

Національний університет

**“КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ”**

**Лабораторна робота №2**

Звіт з лабораторної роботи з курсу “Мови програмування”

Виконали:

Войлов Богдан – ІПЗ - 1

Жорник Дмитро – ІПЗ - 1

Викладач: Кирієнко Оксана Валентинівна

**1.Постановка задачі**

Необхідно автоматизувати роботу невеликого підприємства по роботі з складом.

Існує декілька груп товарів (наприклад: Продовольчі, непродовольчі...). В кожній групі товарів існують конкретні товари (наприклад: борошно, гречка...). У кожного товару є наступні властивості - назва, опис, виробник, кількість на складі, ціна за одиницю. Група товарів містить наступні властивості - назва, опис.

Реалізувати:

1. Реалізувати графічний інтерфейс користувача
2. Збереження даних в файл/файли. Один з варіантів: Існує файл в якому знаходяться назви всіх груп товарів. Товари з кожної групи товарів знаходяться в окремому файлі.
3. Назва товару - унікальна (не може зустрічатися більше в жодній групі товарів).
4. Назва групи товарів - унікальна.
5. Реалізувати додавання/редагування/видалення групи товарів - при видаленні групи товарів, видаляти і всі товари.
6. Реалізувати додавання/редагування/видалення товару в групу товарів (мається на увазі назва, опис, виробник, ціна за одиницю).
7. Реалізувати інтерфейс додавання товару (прийшло на склад крупи гречаної - 10 штук), інтерфейс списання товару (продали крупи гречаної - 5 шт.)
8. Пошук товару.
9. Вивід статистичних даних: вивід всіх товарів з інформацією по складу, вивід усіх товарів по групі товарів з інформацією, загальна вартість товару на складі (кількість \* на ціну), загальна вартість товарів в групі товарів.
10. До роботи додати звіт про виконання роботи з описом розподілу ролей.

