

10. Обробка параметризованих контейнерів

Мета: Розширення функціональності параметризованих класів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кулик Данііл Ігорович
- НТУ “ХП” КІТ118-в
- Варіант 11

1.2 Загальне завдання

Використовуючи програму рішення завдання лабораторної роботи №9:

1. Розробити параметризовані методи (Generic Methods) для обробки колекцій об'єктів згідно прикладної задачі.
2. Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.
 - Автоматичний режим виконання програми задається параметром командного рядка **-auto**. Наприклад, java ClassName -auto.
 - В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.
3. Забороняється використання алгоритмів з Java Collections Framework.

1.3 Задача

11. Прикладна галузь: Магазин. Сортування за найменуванням товару, за ціною одиниці, за датою надходження.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Були розроблені додаткові функції для класу-контейнера. Розроблене діалогове меню та можливість зчитування даних з файлу.

2.2 Ієрархія та структура даних

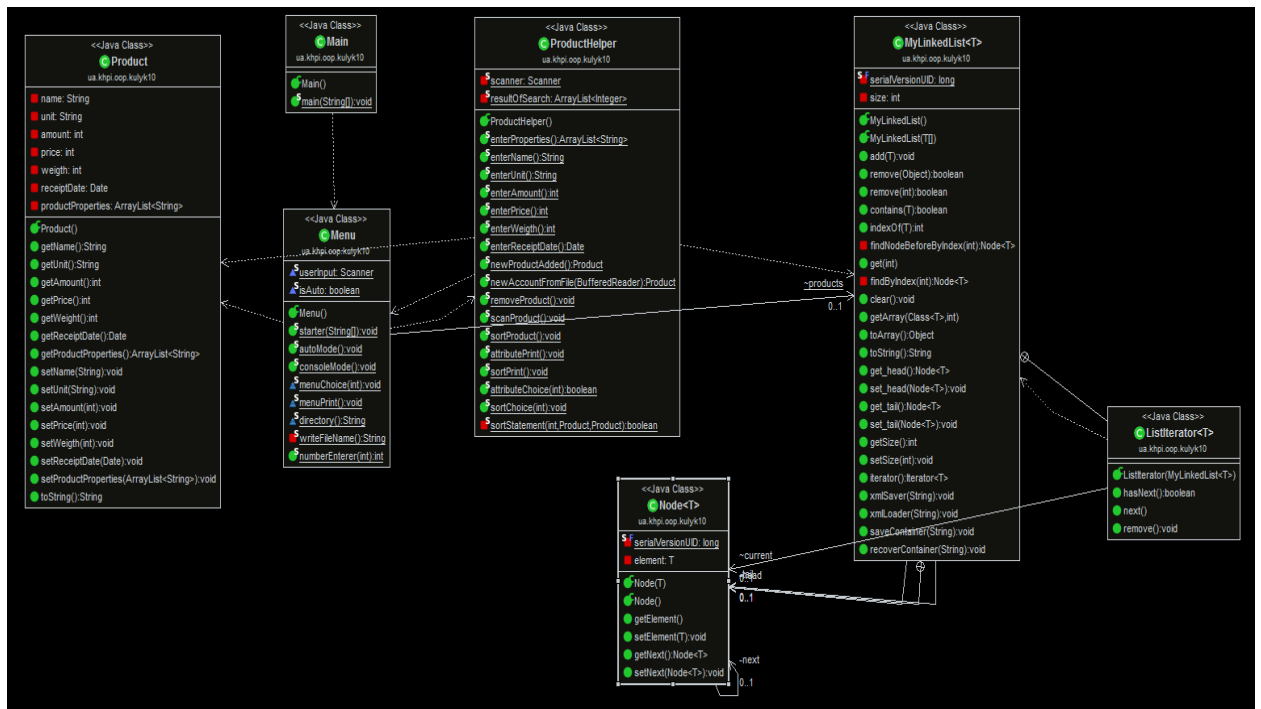


Рисунок 1 – Діаграма класів

2.3 Важливі фрагменти програми

```
static public void autoMode() {
    try {
        FileReader filereader = new FileReader(directory());
        BufferedReader reader = new BufferedReader(filereader);
        String line = reader.readLine();
        int size = Integer.parseInt(line);
        int index = 0;
        while (index++ < size) {
            products.add(ProductHelper.newAccountFromFile(reader));
        }
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    consoleMode();
}
```

