Geekbrains

**Разработка проекта для сбора посылки «Посылка для дедушки» на Java.**

IT-специалисты:

Разработка проекта

Шелыхаева А.А.

Кострома

2024

**Дипломный проект**

**«Разработка проекта для сбора посылки «Посылка для дедушки» на Java»**

**Содержание**

|  |  |
| --- | --- |
| Введение……………………………………………………………………. | 3 |
| Глава 1. Предпроектная стадия…………………………………………… | 4 |
| 1.1. Порядок проведения работ…………………………………………… | 4 |
| 1.2. Определение требований к приложению…………………………… | 5 |
| 1.3. Выбор стека технологий……………………………………………… | 5 |
| Глава 2. Проектная стадия………………………………………………… | 6 |
| 2.1. Проектирование общей архитектуры приложения………………… | 6 |
| 2.2. Проектирование web-интерфейса приложения…………………….. | 15 |
| Заключение………………………………………………………………… | 18 |
| Список использованной литературы…………………………………….. | 20 |

**Введение**

Виртуальная игра "Посылка для дедушки" представляет собой приложение, в котором предлагается на выбор набор товаров для формирования посылки.

В приложении «Посылка для дедушки» реализовано несколько действий:

1. Перечень товаров в наборе для посылки

2. Сколько всего товаров в наборе для посылки

3. Самое длинное название товара наборе

4. Самые популярные товары в наборе

5. Самые редкие товары в наборе

0. Выход из приложения

«Посылка для дедушки» предлагает удобный и интуитивно понятный интерфейс для выбора товаров, предназначенных для формирования посылки.

Основная функциональность приложения включает в себя:

**Возможность выбора**: Пользователь может выбрать любое действие из выпадающего списка.

**Подсчет символов**: в приложении реализовано действие по нахождению слова по максимальному количеству символов.

**Нахождение популярных и редких товаров**: Пользователь может запросить у приложения найти самые популярные и редкие товары из имеющегося списка.

**Выход из приложения**: в приложении реализована команда выхода из приложения.

**Глава 1. Предпроектная стадия**

**1.1. Порядок проведения работ**

В ходе создания проекта были проведены следующие виды работ:

1. **Планированиепроекта**:
   * Определение функциональных и нефункциональных требований к приложению.
   * Проектирование архитектуры приложения и выбор используемых технологий.
2. **Настройка окружения разработки**:
   * Установка и настройка Java,
3. **Создание проекта**:
   * Создан новый проект с использованием Map, HashMap, TreeMap, HashTable
   * Выбраны необходимые зависимости, такие как  Set, HashSet, TreeSeеt, LinkedHashSet, для поддержки функциональности приложения.
4. **Разработка модели данных, сервисов, контроллера.**
5. **Разработка пользовательского интерфейса**.
6. **Отладка и оптимизация**:
   * Отладка кода для исправления ошибок и улучшения производительности.
   * Оптимизация кода и конфигурации приложения для улучшения скорости работы и эффективности использования ресурсов.
7. **Документирование**:
   * Создание документации по проекту, включая описание архитектуры, инструкции по установке и запуску, и другие справочные материалы.

**1.2. Определение требований к приложению**

Для разработки приложения "Посылка для дедушки" были определены следующие требования:

1. Функциональные требования:
   * Пользователи должны иметь право выбора действий из выпадающего списка.
   * Приложение должно предоставлять возможность выбора товаров для формирования посылки.
   * Приложение должно предоставлять возможность визуализации данных о собранной посылке.
2. Нефункциональные требования:
   * Удобство использования: Интерфейс должен быть интуитивно понятным и привлекательным для пользователя.
   * Производительность: Приложение должно обеспечивать быстрый доступ к данным и отзывчивый интерфейс.
3. Требования к интерфейсу пользователя:
   * Интуитивно понятный интерфейс, который позволяет пользователям легко делать выбор действий.
4. Требования к производительности:
   * Быстрый доступ к данным и отзывчивый интерфейс приложения.

**1.3. Выбор стека технологий**

1. **Язык программирования**: Java для разработки серверной части приложения так как этот язык хорошо подходит для создания масштабируемых и надежных веб-приложений.
2. **Среда разработки**: IntelliJ IDEA для разработки серверной части приложения. Этот стек технологий обеспечивает разработку приложения "Посылка для дедушки" с учетом требований к функциональности, производительности и безопасности.

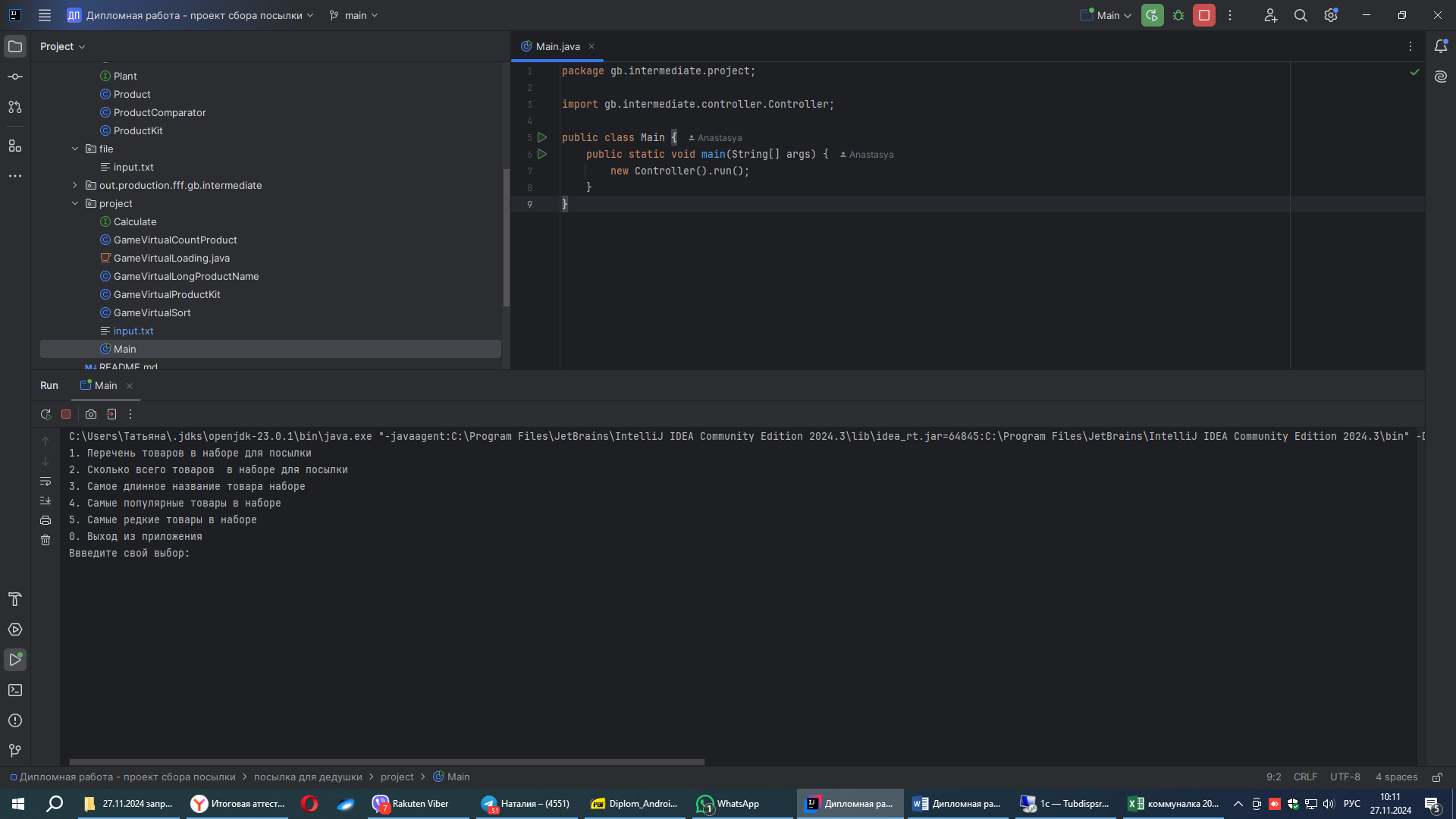
Архитектура приложения в рамках JavaSpring основывается на паттерне проектирования MVC (Model-View-Controller), который позволяет разделить приложение на три основных компонента: модель (Model), представление (View) и контроллер (Controller). Наше приложение состоит из данных составных частей:

1. **Модель (project)**:
   * Модель представляет собой структуру данных приложения.
   * Включает классы, описывающие сущности приложения ("Calculate", "GameVirtualCountProduct", ''GameVirtualLongProductName'', ''GameVirtualProductKit'', ''GameVirtualSort'' ).
2. **Представление (example)**:
   * Представление отвечает за отображение данных пользователю, включает следующие классы: ''Grass'', ''Kit'', ''Plant'', ''Product'', ''ProductComparator'', ''ProductKit''.
3. **Контроллер (Controller)**:
   * Контроллер обрабатывает запросы от пользователя и взаимодействует с моделью и представлением.
   * В Java контроллеры представлены классами: ''Controller'', ''GetChoice'', ''LoadProductFromFile''.
   * Контроллеры содержат методы для обработки различных типов запросов (GET, POST, PUT, DELETE) и вызывают соответствующие сервисы для выполнения логики.
4. **Сервис (Main)**:
   * Сервис содержит логику приложения.
   * Он обычно используется для выполнения операций с данными, координирования действий между различными компонентами приложения и обеспечения единого доступа к функциональности приложения.

Каждый из этих компонентов играет важную роль в построении архитектуры приложения на Java, обеспечивая его модульность, гибкость и масштабируемость. Они позволяют разделить различные аспекты приложения и обеспечить их независимую разработку и тестирование.

|  |
| --- |
| **Глава 2. Проектная стадия** |
| **2.1. Проектирование общей архитектуры приложения** |