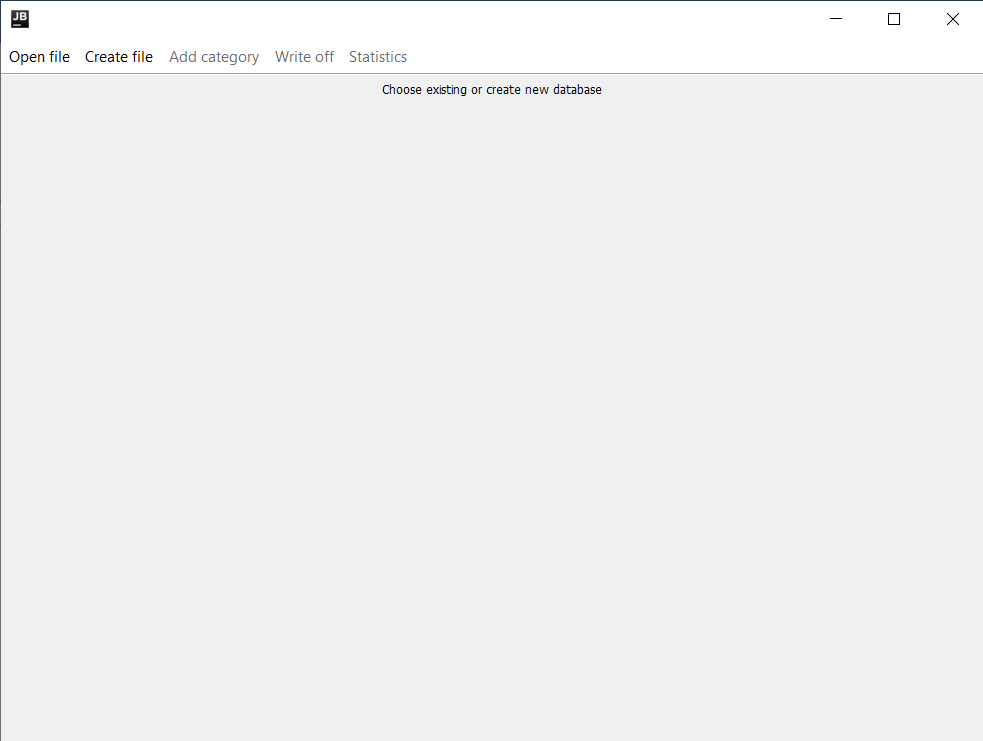
**2.Розподіл ролей**

При виконанні роботи ми вирішили використати архітектуру MVC, але через те що ми самі вибирали як зберігати дані та відповідно відсутність потреби у використанні ORM, controller і model ми вирішили об’єднати в один package. Богдан був відповідальний за controller та model. Дмитро – за view.

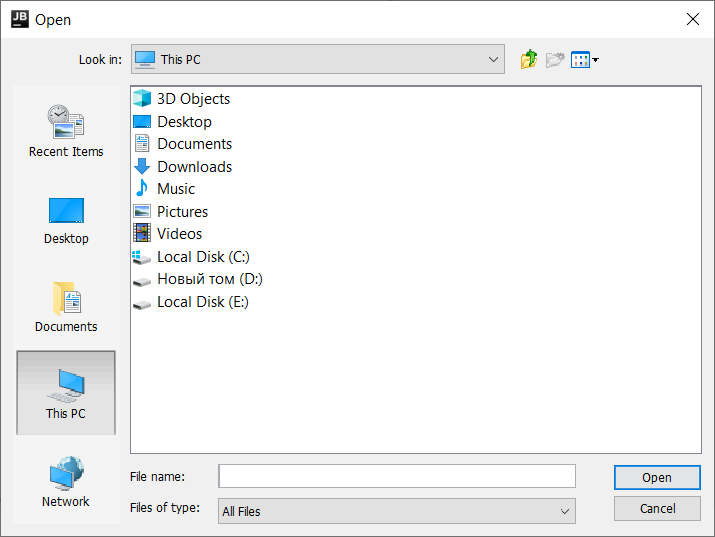
**3.Розподіл ролей**

При виконанні роботи ми вирішили використати архітектуру MVC, але через те що ми самі вибирали як зберігати дані та відповідно відсутність потреби у використанні ORM, controller і model ми вирішили об’єднати в один package. Богдан був відповідальний за controller та model. Дмитро – за view.

**5.Опис усіх реалізованих можливостей**

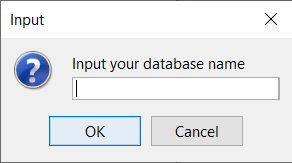


Це головне вікно. Тут можна або відкрити існуючий файл з даними або створити новий. До виконання одної з цих дій інші можливості закриті для користувача.

