

8. Основи введення/виведення Java SE

Мета: Оволодіння навичками управління введенням/виведенням даних з використанням класів платформи Java SE.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кулик Данііл Ігорович
- НТУ “ХП” 1.KIT102.8a
- Варіант 11

1.2 Загальне завдання

1. Забезпечити можливість збереження і відновлення масива об’єктів рішення завдання лабораторної роботи №7.
2. Забороняється використання стандартного протокола серіалізації.
3. Продемонструвати використання моделі Long Term Persistence.
4. Забезпечити діалог з користувачем у вигляді простого текстового меню.
5. При збереженні та відновленні даних забезпечити діалоговий режим вибору директорії з відображенням вмісту і можливістю переміщення по підкаталогах.

1.3 Задача

11. Прикладна галузь: Магазин. Запис в каталозі товарів: найменування; одиниця виміру; кількість; ціна одиниці; дата надходження; опис (необмежений набір характеристик у вигляді “властивість, значення”).

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі присутні об’єктно-орієнтовані методи:

Інкапсуляція – захист даних від неправомірного користування.

2.2 Ієрархія та структура даних

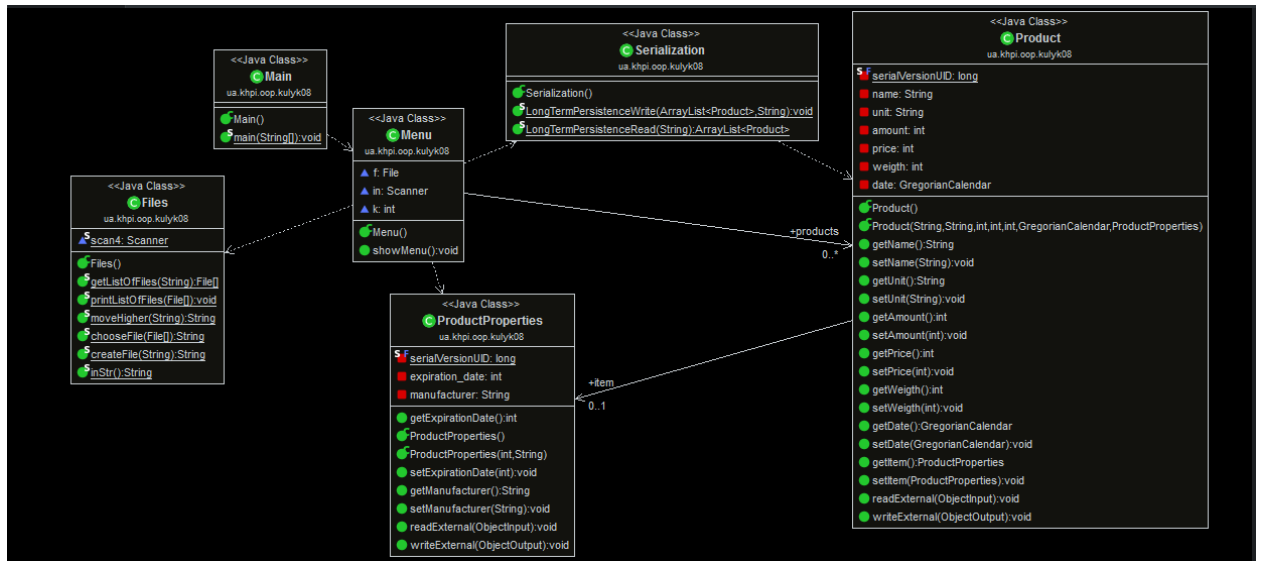


Рисунок 1 – Діаграма класів

2.3 Важливі фрагменти програми

```
@Override
public void readExternal(ObjectInput in) throws IOException, ClassNotFoundException {
    setName((String)in.readObject());
    setUnit((String)in.readObject());
    setAmount(in.readInt());
    setPrice(in.readInt());
    setWeight(in.readInt());
    setDate((GregorianCalendar)in.readObject());
    setItem((ProductProperties)in.readObject());
}

@Override
public void writeExternal(ObjectOutput out) throws IOException {
    out.writeObject(getName());
    out.writeObject(getUnit());
    out.writeObject(getAmount());
    out.writeObject(getPrice());
    out.writeObject(getWeight());
    out.writeObject(getDate());
    out.writeObject(getItem());
}
```

