**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Дисциплина «Информационные технологии и программирование»

Лабораторная работа №6

Коллекции в Java

Выполнил:

Студент группы БВТ2303

Гришко Дмитрий

**Цель работы:**

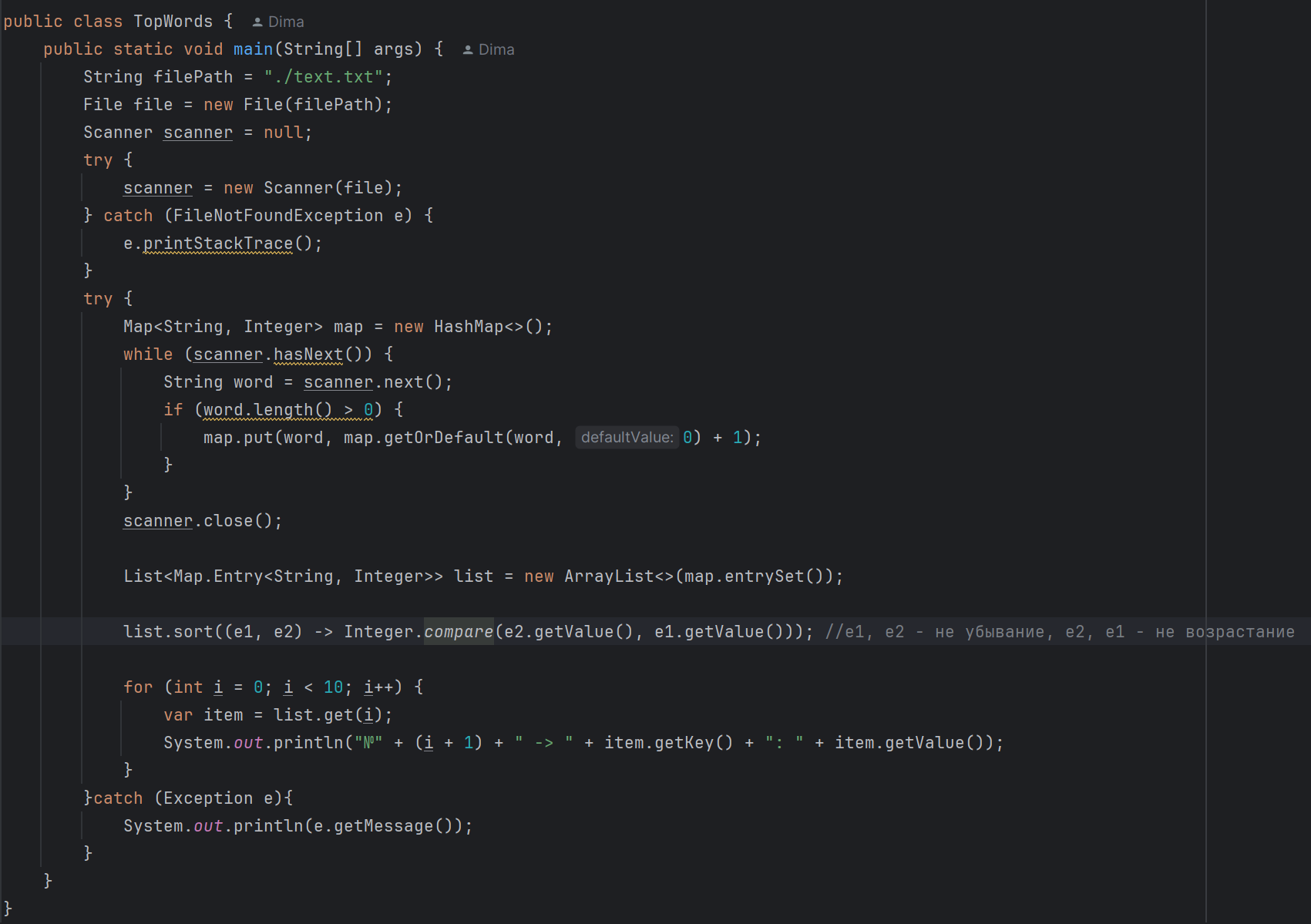
Научиться работать с коллекциями в Java.

