

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут прикладної математики і фундаментальних наук

Кафедра прикладної математики



Лабораторна робота №8
з курсу “Програмування настільних
застосунків”
Тема: “Оброблення текстової інформації в Java з використанням
регулярних виразі”

Виконав студент групи ПМ-33
Венгринюк Олег
Прийняла
Терендій О. В.

Завдання для лабораторної роботи:

1. Ознайомитись з теоретичним матеріалом.
2. Дати відповідь на поставлені питання.
3. Розробити алгоритм розв'язання індивідуального завдання.
4. Скласти програму на мові Java у відповідності з розробленим алгоритмом.
5. Виконати обчислення по програмі.

Перелік контрольних запитань:

1. Наведіть основні переваги використання регулярних виразів?
2. Наведіть основні елементи мови регулярних виразів.
3. Що таке квантифікатори? Наведіть приклад використання квантифікаторів у регулярних виразах.
4. Які оператори пунктуації ви знаєте? Наведіть приклад використання операторів пунктуації в регулярних виразах.
5. Що таке групи регулярних виразів? Наведіть приклад задачі, в якій можливо використати групи?

Відповіді на контрольні запитання:

1. Регулярні вирази – потужний інструмент для пошуку та редагування тексту. Регулярні вирази маю визначені правила, що описують послідовність символів в тексті. Регулярні вирази якісно і зрозуміло описують шаблони. Також регулярні вирази існують не тільки в Java і є широко розповсюджені та є багато прикладів як їх використовувати.
2. Регулярні вирази мають синтаксис їх побуди. Він заснований на наступних символах: `<([{\^-= $!|}])? * + . >` їх можна комбінувати з літерами та числами а також знаками пунктуації.
 - Є метасимволи для пошуку границь стрічок такі як: `^` - початок стрічки, `$` - кінець стрічки та інші.
 - Метасимволи пошуку символічних класів `\d` – цифровий символ, `\D` - нецифровий, `\s` – знак пробілу, `\S` – знак всіх символів окрім пробілу, `\w` – знак букв, цифр та знаку підкреслення, `\W`- заперечення до `\w`, `.` (крапка) – будь-який символ.
 - Метасимволи символів редагування: `\t` – знак табуляції, `\n` – знак переносу каретки на новий рядок, `\r` - знак повернення каретки на початок, `\f` – знак переходу на нову сторіку та інші
 - Метасимволи групування `[]` -об'єднання вказує на те що може бути будь-який знак що містить група, `[^]` - вказує на те, що може бути будь-який знак, що не є в об'єднанні, `[][]` - об'єднання об'єднань, вказує на те що може бути символ з групи або з підгрупи, `[&&]` - об'єднання перетину, вказує на те що може бути будь-який символ з перетину.
 - Метасимволи квантифікатор. `?` - один або 0 символів `*` - довільна кількість `+` - один або більше разів символ може зустрічатись `{n, }` - n або більше разів символ може зустрічатись, `{n,m}` – символ може зустрічатись від n до m разів.
3. Квантифікатори – спеціальний символ у регулярних виразах впроваджений для спрощення формування послідовностей символів. Є декілька квантифікаторів: `?`, `+`,

`*`, `{}`. Також квантифікатори можуть бути у 3 режимах – жадний, наджадний, лінійний. Режим квантифікатора визначає правила пошуку підстрічки в тексті.

`\\d{5}` пошук 5 послідовних цифр у тексті. Квантифікатори зручно використовувати для пошуку слів із заданою довжиною або за умови що символ може бути пропущений.

4. `\t` – табуляцій, `\n` – переходу на новий рядок, `\r` – повернення каретки на початок, `\f` – нова сторінка, `\u 0085` – символ наступної стріки, `\u 2028` – розділення рядків, `\u 2029` – розділення абзаців
5. Групи в регулярних виразах це спеціальний синтаксис для опису можливих символів на місці одного. Пошук букв і цифр в довільному порядку. Пошук послідовності букв але за виключенням деяких. Наприклад пошук номерних знаків. Пошук назв компаній.

Програмний код

```
import java.util.Scanner;
import java.util.regex.Pattern;
import java.util.regex.Matcher;
public class Main {
    static String zipRuleUsa = "\\b\\d{5}-?\\d{4}\\b";
    static String zipRulePoland = "\\b\\d{2,2}-\\d{3,3}\\b";
    static Pattern patternUsa = Pattern.compile(Main.zipRuleUsa);
    static Pattern patternPol = Pattern.compile(Main.zipRulePoland);
    public static void main(String[] args){
        System.out.print("Enter a string : ");
        Scanner scanner = new Scanner(System.in);
        String inputString = scanner.nextLine();
        // Debug line
        // String inputString = "12345-6789 12-111";
        System.out.println("USA ZIP code");
        findAndOutput(inputString, Main.patternUsa);
        System.out.println("Poland ZIP code");
        findAndOutput(inputString, Main.patternPol);
    }
    static void findAndOutput(String text, Pattern pattern){
        Matcher matcher = pattern.matcher(text);
        while (matcher.find()) {
            System.out.println(text.substring(matcher.start(), matcher.end()));
        }
    }
}
```

Результат виконання:

```
Enter a string : 1111-4444 11-333 111111111 some text between 91-123
USA ZIP code
111111111
Poland ZIP code
11-333
91-123
```

Висновок: в ході виконання даної лабораторної роботи було освоє регулярні вирази у мові програмування Java. Було виконано поставле завдання.