Рисунок 2 – Реалізація нестандартного протоколу серіалізації та десеріалізації

```

public static void LongTermPersistenceWrite(ArrayList<Product> object,String path) throws FileNotFoundException {

    XMLEncoder encoder = new XMLEncoder(
        new BufferedOutputStream(
            new FileOutputStream(path)));

    encoder.writeObject(object);
    encoder.close();
}

@SuppressWarnings("unchecked")
public static ArrayList<Product> LongTermPersistenceRead(String path) throws FileNotFoundException {
    XMLDecoder decoder = new XMLDecoder(
        new BufferedInputStream(
            new FileInputStream(path)));

    ArrayList<Product> object = (ArrayList<Product>) decoder.readObject();
    decoder.close();
    return object;
}

```

Рисунок 4 – Реалізація моделі Long Term Persistence

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє створювати об'єкти – товари – , що заносяться у запис каталогу, тобто створюється масив об'єктів. Користувач може додавати об'єкти до масиву, видаляти елементи вибірково, а також очистити увесь масив одним викликом відповідної кнопки меню. Також присутня можливість серіалізувати/десеріалізувати об'єкти з файлу.

```

Каталог товаров
1. Список товаров
2. Добавить товар в список
3. Убрать товар из списка
4. Очистить весь список
5. Сохранить данные в файл
6. Восстановить данные из файла
7. Выбрать директорию для хранения файла
8. Завершить работу

Выберите опцию:
1
1 товар:
Название: Tomatoes
Единица измерения: kg
Количество: 1200
Цена: 24
Вес: 100
Дата поступления: Fri Dec 20 12:00:00 EET 2019
Срок годности: 15 дней
Производитель: Lakad Matatag

```

Рисунок 3 – Результати виведення масиву об'єктів

```

<?xml version="1.0" encoding="UTF-8"?>
- <java class="java.beans.XMLDecoder" version="1.8.0_221">
  - <object class="java.util.ArrayList">
    - <void method="add">
      - <object class="ua.khpi.oop.kulyk08.Product" id="Product0">
        - <void property="amount">
          <int>1200</int>
        </void>
        - <void property="date">
          - <void property="time">
            - <object class="java.util.Date">
              <long>1576836000000</long>
            </object>
          </void>
        </void>
        <void id="ProductProperties0" property="item"/>
        - <void property="item">
          <object idref="ProductProperties0"/>
        </void>
        - <void property="name">
          <string>Tomatoes</string>
        </void>
        - <void property="price">
          <int>24</int>
        </void>
        - <void property="unit">
          <string>kg</string>
        </void>
        - <void property="weighth">
          <int>100</int>
        </void>
      </object>
    </void>
  </object>
  - <void class="ua.khpi.oop.kulyk08.Product" method="getField" id="Field0">
    <string>item</string>
    - <void method="get" id="ProductProperties1">
      <object idref="Product0"/>
      - <void property="manufacturer">
        <string>Lakad Matatag</string>
      </void>
      - <void property="expirationDate">
        <int>15</int>
      </void>
    </void>
  </void>
  <void idref="ProductProperties1"/>
</java>

```

Рисунок 4 – Зміст файлу *Test.xml*

ВИСНОВКИ

В даній лабораторній роботі розробив та реалізував класи та методи відповідно прикладної галузі, реалізував управління масивом domain-об'єктів, а також забезпечив та продемонстрував коректне відображення кирилиці.