**Задание:**

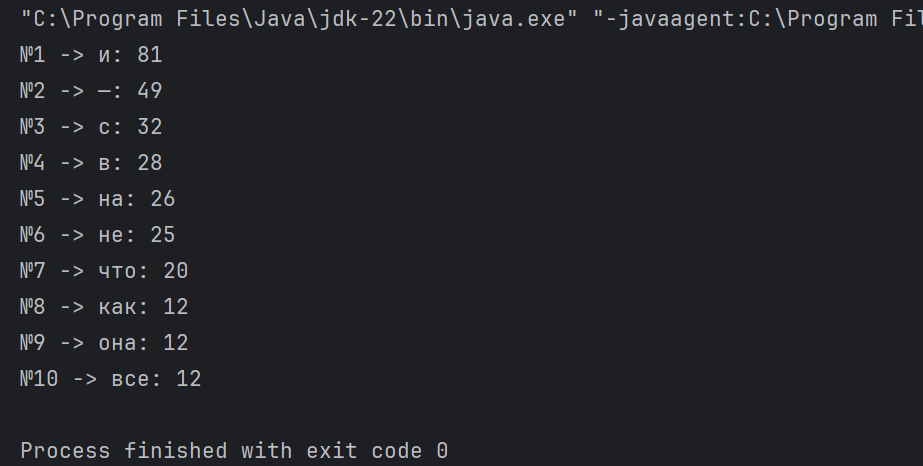
1. Написать программу, которая считывает текстовый файл и выводит на экран топ-10 самых часто встречающихся слов в этом файле. Для решения задачи использовать коллекцию Map, где ключом будет слово, а значением - количество его повторений в файле.
2. Написать обобщенный класс Stack, который реализует стек на основе массива. Класс должен иметь методы push для добавления элемента в стек, pop для удаления элемента из стека и peek для получения верхнего элемента стека без его удаления.
3. Необходимо разработать программу для учета продаж в магазине. Программа должна позволять добавлять проданные товары в коллекцию, выводить список проданных товаров, а также считать общую сумму продаж и наиболее популярный товар. Использовать ArrayList для хранения списка проданных товаров.

**Ход работы:**

1. Я написал программу, которая считывает файл и выводит 10 самых часто встречаемых слов в этом файле.



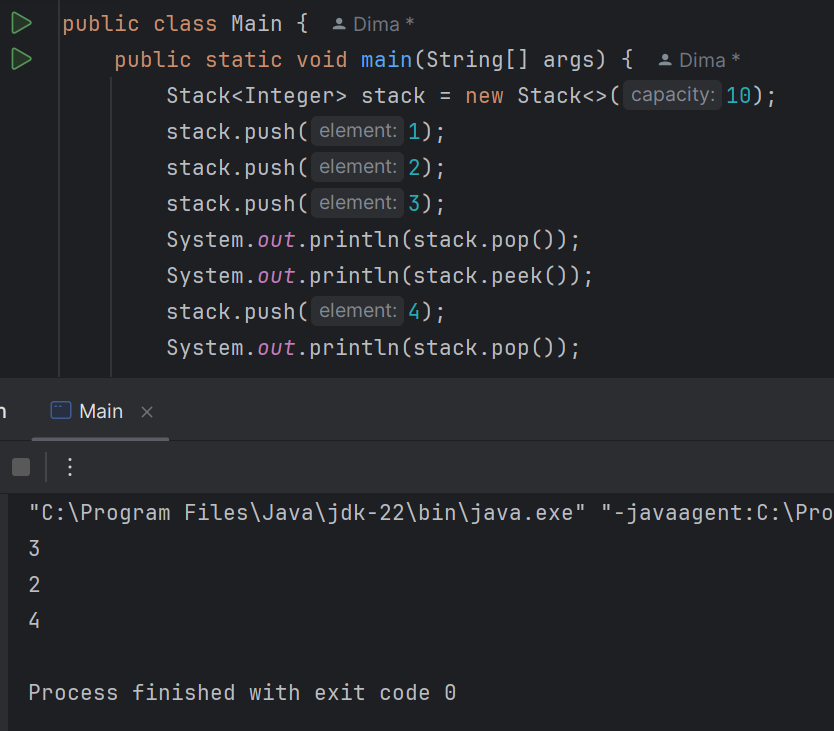
Для этого я создал хэш-таблицу из коллекции map, в которой ключ – слово из текста, а значение – частота его появления в тексте. Пока в сканнер может считывать слово, будем заполнять нашу хэш-таблицу. После ее заполнения создадим массив ArrayList, содержащий пары ключ-значение из нашей хэш-таблицы. С помощью метода sort, отсортируем его по не возрастанию. Теперь для решение поставленной задачи остается вывести 10 первых элементов этого массива.