Рисунок 2 – Автоматичний режим виконання

```

static void menuChoice(int numMenu) {
    switch (numMenu) {
        case 1 :
            if(isAuto)
                System.out.println("Sorry, in auto mode this function is not available");
            else
                products.add(ProductHelper.newProductAdded());
            break;
        case 2 :
            if(products.getSize() == 0) System.out.println("Enter data first.");
            else System.out.println(products.toString());
            break;
        case 3 :
            if(products.getSize() == 0) System.out.println("Enter data first.");
            else ProductHelper.removeProduct();
            break;
        case 4 :
            if(products.getSize() == 0) System.out.println("Enter data first.");
            else products.clear();
            break;
        case 5 :
            if(products.getSize() == 0) System.out.println("Enter data first.");
            else ProductHelper.scanProduct();
            break;
        case 6 :
            if(products.getSize() == 0) System.out.println("Enter data first.");
            else products.xmlSaver(directory());
            break;
        case 7 :
            products.xmlLoader(directory());
            break;
        case 8 :
            if(products.getSize() == 0) System.out.println("Enter data first.");
            else
                try {

```

Рисунок 3 – Консольне меню

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.

```
Good afternoon!
0 - Exit
1 - Add product
2 - Show all products
3 - Remove product
4 - Remove all products
5 - Scan account
6 - Use XML encoder
7 - Use XML decoder
8 - Serialize data
9 - Deserialize data
10 - Sort data
[0] User> 1
Enter product name: Sausages
Enter product unit: kg
Enter amount of products: 100
Enter price of product: 120
Enter weighth of product: 2
Enter receipt date of product (in DD/MM/YYYY format): 21/05/2011
Enter product properties (click double Enter to stop adding):
[1] User> 2

                Name: Sausages
                Unit:  kg
                Amount: 100
                Price:  120
                Weigth:  2
                Receipt date: Sat May 21 00:00:00 EEST 2011
                Product properties: []

[1] User>
```

Рисунок 4 – Додавання елемента та демонстрація його існування

```

7 - Use XML decoder
8 - Serialize data
9 - Deserialize data
10 - Sort data
[0] User> 7
Enter file name: Test10.xml
1) .                2) ..                3) bin                4) src
5) ua
C:\Users\Daniil\eclipse-workspace\kulyk-daniil> 4
1) .                2) ..                3) kulyk05            4) kulyk06
5) kulyk07          6) ua
C:\Users\Daniil\eclipse-workspace\kulyk-daniil\src> 6
1) .                2) ..                3) khpi
C:\Users\Daniil\eclipse-workspace\kulyk-daniil\src\ua> 3
1) .                2) ..                3) oop
C:\Users\Daniil\eclipse-workspace\kulyk-daniil\src\ua\khpi> 3
1) .                2) ..                3) kulyk01            4) kulyk02
5) kulyk03          6) kulyk04          7) kulyk05            8) kulyk06
9) kulyk07          10) kulyk08         11) kulyk09           12) kulyk10

C:\Users\Daniil\eclipse-workspace\kulyk-daniil\src\ua\khpi\oop> 12
1) .                2) ..
C:\Users\Daniil\eclipse-workspace\kulyk-daniil\src\ua\khpi\oop\kulyk10> 1
[2] User> 2

      Name: Banana
      Unit:  kg
      Amount: 400
      Price:  60
      Weigth: 0
      Receipt date: Fri Sep 02 00:00:00 EEST 2011
      Product properties: []

      Name: Bread
      Unit:  bun
      Amount: 150
      Price:  15
      Weigth: 0
      Receipt date: Sun Dec 06 00:00:00 EET 2020
      Product properties: []

[2] User>

```

Рисунок 5 – Результати виконання десереалізації

```

<?xml version="1.0" encoding="UTF-8"?>
- <java class="java.beans.XMLDecoder" version="1.8.0_221">
  - <object class="ua.khpi.oop.kulyk10.MyLinkedList">
    - <void property="_head">
      - <object class="ua.khpi.oop.kulyk10.MyLinkedList$Node">
        - <void property="element">
          - <object class="ua.khpi.oop.kulyk10.Product">
            - <void property="amount">
              <int>400</int>
            </void>
            - <void property="name">
              <string>Banana</string>
            </void>
            - <void property="price">
              <int>60</int>
            </void>
            - <void property="receiptDate">
              - <object class="java.util.Date">
                <long>1314910800000</long>
              </object>
            </void>
            - <void property="unit">
              <string>kg</string>
            </void>
          </object>
        </void>
      - <void property="next">
        - <object class="ua.khpi.oop.kulyk10.MyLinkedList$Node" id="MyLinkedList$Node0">
          - <void property="element">
            - <object class="ua.khpi.oop.kulyk10.Product">
              - <void property="amount">
                <int>150</int>
              </void>
              - <void property="name">
                <string>Bread</string>
              </void>
              - <void property="price">
                <int>15</int>
              </void>
              - <void property="receiptDate">
                - <object class="java.util.Date">
                  <long>1607205600000</long>
                </object>
              </void>
              - <void property="unit">
                <string>bun</string>
              </void>
            </object>
          </void>
        </object>
      </void>
    </object>
  </void>
</object>

```

Рисунок 6 – Зміст файлу *Test10.xml*

ВИСНОВКИ

В даній лабораторній роботі розширив функціональності параметризованих класів: були додані режими консольного та автоматичного виконання програми, а також додані параметризовані методи.