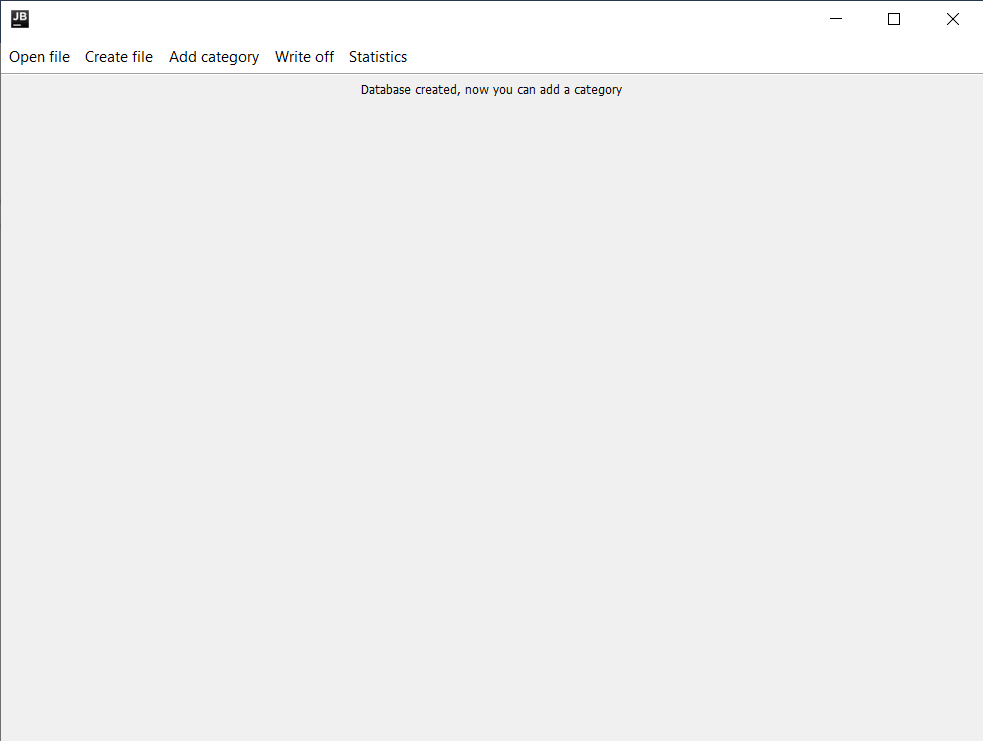


Це вікно вибору файлу з даними.

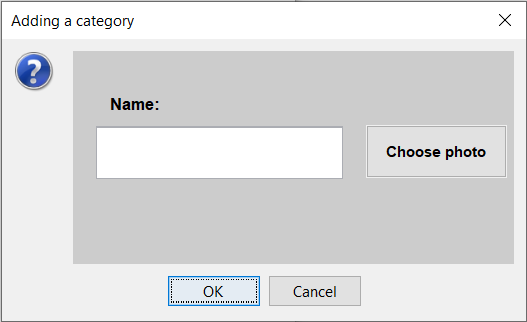
Якщо у вас ще немає готового файлу, його можна створити відповідною кнопкою. Система вам запропонує ввести ім’я.

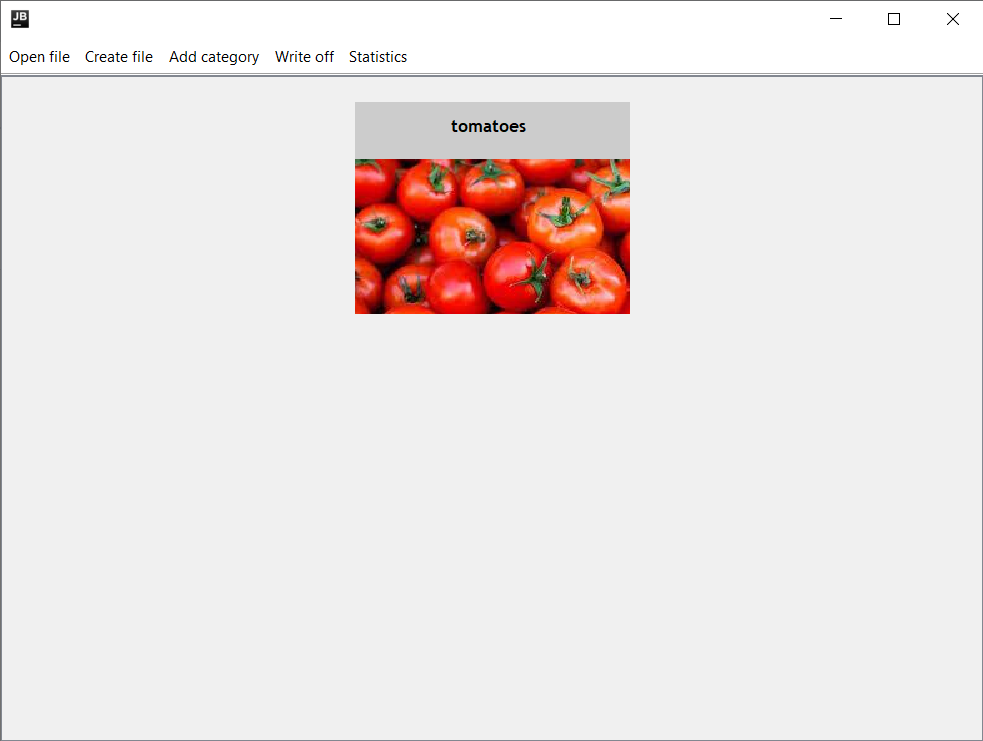


Після того як ви або відкриєте існуючий, або створите новий файл, для вас відкриється решта можливостей.



