

## **11. Регулярні вирази. Перевірка даних**

**Мета:** Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.

### **1 ВИМОГИ**

#### **1.1 Розробник**

Інформація про розробника:

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

#### **1.2 Загальне завдання**

Продемонструвати ефективне (оптимальне) використання регулярних виразів для перевірки коректності (валідації) даних, що вводяться, перед записом в domain-об'єкти відповідно до призначення кожного поля для заповнення розробленого контейнера:

- при зчитуванні даних з текстового файлу в автоматичному режимі;
- при введенні даних користувачем в діалоговому режимі.

#### **1.3 Задача**

**11.** Прикладна галузь: Магазин. Сортуння 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**

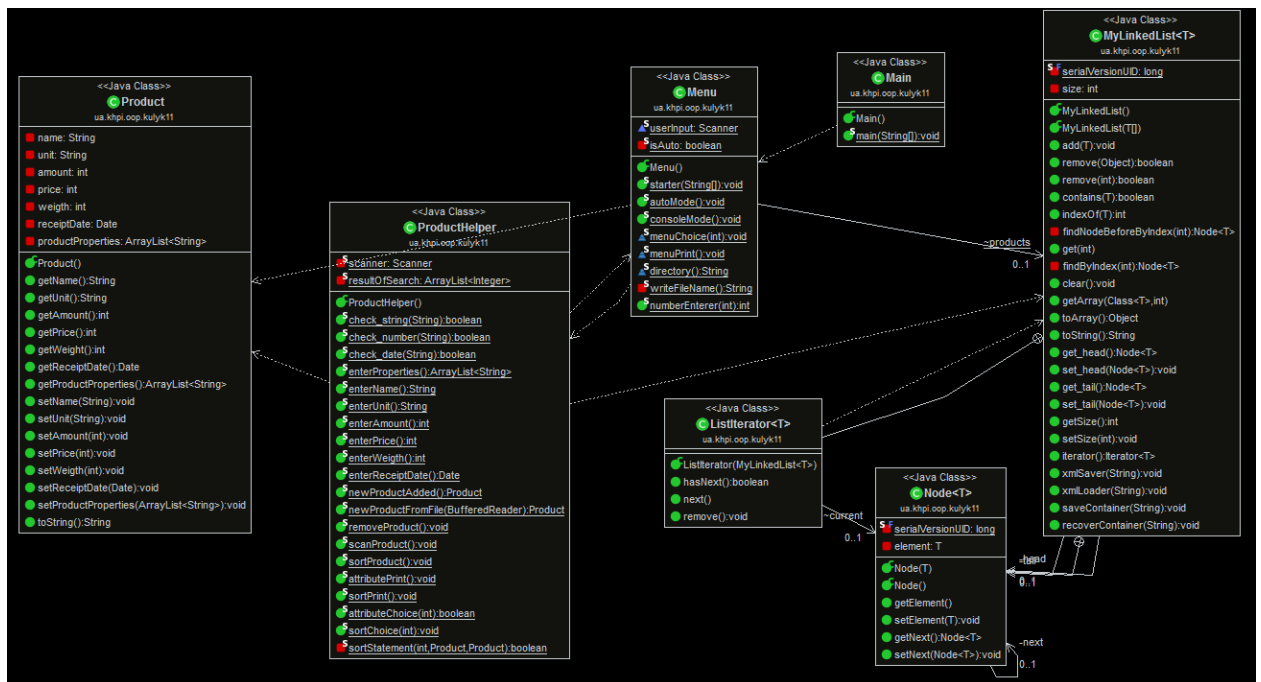


Рисунок 1 – Діаграма класів

## 2.3 Важливі фрагменти програми

```

public static boolean check_string(String str) {
    return str.matches("[A-Z][a-z]*");
}

public static boolean check_number(String number) {
    return number.matches("[1-9]?(\\d+)");
}

public static boolean check_date(String date) {
    return date.matches("\\d{2}/\\d{2}/\\d{4}");
}

```

Рисунок 2 – Регулярні вирази

```

public static int enterWeigth() {
    System.out.print("Enter weigth of product: ");
    String str = scanner.nextLine();
    while(!check_number(str)) {
        System.out.print("WARNING: Invalid name. Try again: ");
        str = scanner.nextLine();
    }
    int weigth = Integer.parseInt(str);
    return weigth;
}

public static Date enterReceiptDate() {
    System.out.print("Enter receipt date of product (in DD/MM/YYYY format): ");
    Date theReceiptDate = null;
    try {
        String date = scanner.nextLine();
        while(!check_date(date)) {
            System.out.print("WARNING: Invalid date. Try again: ");
            date = scanner.nextLine();
        }
        theReceiptDate = new SimpleDateFormat("ddMMyyyy").parse(date.replaceAll("/", ""));
    } catch (ParseException e) {
        System.out.println(e);
    }
    return theReceiptDate;
}
}

```

Рисунок 3 – Використання регулярних виразів у програмі

### 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

## ВИСНОВКИ

В даній лабораторній роботі ознайомився з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.