Тема: Регулярні вирази. Обробка тексту.
Мета: Ознайомлення з принципами використання регулярних виразів для обробки тексту.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:
- Когутенко Олександр Олексійович;
- КІТ-119Д;
- 11 варіант.

1.2 Загальне завдання

1. Використовуючи програми рішень попередніх задач, продемонструвати ефективне (оптимальне) використання регулярних виразів при вирішенні прикладної задачі:

   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. Передбачити можливість незначної зміни умов пошуку.
3. Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Використовуеться наслідування, інтерфейс, поліморфізм.

2.2 Ієрархія та структура класів

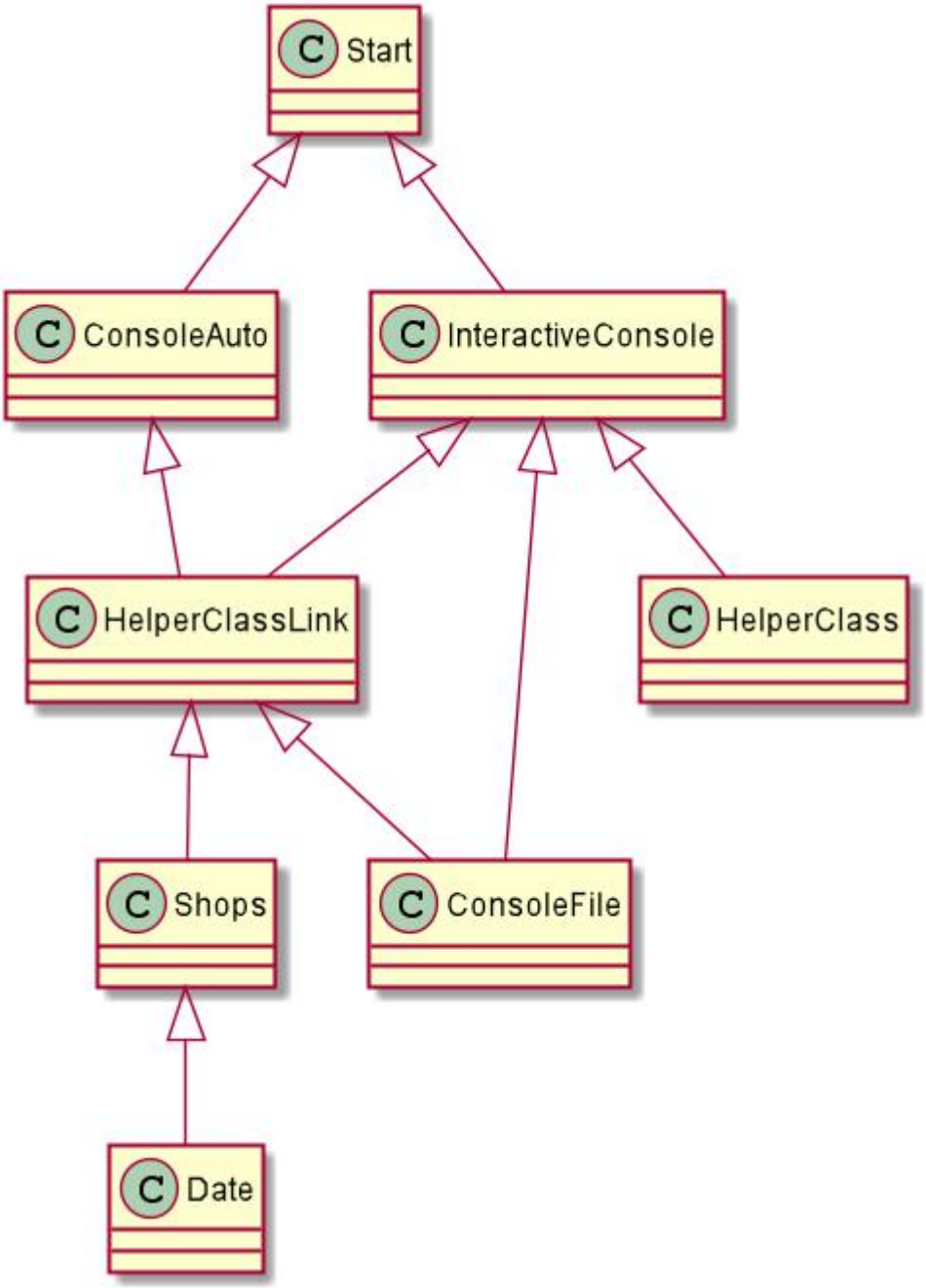

Рисунок 12.1 - иерархія класів

Використовую 7 классів: InteractiveConsole, Start, Date, Shops, ConsoleFile, HelperClassLink, ConoleAuto.

- InteractiveConsole клас для налагодженого спілкування програми з користувачем та методами нестандартних протоколів серіалізації.
- Start клас який має точку входу у програму.
- Date клас використовуется для збереження дати.
- Shops клас приклодної галузі.
- ConoleFile клас за допомогою якого користувач може наблюдати за відображенням вмісту каталогів.
- HelperClassLink використовуеться за для реалізації зв’язного списку з методами стандартних методів серіалізації.
- ConsoleAuto класс для автоматичної роботи із списком.