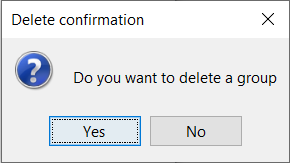
Щоб додати категорію або групу товарів потрібно натиснути відповідну кнопку, і система звернеться до вас з проханням назвати її і вибрати для неї певне зображення (краще у jpg форматі).



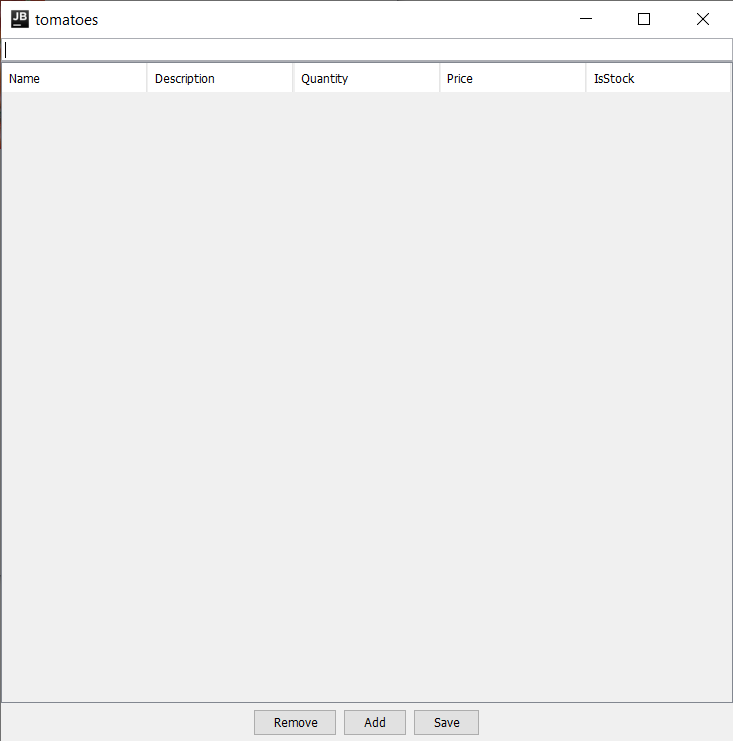


Тепер коли група створена, можна або редагувати вміст натиснувши лівою кнопкою миші на картку, або видалити натиснувши правою кнопкою миші.

Якщо ви спробуєте видалити система спитає у вас чи ви впевнені.