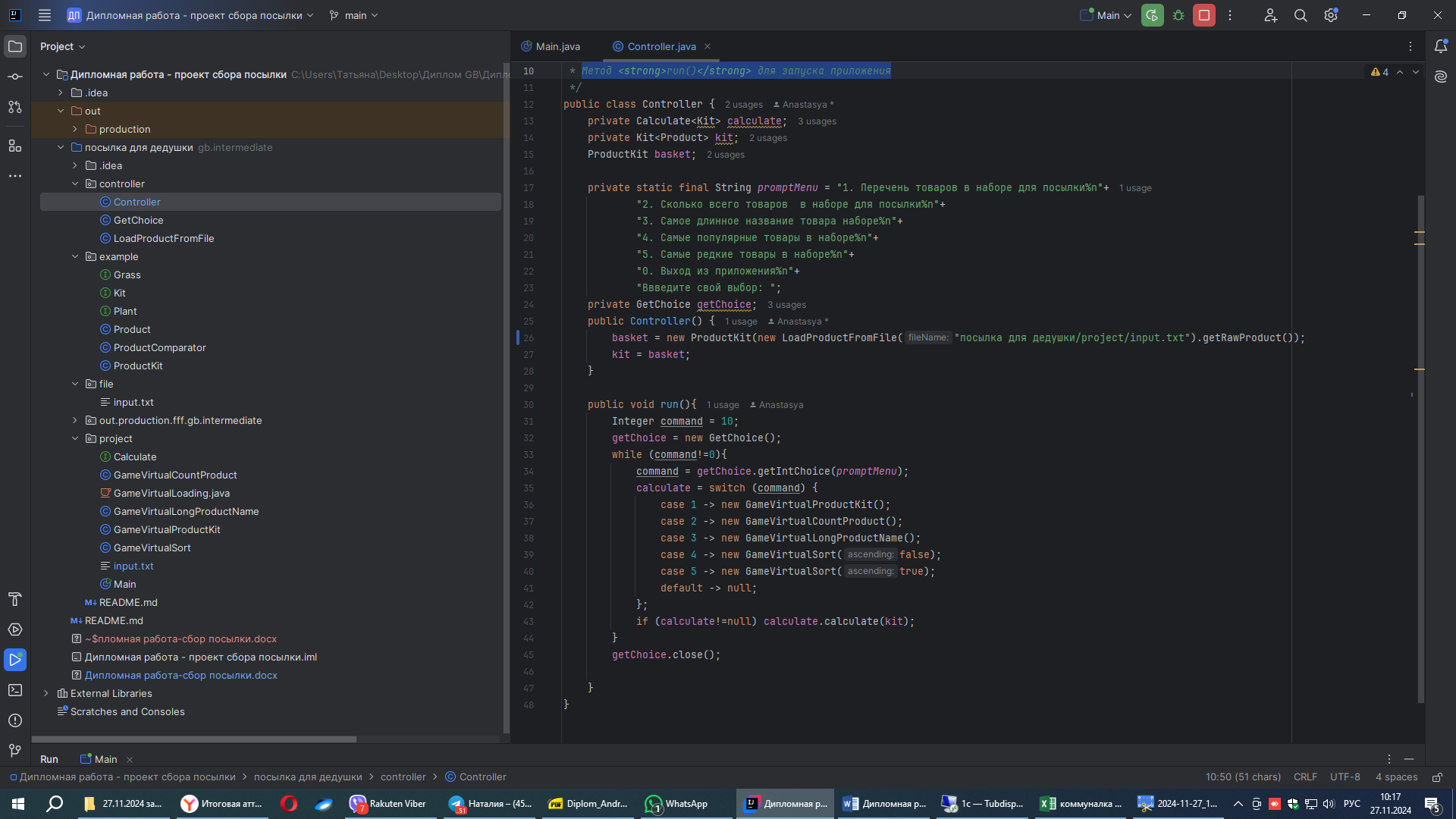
Первое, с чего начиналась работа - это запуск Main. Класс Main представляет собой реализацию приложения с выпадающем списком действий. Пользователю предлагается право выбора.



Controller является компонентом веб-приложения, отвечающим за взаимодействие с пользователем и обработку запросов, связанных с управлением перечня действий и проектом (project). Контроллер предоставляет различные методы для обработки запросов пользователей и взаимодействия с соответствующими сервисами.

Метод strong используется для запуска приложения

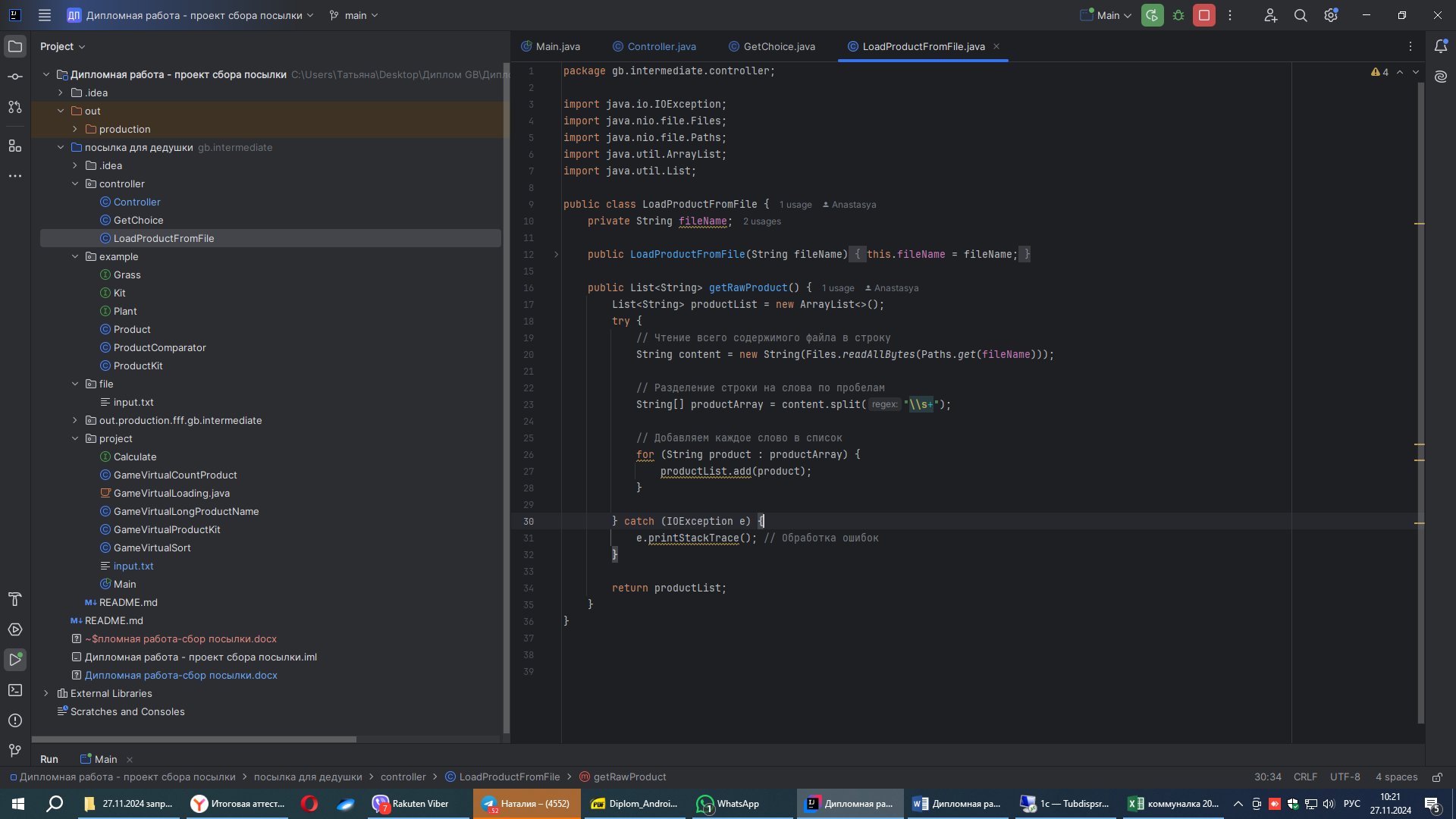
Методы контроллера позволяют пользователю выбирать список действий:



Класс GetChoice приглашает пользователя к выбору:



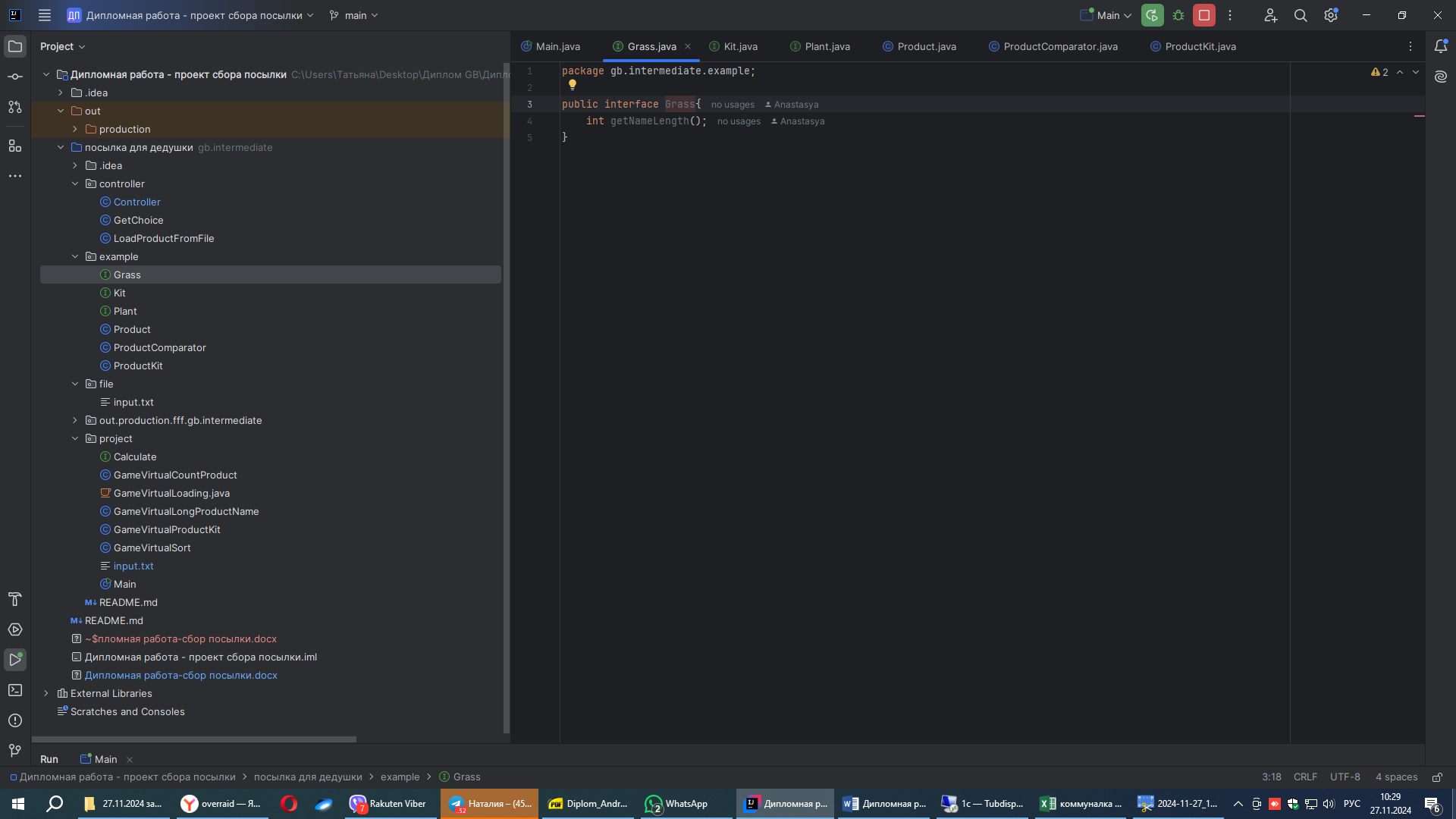
Класс LoadProductFromFile обрабатывает зарос в части чтения файла, разделяет строку на слова по пробелам, добавляет каждое слово в список:



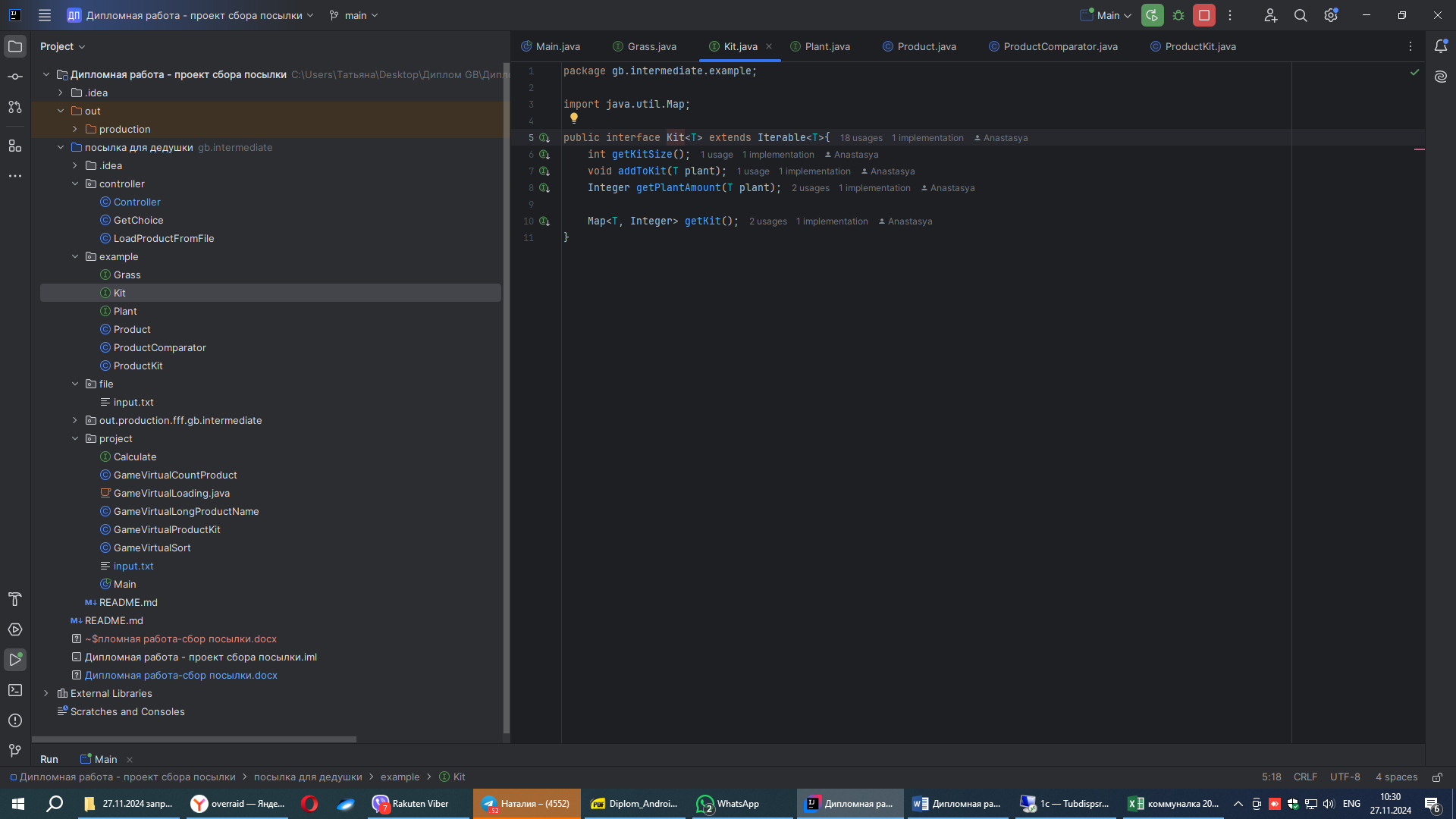
В целом, Controller является важной частью приложения, обеспечивающей удобный интерфейс для работы с выбором действий и проектом.

Компонент Example содержит классы, которые переопределяют базовые методы с помощью аннотации @Ovverride:

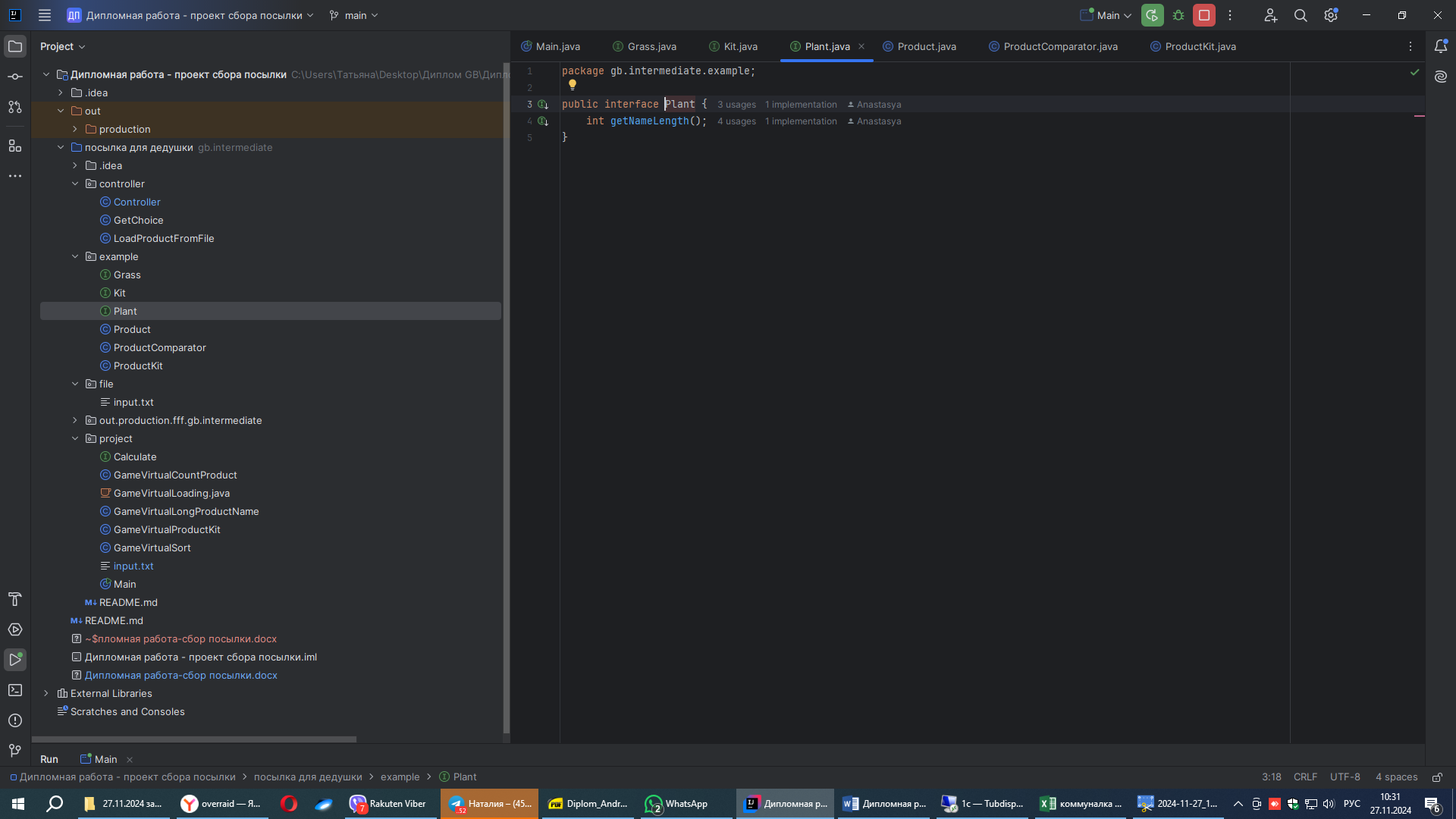
Класс Grass



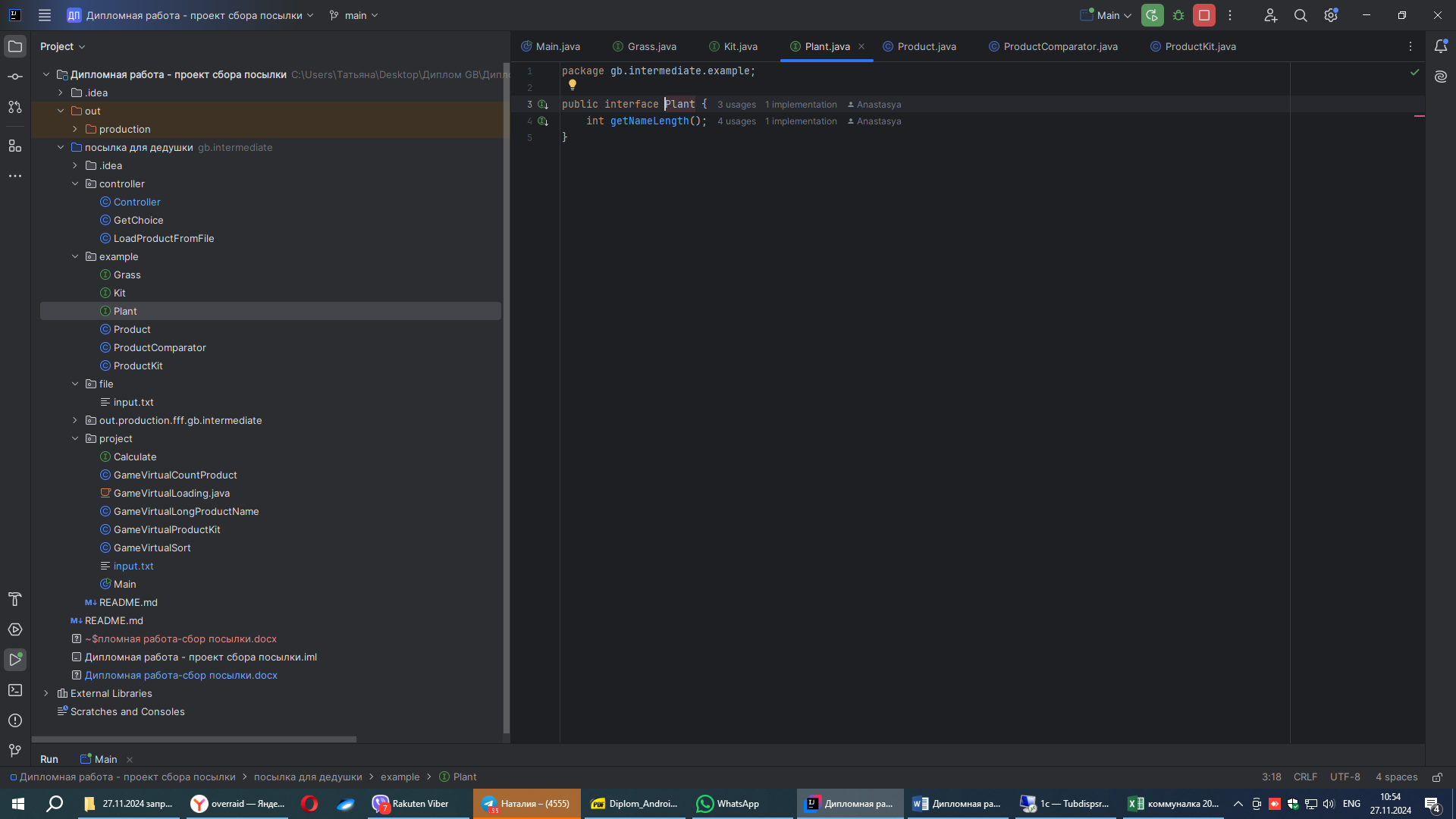
Класс Kit



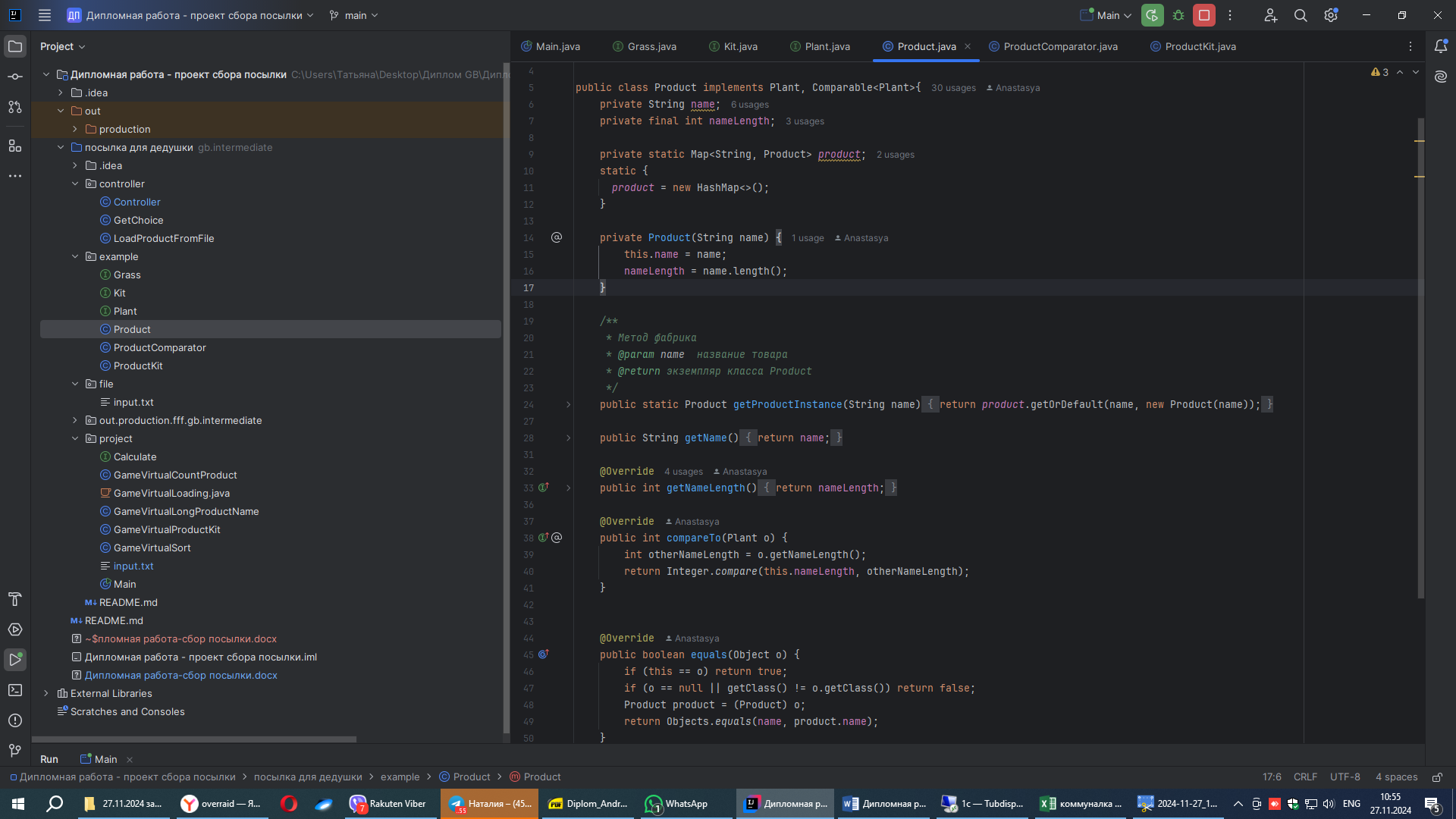
Класс Pit



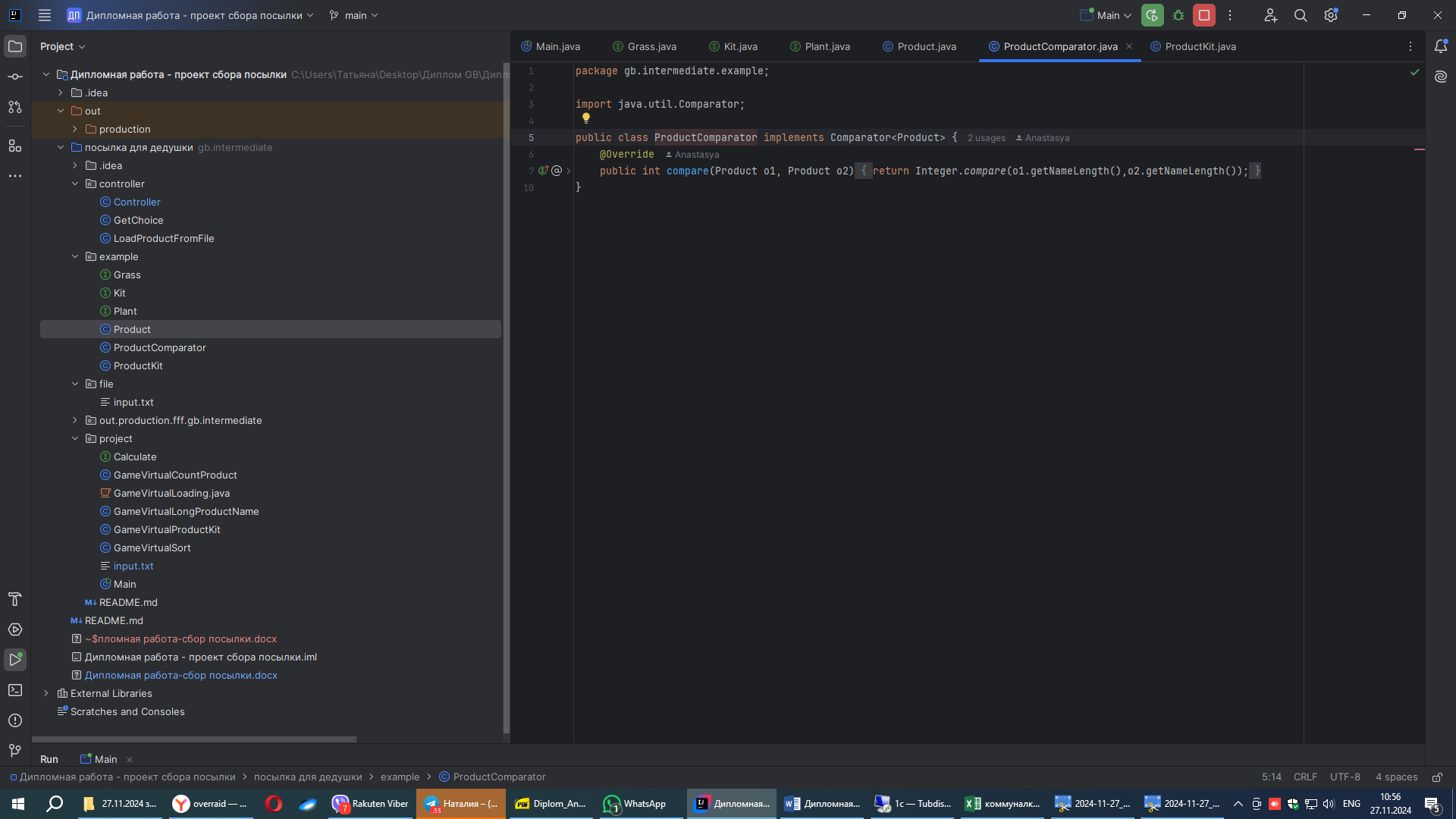
Класс Plant



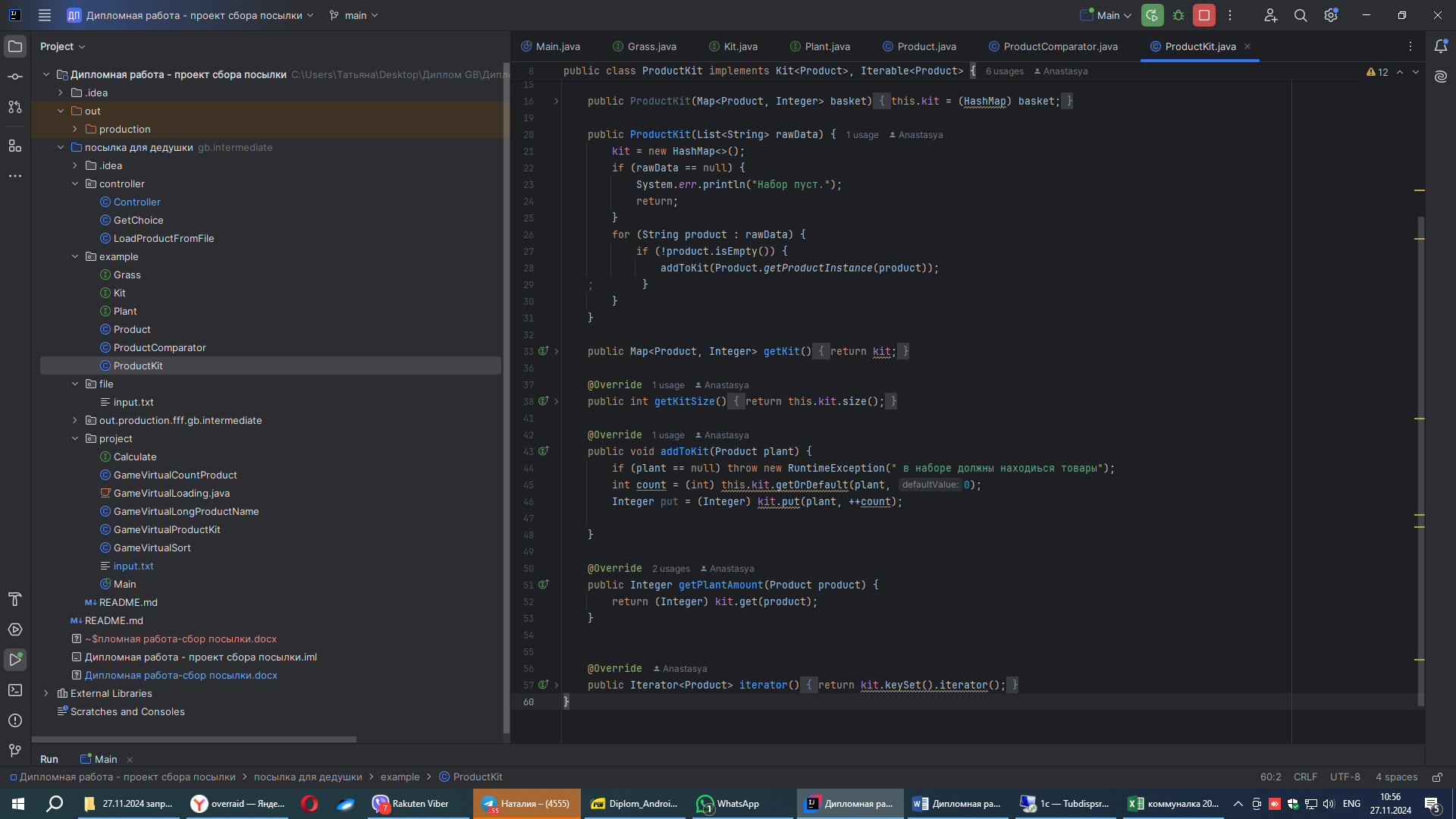
Класс Product



Класс ProductComparator

