

11. Регулярні вирази. Перевірка даних

Мета: Розширення функціональності параметризованих класів.

1 ВИМОГИ

1.1 Розробник

Інформація про розробника:

- Кабак Олександр Русланович
- НТУ “ХП” 1.KIT102.8a
- Варіант 5

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 файла в автоматичному режимі;
- при введенні даних користувачем в діалоговому режимі.

1.3 Задача

5. Прикладна галузь: Довідник покупця. Торговельна точка: назва; адреса; телефони (кількість не обмежена); спеціалізація; час роботи (з зазначенням днів тижня).

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі присутні:

- 1) Двоб'язний список, що параметризується;
- 2) Збереження та відновлення об'єктів без протоколу серіалізації та з ним;
- 3) Спілкування з користувачем за допомогою меню та автоматичний режим для перегляду;
- 4) Регулярні вирази для коректного запису даних в базу.

2.2 Важливі фрагменти програми

```
private String regName() {
    String tmp = in.nextLine();
    while (!tmp.matches( regex: "(^[A-Z])(\\D)([a-zA-Z[ -]]+)" )) {
        System.out.println("Введите название корректно(с заглавной,без знаков(допускается \"-\"и \" \"))");
        tmp = in.nextLine();
    }
    for (Store tmpStore : stores) {
        if (tmp.equals(tmpStore)) {
            System.out.println("Данное название уже занято, попробуйте ещё раз");
            tmp = regName();
            break;
        }
    }
    return tmp;
}
```

Рис. 1 – Регулярний вираз для перевірки назви магазину

```
private String regSpecialization() {
    String tmp = in.nextLine();
    while (!tmp.matches( regex: "(^[A-Z])(\\D)([a-zA-Z[ -]]+)" )) {
        System.out.println("Введите название корректно(с заглавной,без знаков(допускается \"-\"и \" \"))");
        tmp = in.nextLine();
    }
    return tmp;
}
```

Рис. 2 - Регулярний вираз для перевірки назви спеціалізації магазину

```
private ArrayList<Integer> checkTelephone() {
    ArrayList<Integer> tmp = new ArrayList<>();
    int tmpInt;
    do {
        tmpInt = in.nextInt();
        if (tmpInt > 99999) {
            tmp.add(tmpInt);
        }
        System.out.println("Желаете добавить ещё один номер телефона?(Нажмите 0 если не желаете добавлять)");
    } while (in.nextLine().equals("0"));
    return tmp;
}
```

Рис. 3 - Перевірка для контактних телефонів магазину

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє створювати об'єкт – список магазинів, що заносяться у запис каталогу. Користувач може додавати інші магазини до списку, видаляти елементи вибірково, а також очистити весь масив одним викликом відповідної кнопки меню. Також присутня можливість серіалізувати /десеріалізувати об'єкти з файлу.

```
Справочник покупателя
1. Список торговых магазинов
2. Добавить торговый магазин
3. Убрать торговый магазин списка
4. Очистить весь список
5. Сортировка магазинов
6. Просмотр по номеру телефона
7. Запуск параллельного вычисления
8. Проверка по времени параллельного вычисления
0. Завершить работу

Выберите опцию:
2
Введите название корректно(с заглавной,без знаков(допускается "-"и " "))
PenCil

Введите название корректно(с заглавной,без знаков(допускается "-"и " "))
tationerys store
Stationery-Store
Введите адресс:
Pushkina 12
Введите номер телефона(Должен содержать больше 6 цифр): 759846
Желаете добавить ещё один номер телефона?(Нажмите 0 если не желаете добавлять)
```

Рис. 4 – Результаты створення об'єкта за допомогою регулярних виразів

Висновки: даній лабораторній роботі розроблено та реалізовано регулярні вирази для створення об'єкта списку *domain*-об'єктів за допомогою функцій класу *Regex* та самих регулярних виразів.