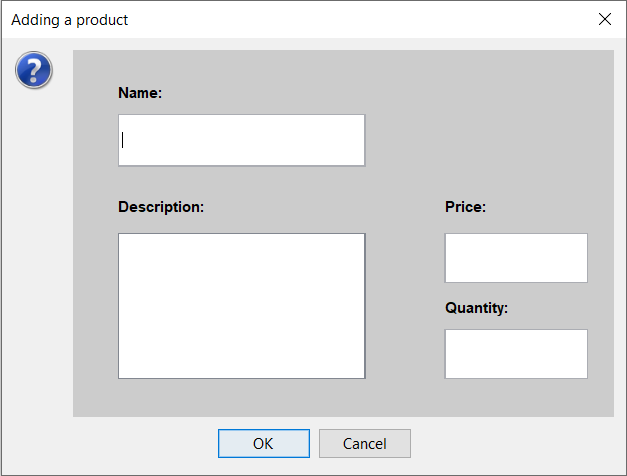


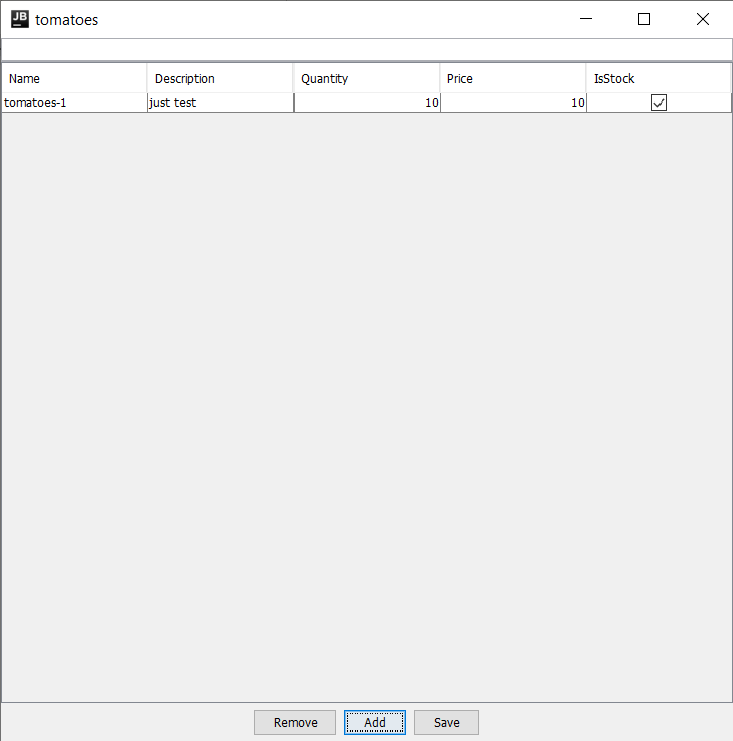
Якщо ви натиснете лівою кнопкою відкриється вікно редагування.



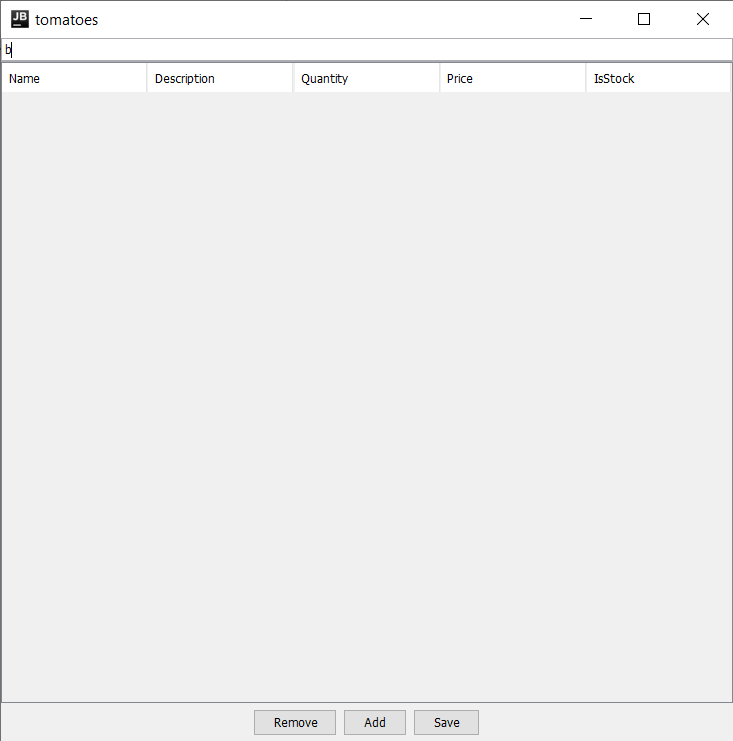
Зверху ми бачимо строку пошуку. Далі назви колонок, місце для таблиці і три кнопки для видалення і додавання стрічки та для збереження даних.

Якщо натиснути на кнопку “Add” відкриється форма в якій потрібно вказати відповідні поля таблиці.