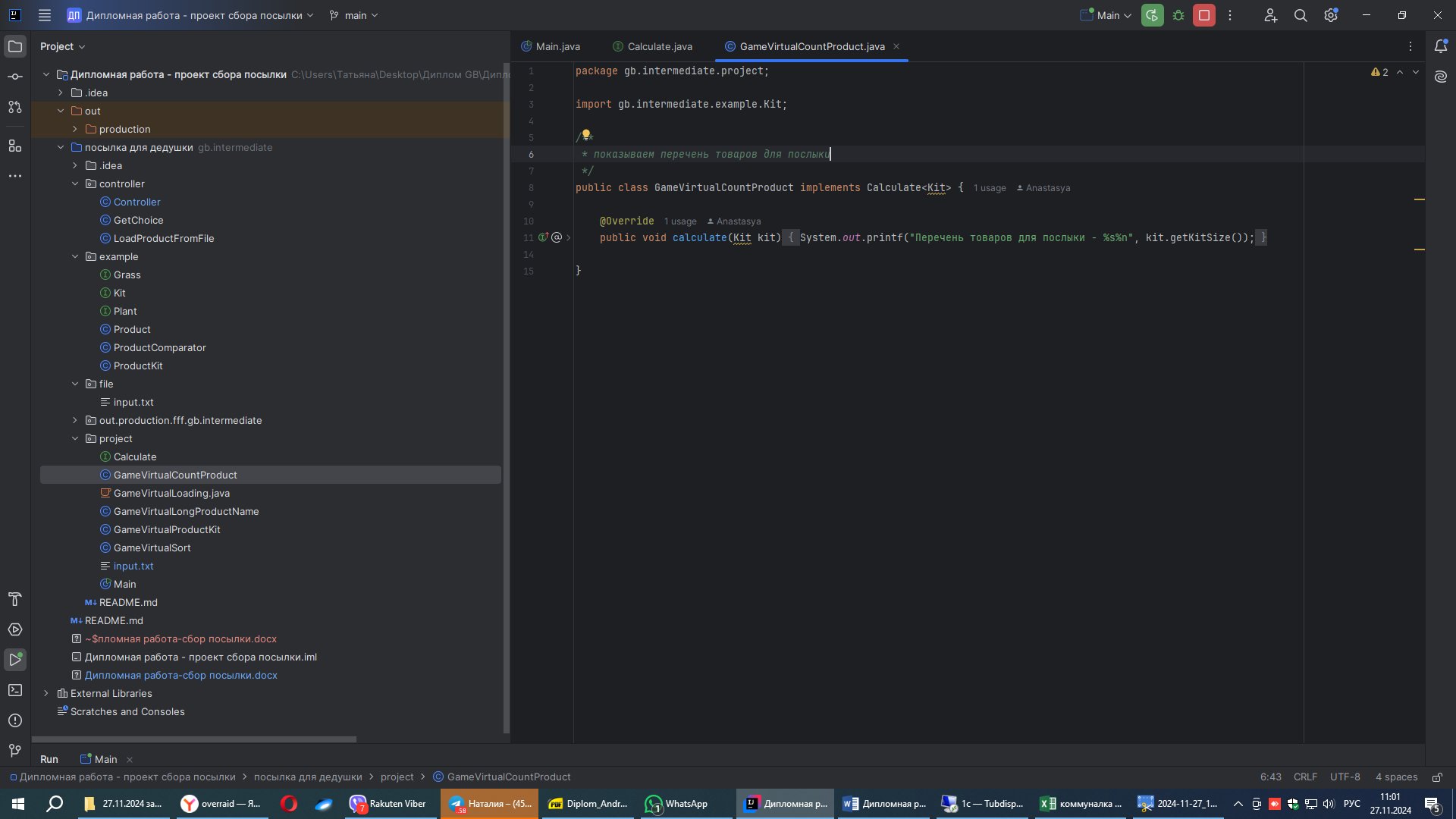


Класс ProductKit

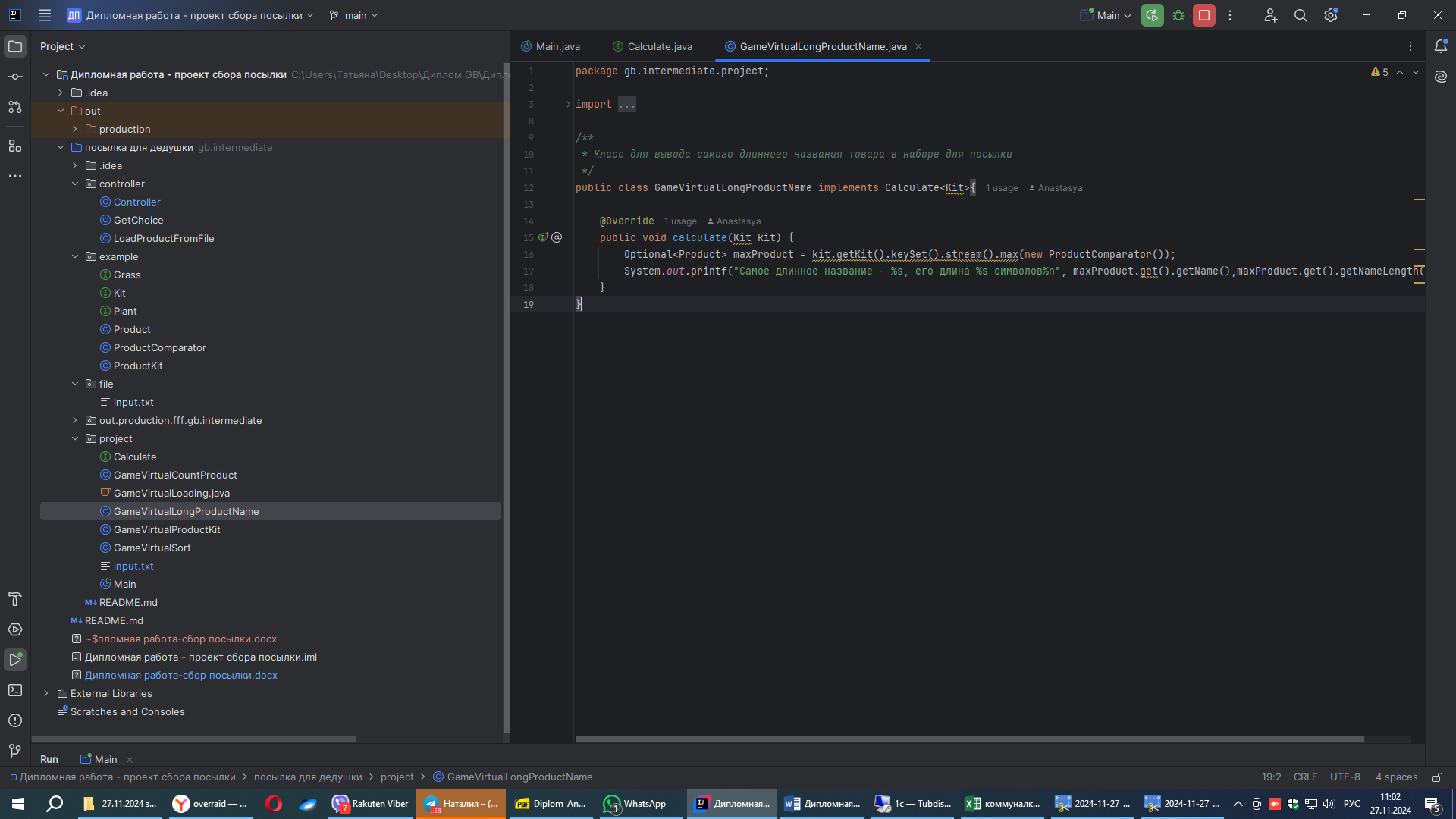


Метод Project содержит набор классов, которые помогают реализовать приложение:

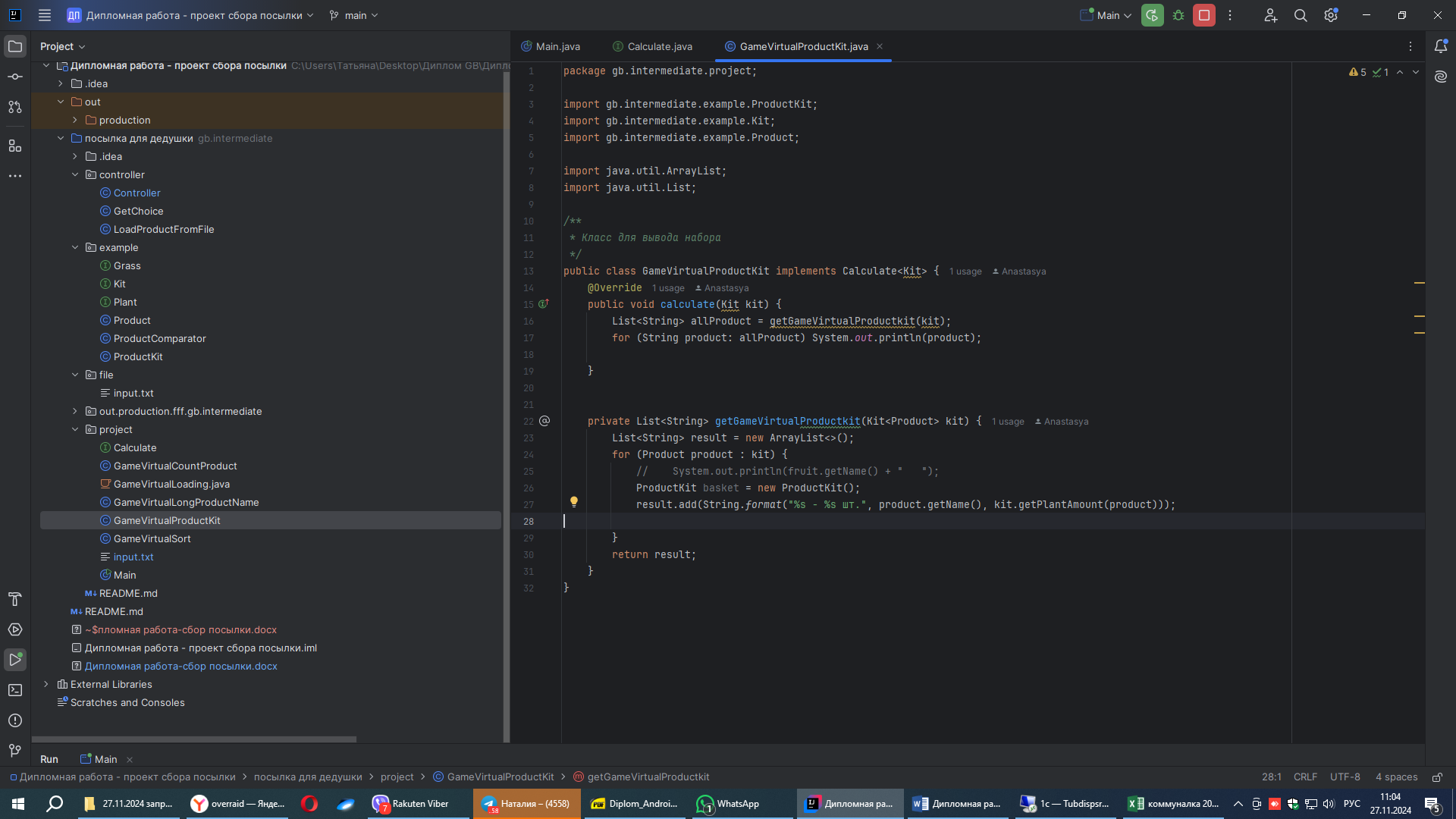
Класс GameVirtualCountProduct показывает перечень товаров для посылки



