# Лабораторна робота №5.

**Мета:**

# Ознайомлення з принципами використання регулярних виразів для перевірки рядка на відповідність шаблону.

# Ознайомлення з принципами використання регулярних виразів для обробки тексту.

# Вимоги:

# 1. 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 в попередній роботі:

# ● при зчитуванні даних з текстового файла в автоматичному режимі;

# ● при введенні даних користувачем в діалоговому режимі.

# 2. Використовуючи програми рішень попередніх задач, продемонструвати ефективне (оптимальне) використання регулярних виразів при вирішенні практичного завдання з Прикладні задачі. Список №2. 20 варіантів.

# ● Передбачити можливість незначної зміни умов пошуку.

# ● Продемонструвати розроблену функціональність в діалоговому та автоматичному режимах.

# Розробник

Зінько Павло, КН-108

# Задача

12.Аудіотека. Композиція: назва; жанр; виконавець; текст (якщо є); дата створення; тривалість; формат даних; рейтинг (набір характеристик вигляду “властивість, значення”).

# Опис програми

Програма створює вашу власну аудіотеку за допомогою класу, можна додавати, чистити, виписувати, серіалізувати та десеріалізувати ваші аудіотеки у файл вибраний користувачем

# Засоби ООП

Були використані різні класи та методи, структури даних та модифікатори доступу.

# Ієрархія та структура класів

1. Клас Main, який викликає всі методи та класи створенні користувачем
2. Клас Album - domain-об’єкт
3. Клас LocationFounder повертає шлях збереження файлу та демонструє вміст
4. Клас Rating під-клас класу Album
5. Клас LinkedList спеціально розроблений за для виконання умови «Лабораторна №4». Представляє собою типовий зв’язний список.
6. Інтерфейс Generic параметризує LinkedList
7. Клас Node вузол для LinkedList
8. Клас Matcher клас для перевірки на коректність

# Важливі фрагменти програми.

**1) LocationFounder**

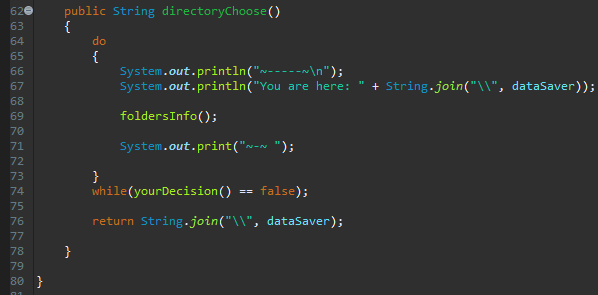
**2) LinkedList**

**3) Matcher**

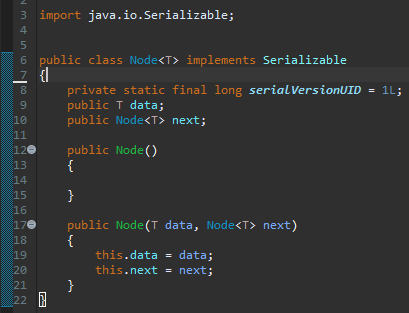
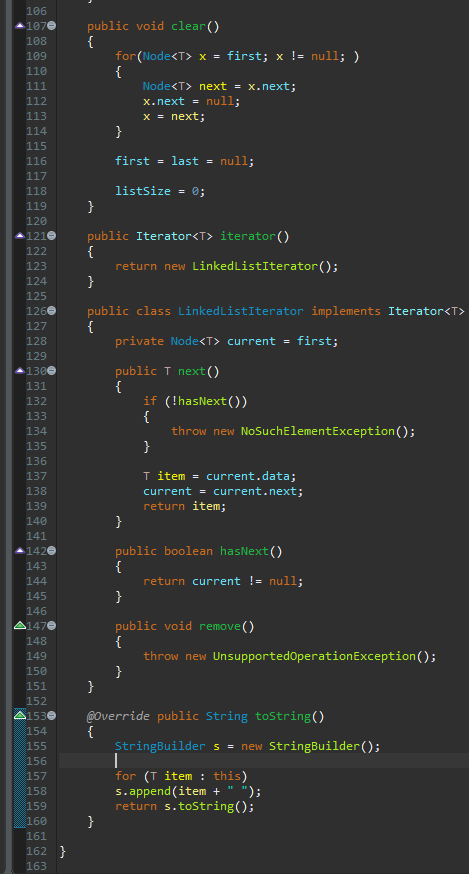
**New from laba3: sort, remove, search,LinkedList,Matcher added.**