Результат работы: 

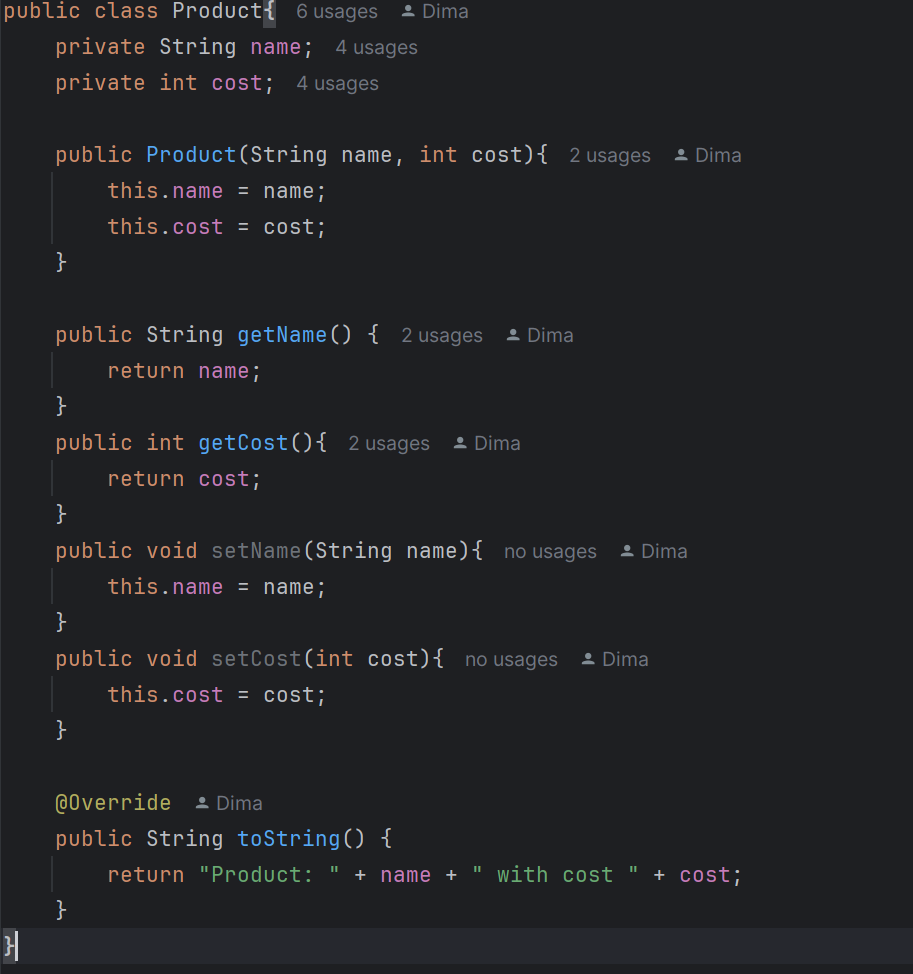
1. Я написал обобщенный класс Stack.