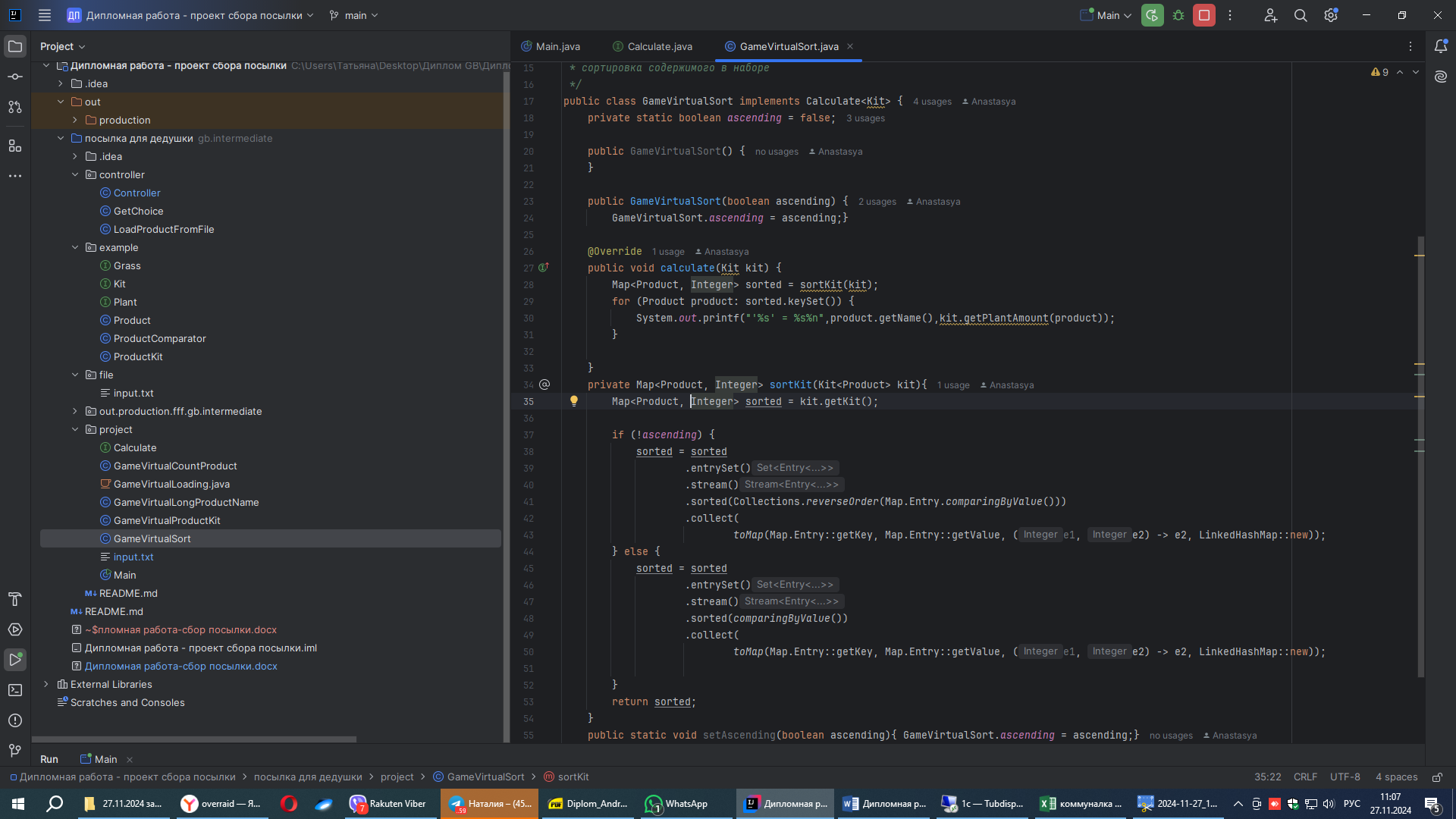
Класс GameVirtualLongProductName показывает самое длинное название товара в посылке



Класс GameVirtualProductKit выводит перечень товаров в наборе:

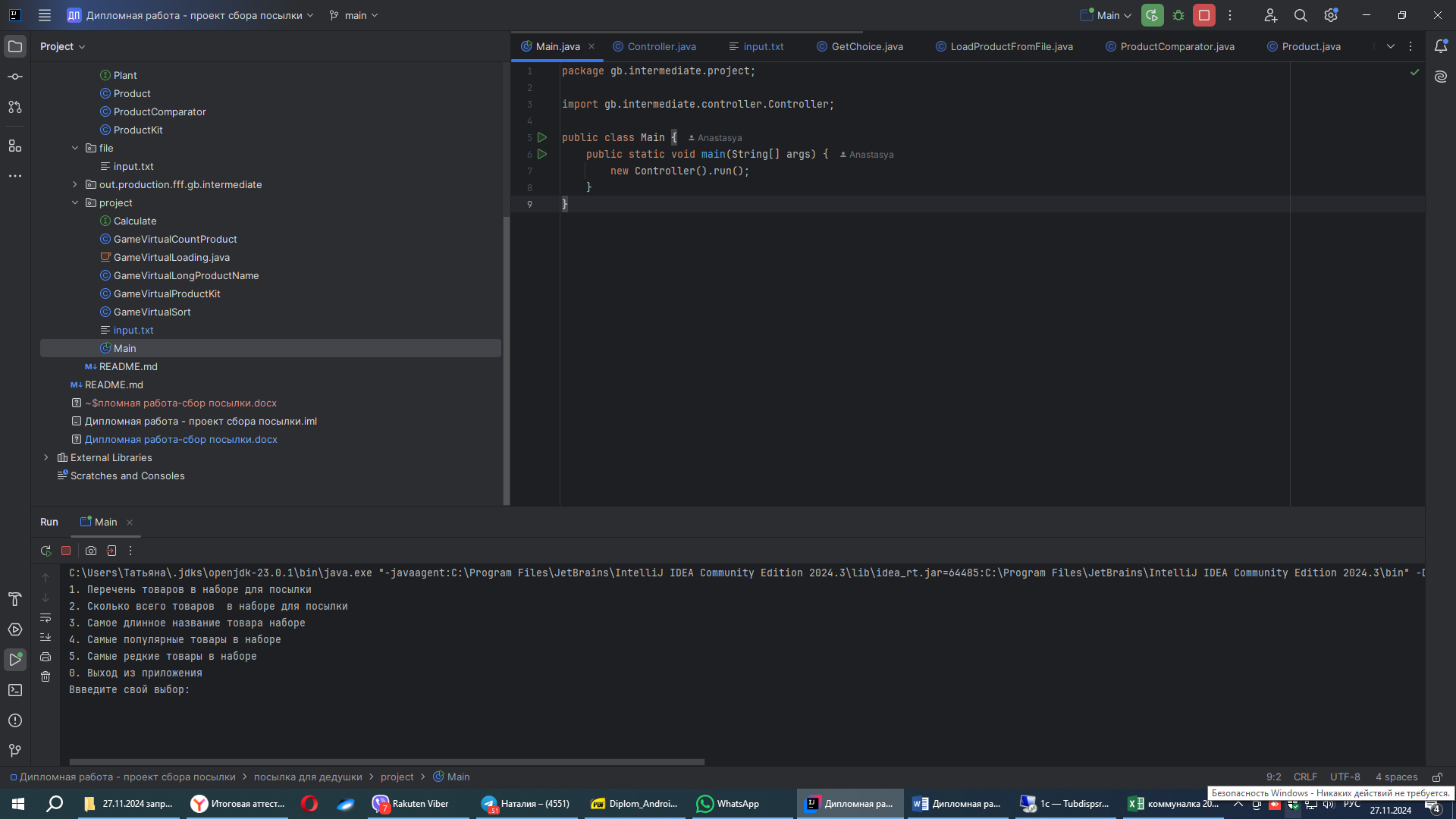


Класс GameVirtualSort сортирует содержимое в наборе:



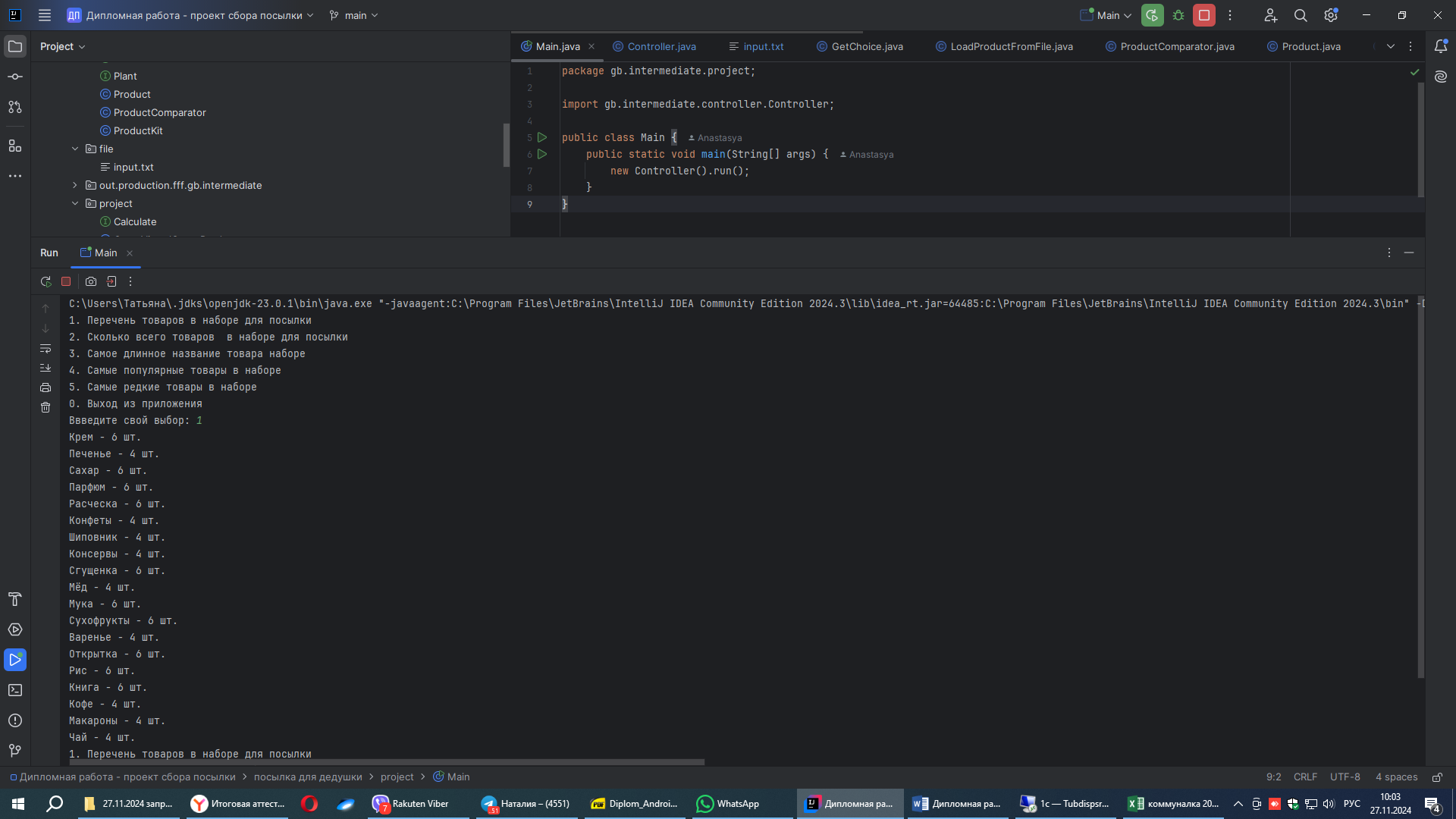
**2.2. Проектирование WEB – интерфейса приложения**

