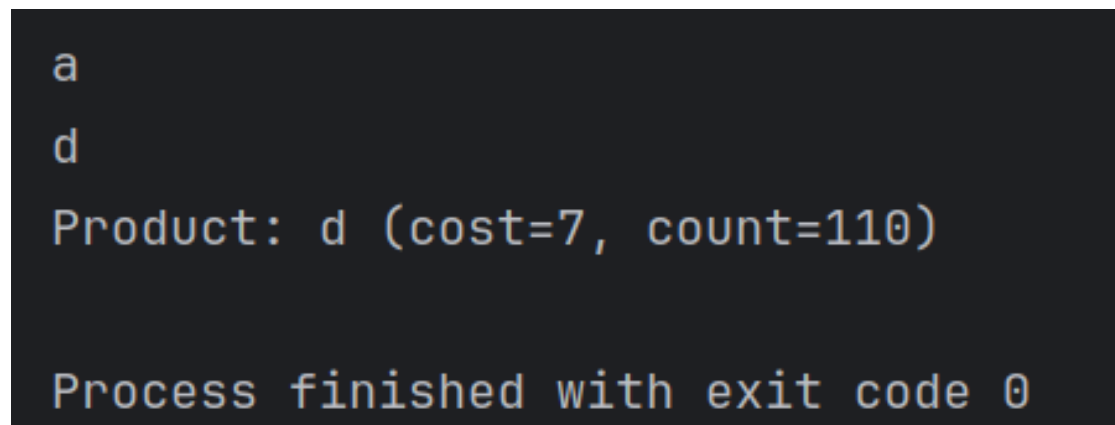
```

        ProductTable t = new ProductTable();
        t.add("a", 10, 10);
        t.add("b", 6, 10);
        t.add("c", 2, 5);
        t.add("d", 7, 110);
        t.nameStream(90).sorted(new
            ↪ NameComparator()).forEach(System.out::println);
        System.out.println(t.getProduct().get());
    }
}

```

Код 1: Test.java

3.2. Скриншоты



```

a
d
Product: d (cost=7, count=110)
Process finished with exit code 0

```

Рис. 1: Пример работы программы