LocationFounder

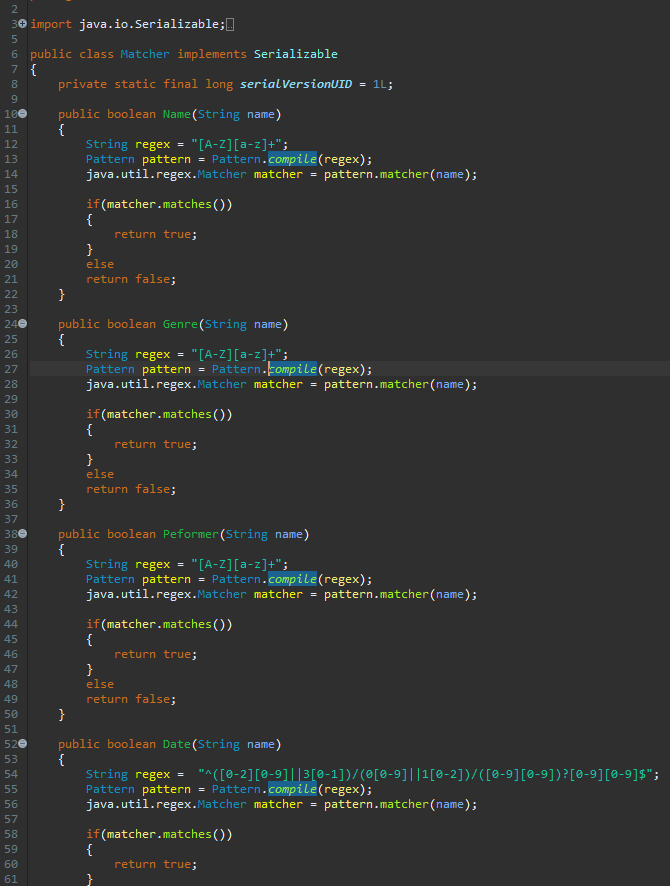
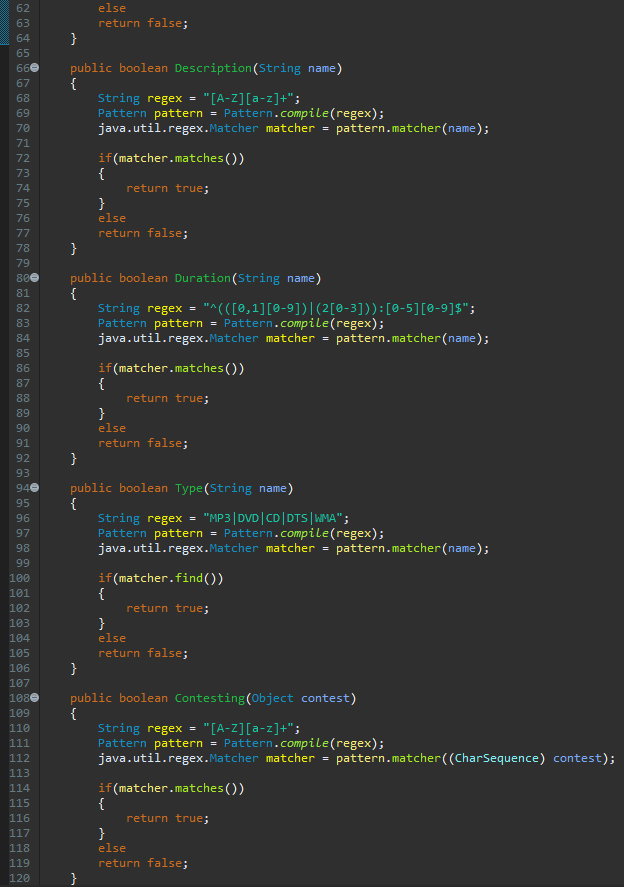


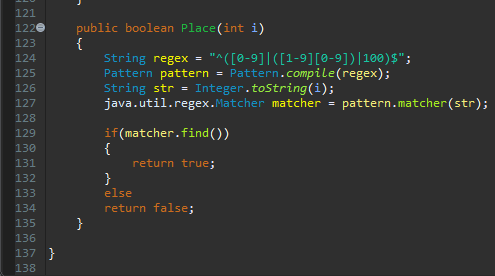


LinkedList with Node

Matcher



# 2.4 Варіанти використання

Можна використовувати для запису своїх аудіотек

# ВИСНОВКИ

У ході роботи ми ознайомились з принципами використання регулярних виразів для перевірки рядка та для обробки тексту