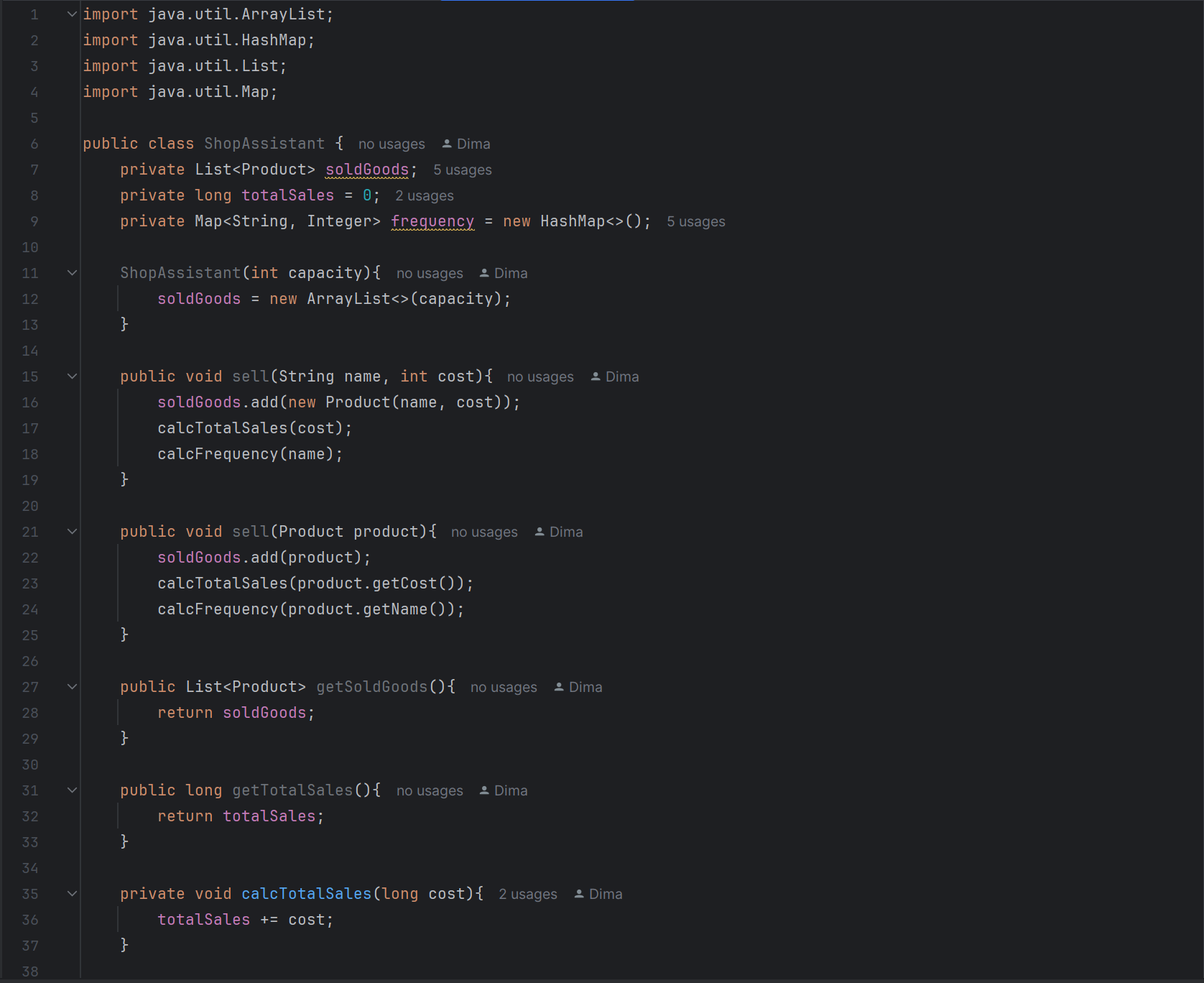
data – массив, который хранит значения, лежащие в стеке, head – указатель на вершину массива. При создании массива, конструктор принимает в себя один параметр – размер массива (стека). Также в этом классе есть три метода. Метод push добавляет элемент на вершину стека. Метод pop снимает верхний элемент со стека и возвращает его. Метод peek позволяет посмотреть какой-элемент находиться на вершине стека, при этом не убирая его из стека. Также есть вспомогательный метод isEmpty, который используется для проверки пустоты массива (чтобы лишний раз не выходить за границы).

Пример работы: 

1. Я написал программу для учета продаж в магазине. Для этого я сначала реализовал вспомогательный класс Product:



Этот класс представляет собой продукты, которые будет продавать магазин. Его можно удобно расширить, но пока что, для решения поставленных задач мне понадобиться только два поля: name, cost. Имя продукта и его стоимость. Также в классе есть один конструктор, геттера и сеттеры.