После вызова команды Main пользователю выпадает список действий:

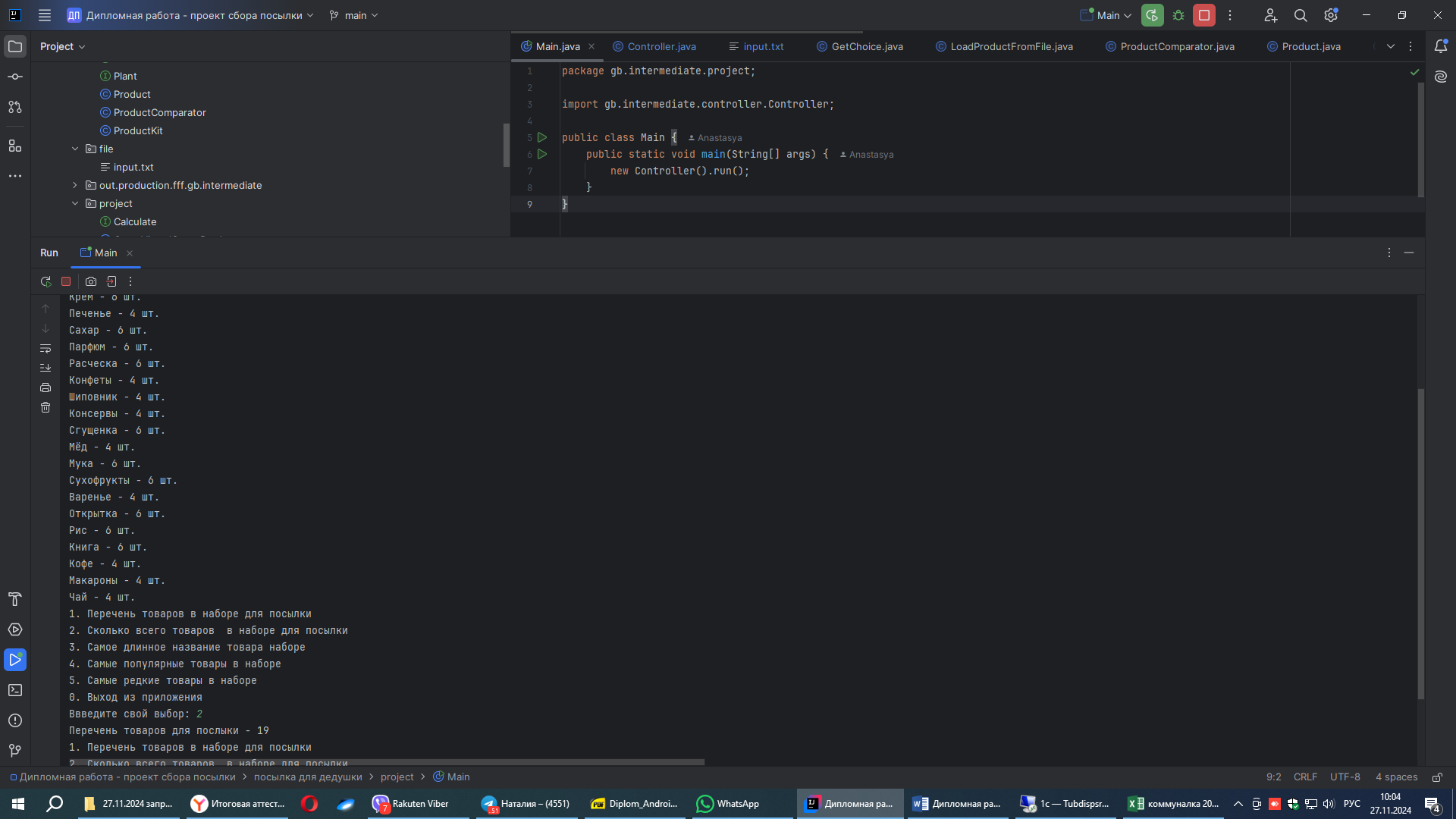


Начало формы

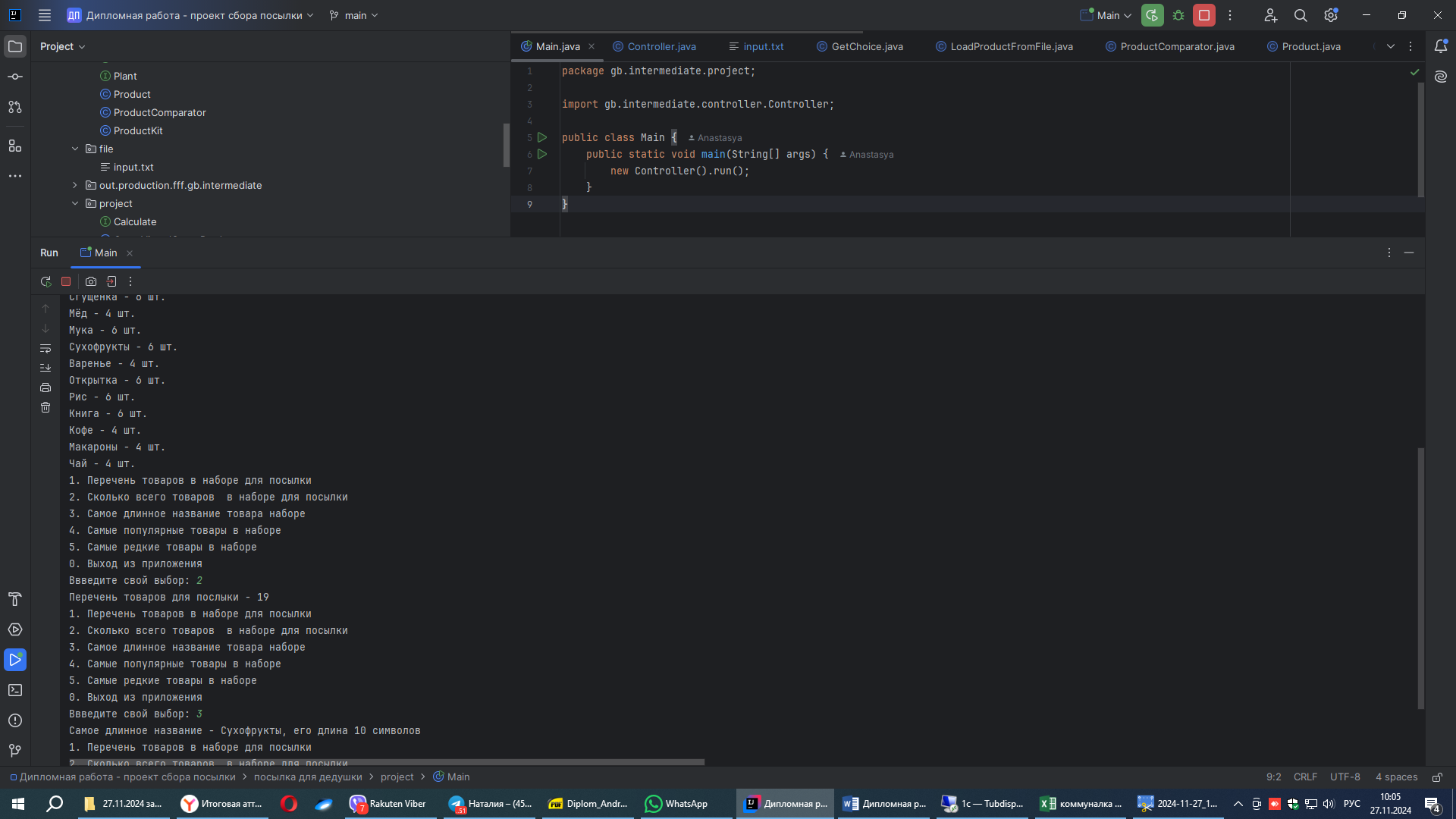
При выборе первого действия - открывается «Перечень товаров в наборе для посылки»:



