

12. Регулярні вирази. Обробка тексту

Мета: Розширення функціональності параметризованих класів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кулик Данііл Ігорович
- НТУ “ХПІ” КІТ-118в
- Варіант 11

1.2 Загальне завдання

1. Використовуючи програми рішень попередніх задач, продемонструвати ефективне (оптимальне) використання регулярних виразів при вирішенні прикладної задачі.
2. Передбачити можливість незначної зміни умов пошуку.
3. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

1.3 Задача

11. Магазин. Знайти усі товари з актуальним терміном придатності. Дата виробництва, термін придатності (час зберігання або дата закінчення) можуть бути вказані в опису до товару.

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


```

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 product
6 - Use XML encoder
7 - Use XML decoder
8 - Serialize data
9 - Deserialize data
10 - Sort data
[0] User> 1
Enter product name: tomatoes
WARNING: Invalid name. Try again: TOMATOES
WARNING: Invalid name. Try again: 1345
WARNING: Invalid name. Try again: Tomatoes
Enter product unit: Kg
Enter amount of products: many
WARNING: Invalid name. Try again: 15
Enter price of product: 120
Enter weighth of product: 2
Enter receipt date of product (in DD/MM/YYYY format): 32/05/2008
Enter product properties (click double Enter to stop adding):
Day

[1] User> 2
                Name: Tomatoes
                Unit: Kg
                Amount: 15
                Price: 120
                Weigth: 2
                Receipt date: Sun Jun 01 00:00:00 EEST 2008
                Product properties: [Day]

[1] User>
```

Рисунок 4 – Приклад спрацьовування перевірки на валідацію даних

```

<?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 ознайомився з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.