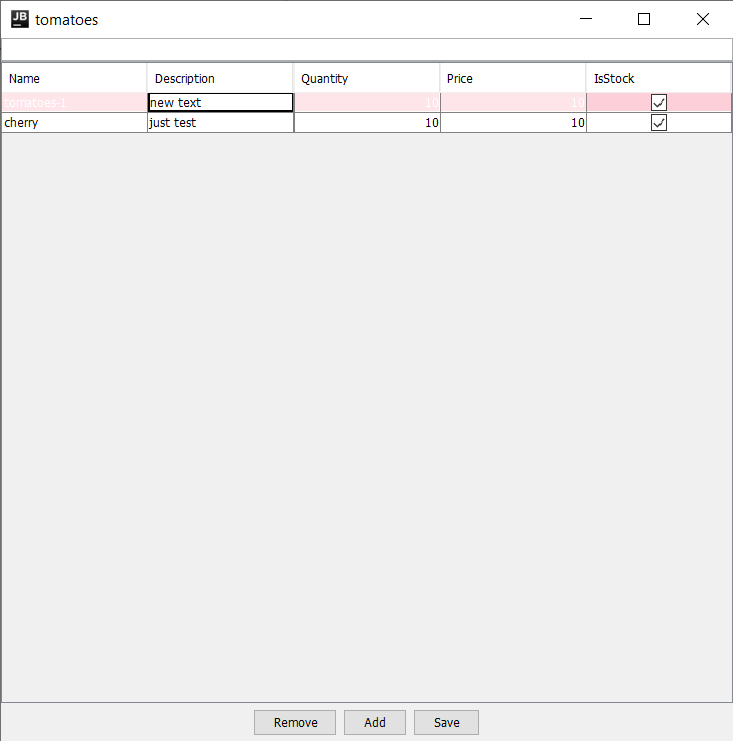


Після додавання наша стрічка опиниться у таблиці. Вже є сенс скористатися пошуком. Система шукає входження стрічки введеної користувачем у ім’я товару.



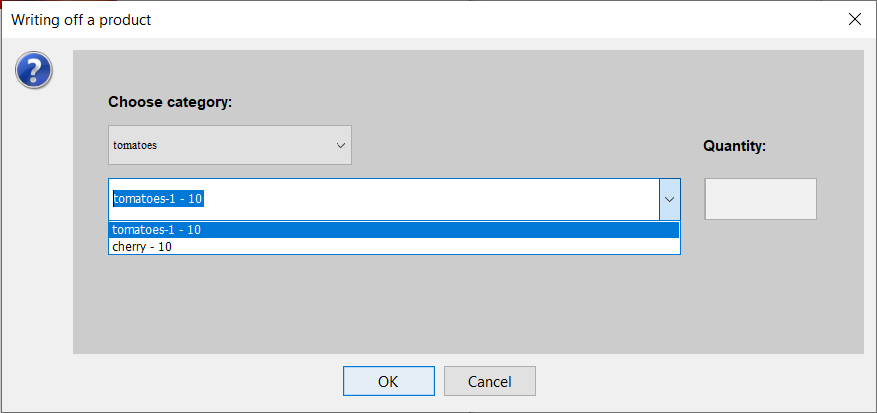
Як ми бачимо, у нашій таблиці немає товару з літерою “b” в імені.

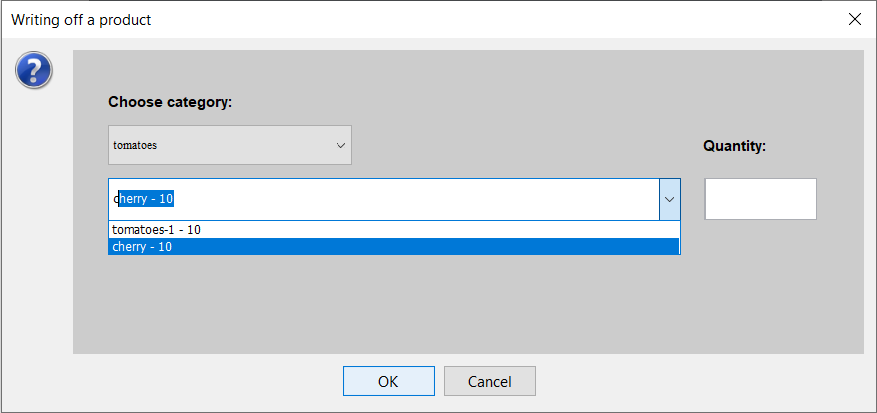
Натиснувши на відповідну клітинку у таблиці бачимо, що вся стрічка стала рожевою, і також є можливість змінювати вміст клітини.