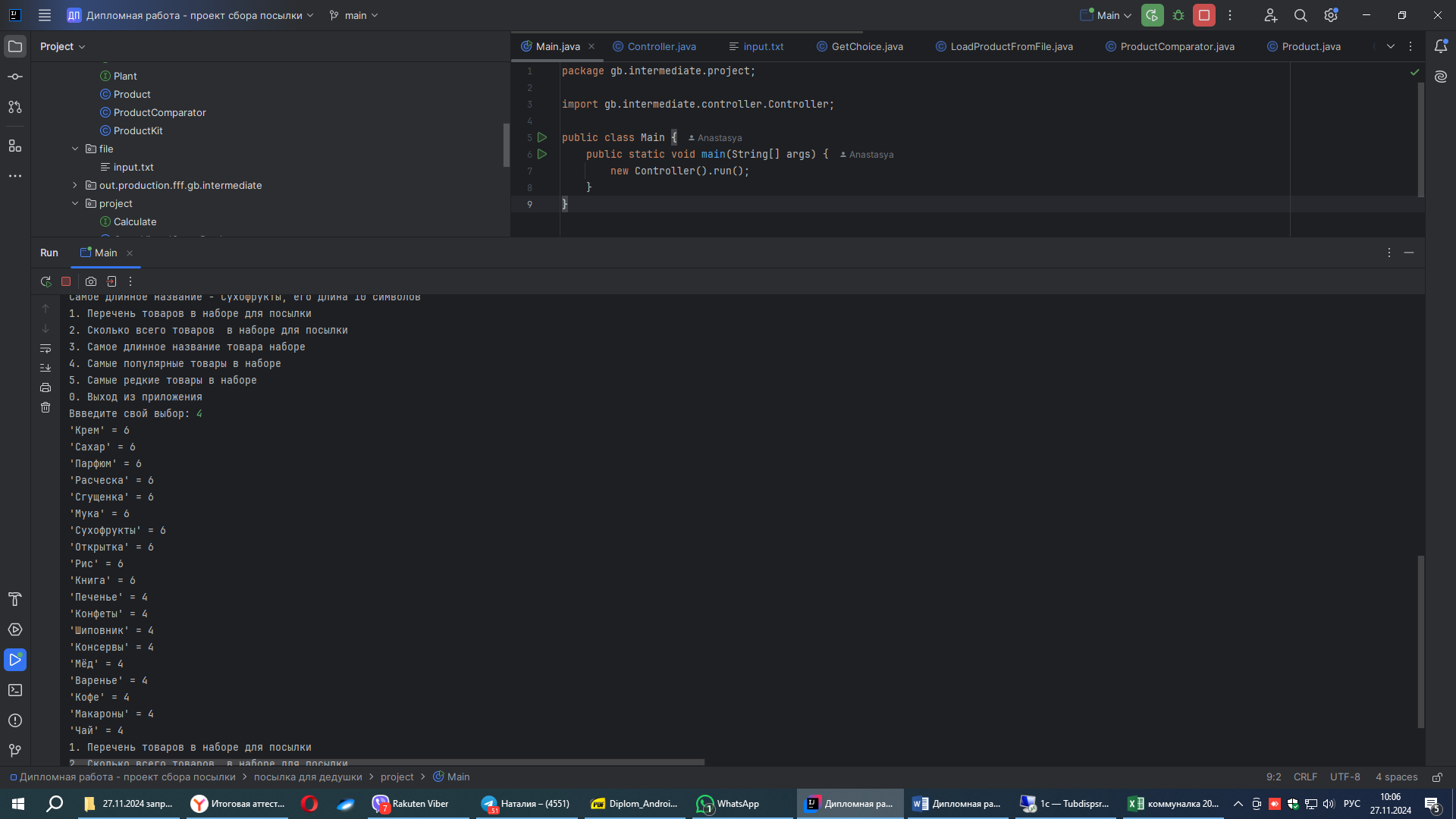
При выборе второго действия – увидим «Сколько всего товаров в наборе для посылки»:



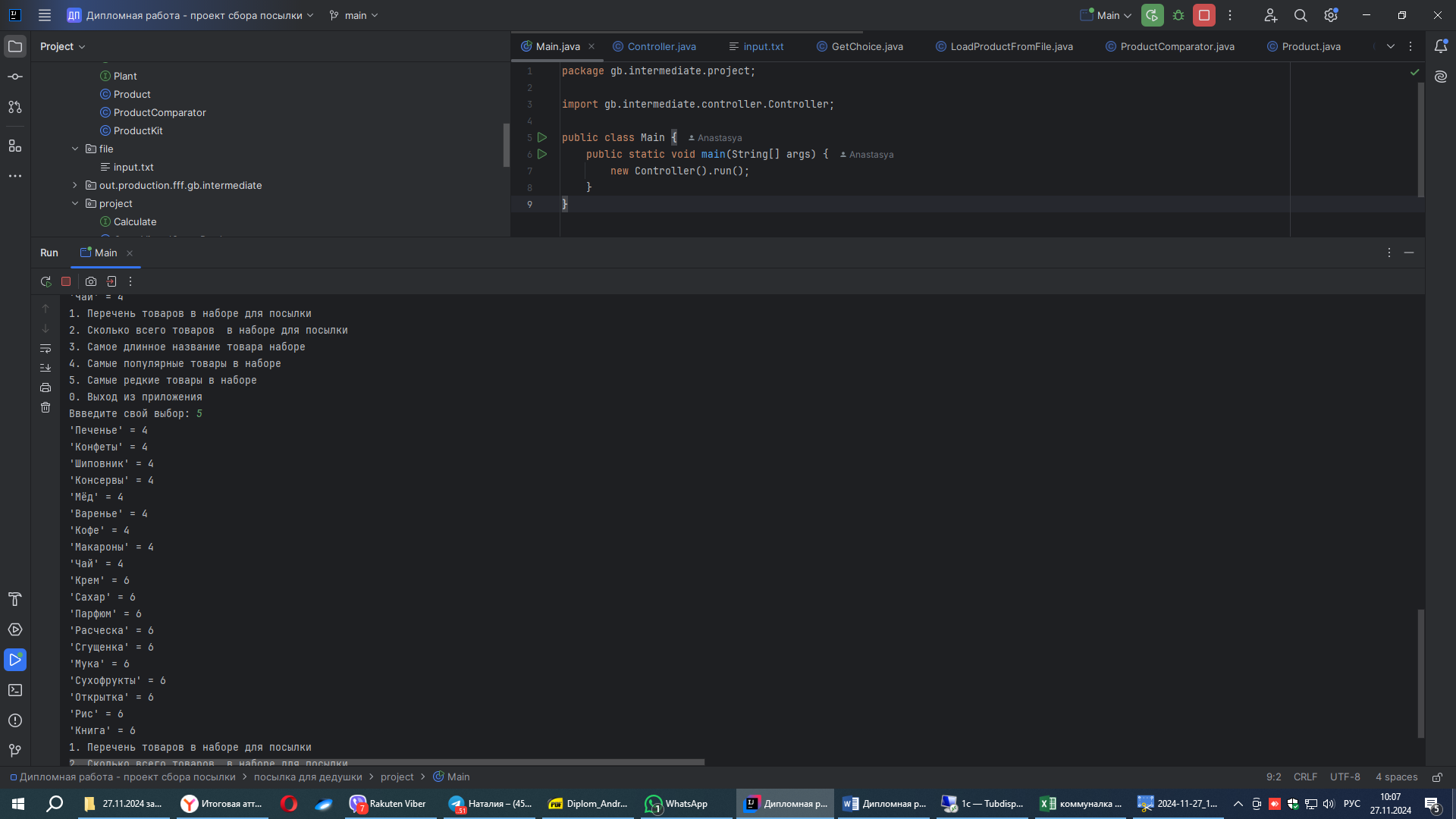
Третье действие покажет «Самое длинное товара в наборе»:



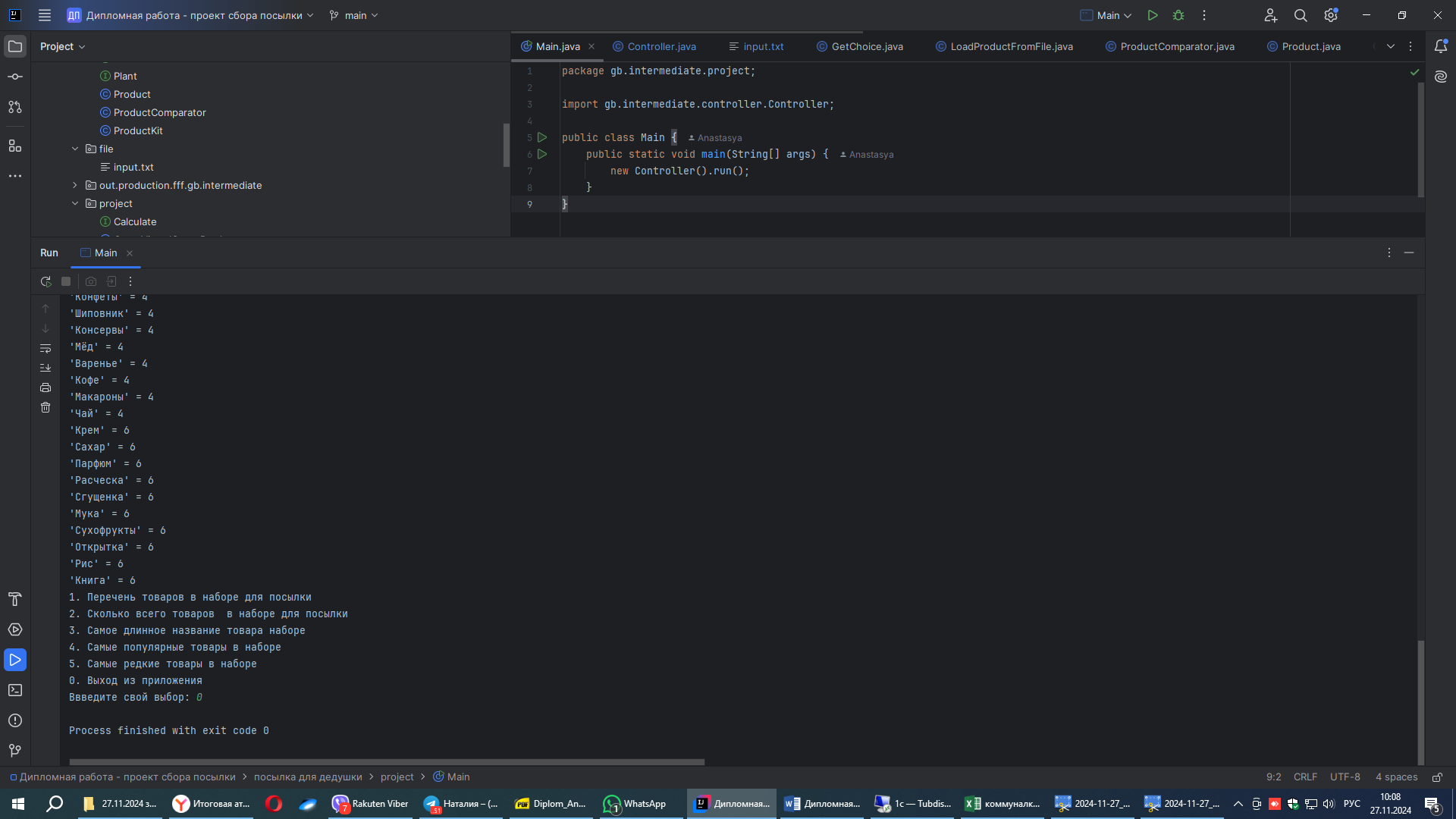
Четвертое действие выведет на экран «Самые популярные товары в наборе»:



Пятое действие покажет «Самые редкие товары в наборе»:



И, последнее действие, позволяет выйти из приложения:



**Заключение**

В рамках данного дипломного проекта было разработано приложение «Посылка для дедушки», предназначенное для выбора действий пользователя. Проект был реализован с использованием современных технологий веб-разработки, таких как Java и др.

В процессе разработки были достигнуты следующие цели:

* Создание удобного и интуитивно понятного интерфейса для пользователей приложения.
* Реализация функционала в выборе действий из выпадающего списка.

Дальнейшее развитие приложения может включать в себя добавление нового функционала, такого как выход в приложение нескольких пользователей, возможность подключения календаря для отображения собранных посылок во времени.

В целом, дипломный проект "Посылка для дедушки" позволил не только показать навыки разработки веб-приложений с использованием современных технологий, но и продемонстрировать понимание принципов удобства использования для конечного пользователя.

**Список литературы**

1. Учебный материал Гикбрейнс.
2. Статьи и видео-материалы из сети интернет.