2.3 Важливі фрагменти програми

Доданно функцію відбору “свіжих” (свіжість яких не більше місяцю від сьогодні) товарів, але не розумію як реалізувати це через PB, тому виконав як зрозумів:

```
package ua.khpi.oop.kogutenko12;

import java.io.*;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

/**
 * The type Interactive console.
 */
public class InteractiveConsole {
   …
    /**
     * Sort helper class link.
     *
     * @param list the list
     * @return the helper class link
     */
    public HelperClassLink<Shops> sort(HelperClassLink<Shops> list) {
        System.out.println("list before:\n");
        System.out.println("\n---------------------------------\n");
        list.printList();
        System.out.println("\n---------------------------------\n");
        Shops[] shops = new Shops[list.size()];
        for (int i = 0; i < shops.length; i++) {
            shops[i] = list.get(i);
        }
        Integer field;
        while (true) {
            System.out.print("Enter field sorted (1 - name; 2 - price; 3 - date\n>>>");
            p = Pattern.compile("[123]");
            field = scanner.nextInt();
            m = p.matcher(field.toString());
            if (m.matches()) {
                break;
            } else {
```

```
            System.out.println("Enter info correctly!!!");
        }
    }

    //
    bubbleSort(shops, field);
    return new HelperClassLink<>(shops);
}

private void bubbleSort(Shops[] array, int field) {
    boolean sorted = false;
    while (!sorted) {
        sorted = true;
        for (int i = 1; i < array.length; i++) {
            if (compare(array[i], array[i - 1], field)) {
                swap(array, i, i - 1);
                sorted = false;
            }
        }
    }
}

private boolean compare(Shops a, Shops b, int field) {
    switch (field) {
        case 1:
            return a.getName().compareTo(b.getName()) >= 0;
        case 2:
            return a.getPrice() < b.getPrice();
        case 3:
            return (a.getDate().getYear() > b.getDate().getYear())
                    || (a.getDate().getYear() == b.getDate().getYear() &&
a.getDate().getMonth() > b.getDate().getMonth())
                    || (a.getDate().getYear() == b.getDate().getYear() &&
a.getDate().getMonth() == b.getDate().getMonth()
                    && a.getDate().getDay() > b.getDate().getDay());

    }
    return false;
}

private void swap(Shops[] array, int ind1, int ind2) {
    Shops tmp = array[ind1];
    array[ind1] = array[ind2];
    array[ind2] = tmp;
}
```

```java
    /**
     * Find fresh string.
     *
     * @return the string
     */
    public String findFresh() {
        DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
        java.util.Date date = new Date();
        System.out.println(dateFormat.format(date));
        String dateStr = dateFormat.format(date);
        String[] dateArr = dateStr.split("/");
        int currYear = Integer.parseInt(dateArr[0]), currMon =
Integer.parseInt(dateArr[1]), currDay = Integer.parseInt(dateArr[2]);
        String str = "";
        for (Shops shop : helperL) {
            int prodY = shop.getDate().getYear();
            int prodM = shop.getDate().getMonth();
            int prodD = shop.getDate().getDay();
            if (prodY == currYear) { // if year prod == curr year
                if (prodM == currMon) {
                    if (prodD >= currDay) {
                        str += shop.toString();
                    }
                } else if (currMon - prodM == 1 && prodD >= currDay) {
                    str += shop.toString();
                }
            } else if (prodY == currYear - 1 && prodM == 12 && prodD >= currDay)
{
                str += shop.toString();
            }
        }
        return str;
    }
}
```

3 ВАРІАНТИ ВИКОРИСТАННЯ

```
id: 1   | name: prod_1     | unit: obj  | count: 50       | price: 10       | date: 15.04.2021 | description: width - 50,height - 55,
id: 2   | name: prod_2     | unit: obj  | count: 60       | price: 20       | date: 16.04.2021 | description: width - 50,height - 55,
id: 3   | name: prod_3     | unit: obj  | count: 70       | price: 50       | date: 12.04.2021 | description: width - 50,height - 55,
id: 4   | name: prod_4     | unit: kg   | count: 90       | price: 60       | date: 01.05.2021 | description: width - 50,height - 55,
id: 5   | name: prod_5     | unit: l    | count: 80       | price: 80       | date: 11.02.2021 | description: width - 50,height - 55,
id: 6   | name: prod_6     | unit: l    | count: 30       | price: 190      | date: 21.02.2021 | description: width - 50,height - 55,
id: 7   | name: prod_7     | unit: kg   | count: 10       | price: 100      | date: 05.03.2021 | description: color - white,
id: 8   | name: prod_8     | unit: l    | count: 130      | price: 130      | date: 17.04.2021 | description: color - new_color,
id: 9   | name: prod_9     | unit: obj  | count: 50       | price: 10       | date: 15.03.2021 | description: width - 50,height - 55,
id: 10  | name: prod_10    | unit: obj  | count: 60       | price: 20       | date: 13.01.2021 | description: width - 50,height - 55,
```

Рисунок 12.2 - список всіх товарів

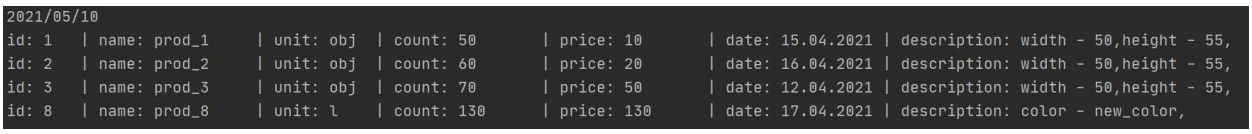

Рисунок 12.3 - вивід свіжих товарів

ВИСНОВКИ

Ознайомився з принципами використання регулярних виразів для обробки тексту.