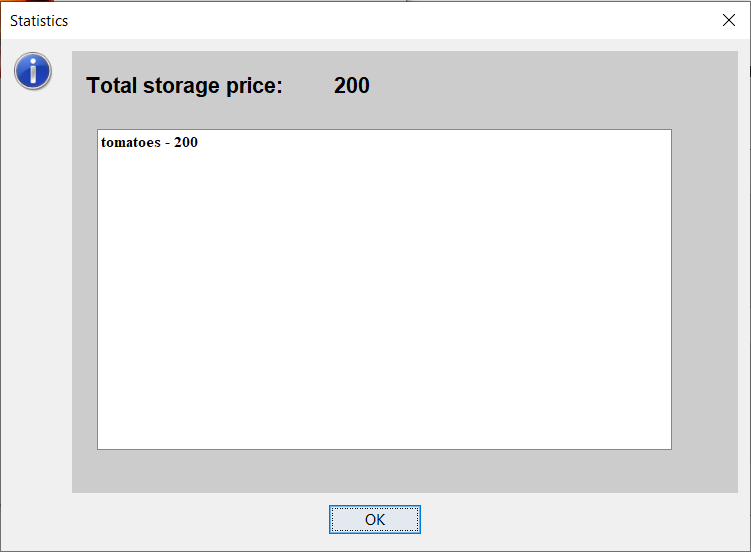
Збережемо нашу таблицю і повернемося до основного вікна.

Виберемо “write-off”. Система відкриє форму. У полі товару підтримується автодоповнення. Все інше думаю і так зрозуміло.

