

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:

- Куйдін Михайло Андрійович
- НТУ “ХП” 1.KIT102.8a
- Варіант 10

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.
- Забороняється використання алгоритмів з Java Collections Framework.

1.3 Задача

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:

1. при зчитуванні даних з текстового файла в автоматичному режимі;
2. при введенні даних користувачем в діалоговому режимі.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

У даній програмі присутні об'єктно-орієнтовані методи: Інкапсуляція – захист даних від неправомірного користування та поліморфізм.

2.2 Ієрархія та структура даних

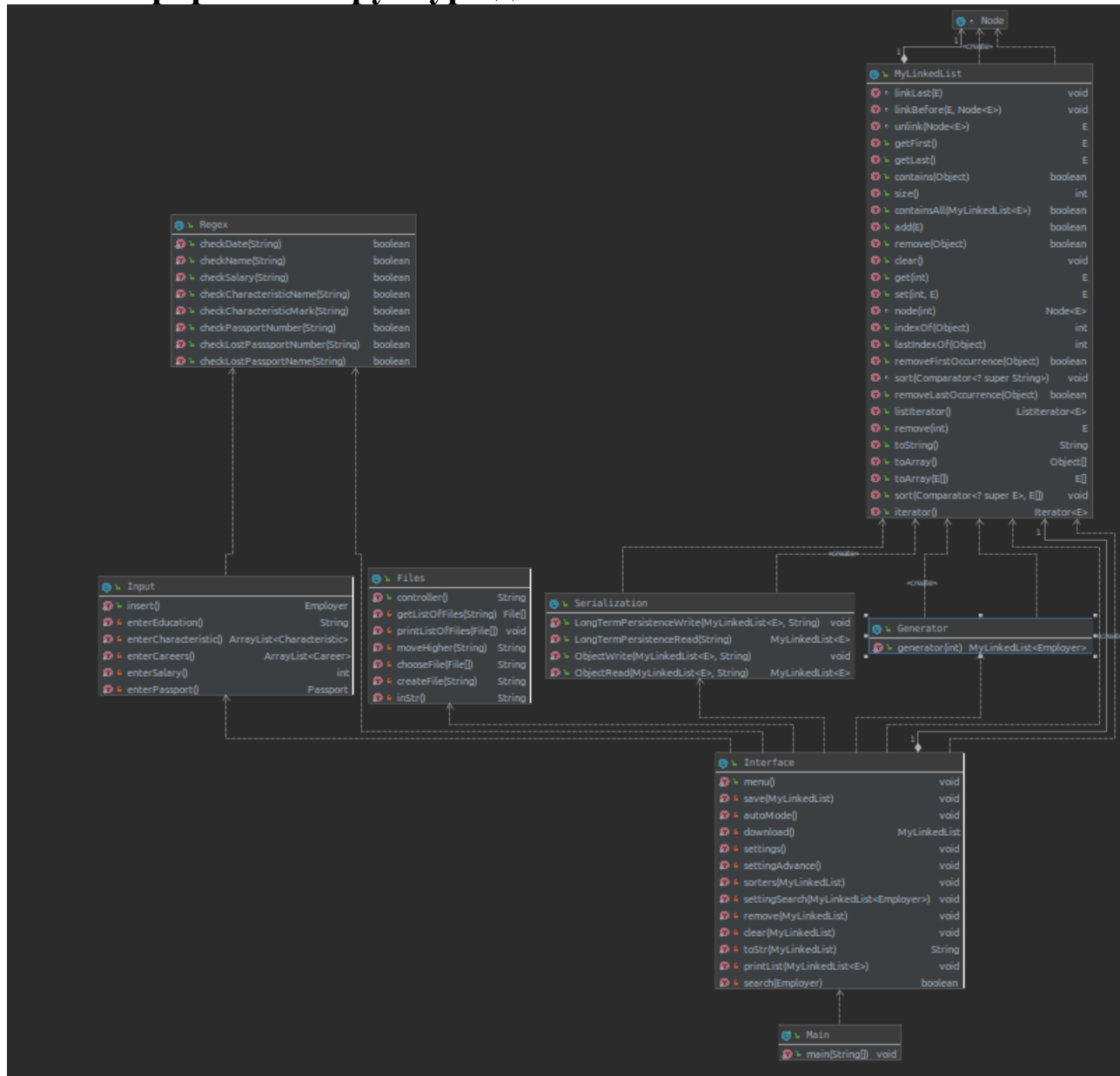


Рисунок 1 – Діаграма класів

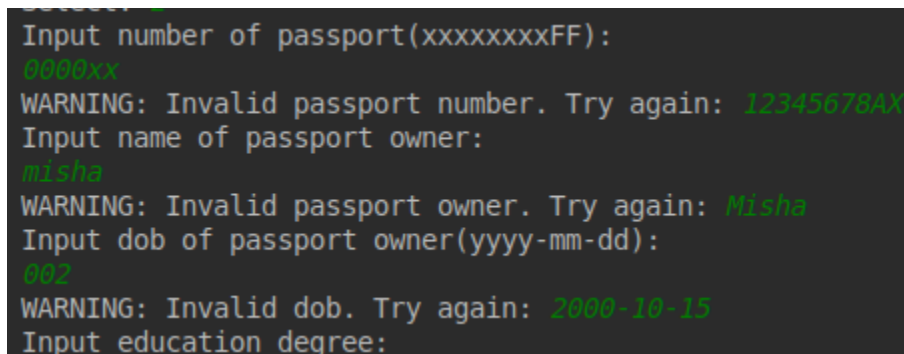
2.3 Важливі фрагменти програми

```
public static boolean checkDate(String input) {  
    return input.matches( regex: "^\\d{4}\\\\-(0?[1-9]|1[012])\\\\-(0?[1-9]|12)[0-9]|3[01])$");  
}  
  
public static boolean checkName(String input) {  
    return input.matches( regex: "[A-Z,A-Я][a-z,a-я]+");  
}  
  
public static boolean checkSalary(String input) {  
    return input.matches( regex: "\\d{1,10}");  
}  
  
public static boolean checkCharacteristicName(String input) {  
    return input.matches( regex: "[A-z\\\\s]+");  
}  
  
public static boolean checkCharacteristicMark(String input) {  
    return input.matches( regex: "[1-9][0-9]?|^100$");  
}  
  
public static boolean checkPassportNumber(String input) {  
    return input.matches( regex: "\\d{8}[A-Z]{2}");  
}  
  
public static boolean checkLostPasssपोर्टNumber(String input) {  
    return input.matches( regex: "\\d+8\\d+6\\d+\\\\wM");  
}  
  
public static boolean checkLostPassportName(String input) {  
    return input.matches( regex: "Ap[A-Я]+e[A-Я]+");  
}
```

Рисунок 2 — регулярні вирази

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма дозволяє створювати об'єкти – “записи в розкладі”, 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. Якщо програма починає свою роботу з параметром “-auto\а”, то перший крок програми зчитування з файлу.



```
Input number of passport(xxxxxxxxFF):  
0000xx  
WARNING: Invalid passport number. Try again: 12345678AX  
Input name of passport owner:  
misha  
WARNING: Invalid passport owner. Try again: Misha  
Input dob of passport owner(yyyy-mm-dd):  
002  
WARNING: Invalid dob. Try again: 2000-10-15  
Input education degree:
```

Рисунок 3 — використання регулярних виразів

ВИСНОВКИ

В даній лабораторній роботі було розроблено та реалізовано класи та методи відповідно прикладної галузі, реалізував управління списком domain-об'єктів, а методи сортування та обробка початкових параметрів.