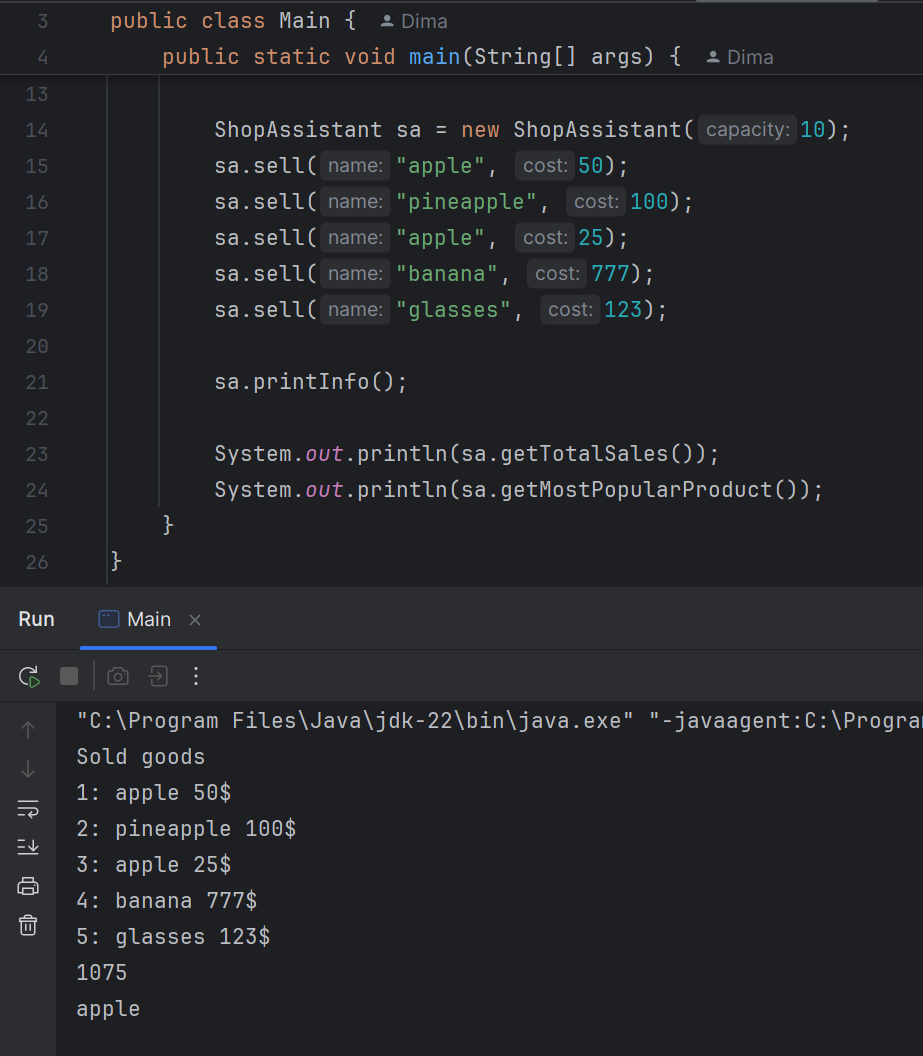
Теперь сам класс ShopAssistant: 

soldGoods – массив из коллекции ArrayList, в котором хранятся товары Product. Также есть поле totalSales и frequency. В totalSales мы будем динамически считать сумму продаж. (Предполагается, что наш магазин не оформляет возвраты и цены каким-то образом меняться не будут). Frequency представляет собой hash-таблицу, в который мы будем динамически подсчитывать частоты продуктов (аналогично п. 1) ).

Есть конструктор, который определяет емкость массива. Два перегруженных метода sell, которые добавляют продукт в массив, считают totalSales и заполняют frequency. Методы, которые нам необходимо было реализовать это getTotalSales(), printInfo(), getMostPopularProduct().

getTotalSales возвращает переменную totalSales. prinInfo() красиво выводит список товаров soldGoods в консоль. getMostPopularProduct() на основе frequency возвращает самый продаваемый продукт.

Пример работы:



Исходный код: <https://github.com/TAskMAster339/ITIP_Lab6>

**Вывод:**

Я научился работать с коллекциями в Java.