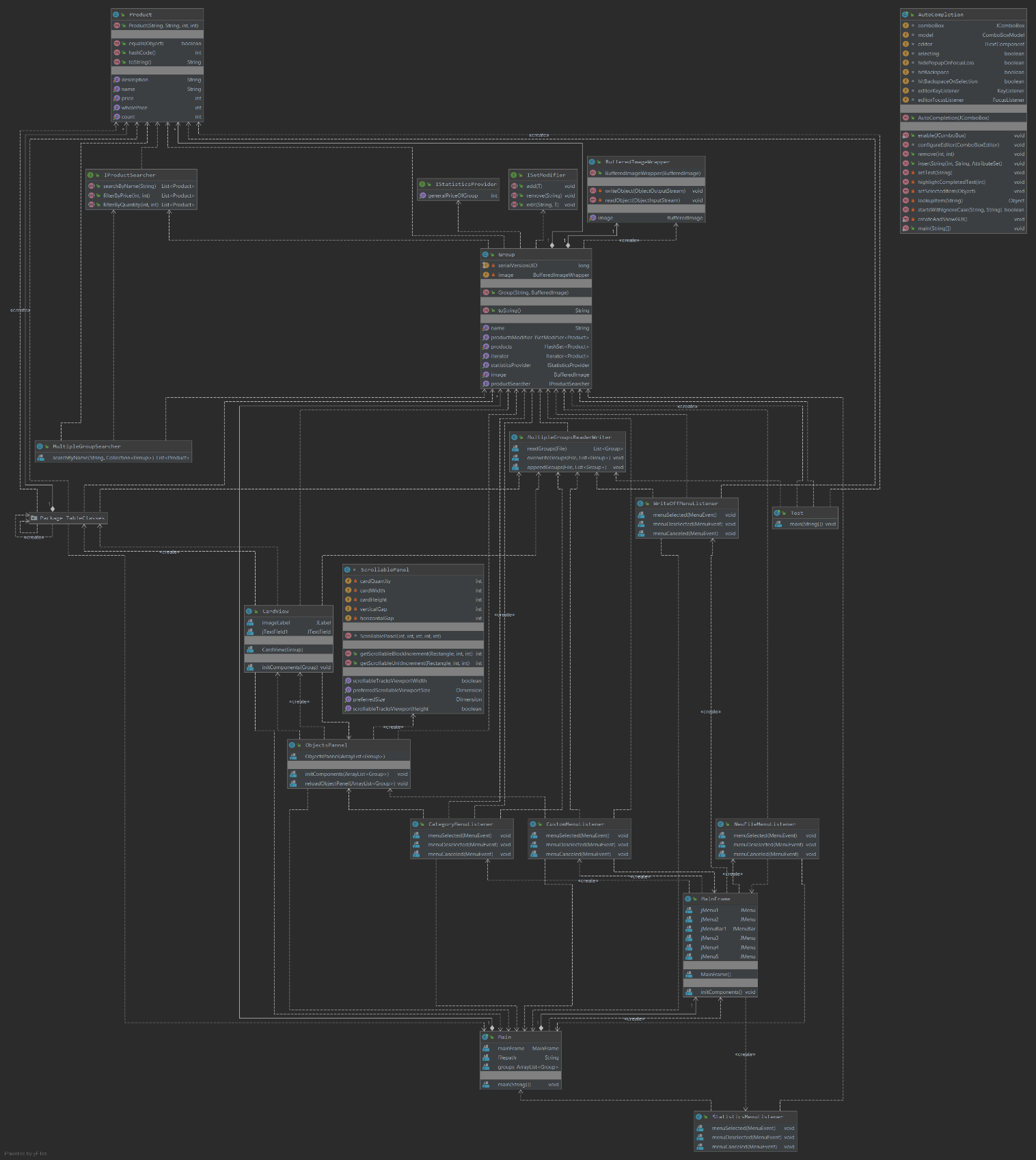




І останнє що залишилося це вікно зі статистикою.



Тут можна побачити загальну вартість всієї бази та кожної з груп.

**4.Структура програми з описом усіх написаних класів**

Як вже зазначалося вище наш проект поділений на два пакети: model та view. В пакеті model за збереження даних відповідають Product і Group що містить у собі HashSet<Product>. Також є два сервіси: “MultipleGroupReaderWriter” та “MultipleGriupSearcher” і обгортка класу “BufferedImage” – “BufferedImageWrapper”, основною метою якої є додати можливість серіалізації. Інтерфейси імплементуються внутрішніми класами Group. Так зроблено щоб код відповідав першому принципу специфікації “SOLID” – “Single responsibility principle”. Загалом, структура програми досить прямолінійна тому мету кожної функції і класу досить легко можна зрозуміти з назви.

**5.Проблеми в роботі.**

Ми виділяємо 2 основні проблеми. Перша – написання GUI. API в порівнянні наприклад з відповідним з браузерів досить погано зроблено. Слід відзначити клас “AbstractTableModel”. Через його дизайн використання його у парі з колекцією що не імплементує “List” спричиняє появу великої кількості так званого “boilerplate code”. Єдине що гріє душу це конструктор вбудований в “Intellij Idea”. Інша проблема – серіалізація. Як вже було зазначено у 4 пункті, був створений клас “BufferedImageWrapper” для того щоб додати можливість серіалізації зображень. Досить дивним є то що інтерфейс “Serializable” не має методів відповідальних за серіалізацію, тому потрібно шукати в документації. Потрібні методи

“private void writeObject(ObjectOutputStream out) throws IOException”

“private void readObject(ObjectInputStream in) throws IOExceptoin”.

Слід зазначити що вони не помічаються анотаціями і явно не викликаються, тому можна і не відразу зрозуміти чому взагалі потрібні ці методи.

**6. Висновки**

У ході роботи ми виконали поставлені нам задачі та переконалися у правильності слів лектора щодо зручності користування Swing.