

ЛАБОРАТОРНА РОБОТА № 9

Регулярні вирази. Рефлексія. Анотації.

Мета: практика роботи з регулярними виразами, використання рефлексії, створення власних анотацій.

Хід роботи:

Завдання 1. Створити консольний Java проект java_lab_9 з пакетом com.education.ztu.

Завдання 2. Робота з регулярними виразами:

- Використати власний текст, що містить дані 5-10 співробітників компанії (ПІБ, вік, посада, досвід роботи, адреса, емайл, телефон і т. д.)
- Знайти в тексті всі номери телефонів та емайли.
- Змінити формати відображення дат народження (наприклад: 20.05.1995 на 1995-05-20)
- Змінити посади кільком співробітникам.
- Результати роботи відобразити в консолі.

Завдання 3. Робота з користувачьким класом методами Reflection API:

- Створити власний клас в якому міститимуться публічні та приватні поля, конструктори і методи з аргументами та без.
- Отримати об'єкт класу Class для користувачького класу трьома способами.
- Отримати всі поля, методи, конструктори, що визначені тільки в цьому класі (не враховувати наслідування) та вивести ці значення в консоль (назву, типи параметрів та значення, що повертається).
- Створити екземпляр класу.
- Викликати метод класу.
- Попрацювати з приватним полем (встановити та отримати значення)
- Результати роботи відобразити в консолі.

Завдання 4. Створення власної анотації:

- Створити власну анотацію, задати їй необхідні поля та значення за замовчуванням для них.
- Встановити їй обмеження застосування через анотацію @Target
- Встановити їй політику утримання через анотацію @Retention
- Додати анотацію до відповідного об'єкту в коді.
- Отримати дані анотації з об'єкту та вивести в консоль.

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000 – Пр9		
Розроб.	Iuciuk Ol.C.				Lім.	Арк.	Аркушів
Перевір.	Піонтківський В.І.					1	13
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					ФІКТ Гр. ІПЗ-23-1		

Лістинг програми:

```

package com.education.ztu;
import java.lang.annotation.*;
import java.lang.reflect.*;
import java.util.regex.*;

public class Main {
    public static void main(String[] args) throws Exception {
        System.out.println("== TASK 2: Регулярні вирази ==\n");
        String text = sampleEmployeesText();
        System.out.println("Оригінальний текст:\n" + text);
        Pattern phonePattern = Pattern.compile(
            "(?:\\+\\d{1,3}[\\s-]?)(?:\\(\\d{2,5}\\)[\\s-]?|\\d{2,5}[\\s-"
        ])?)\\d{3,4}[\\s-]?\\d{2,2}[\\s-]?\\d{2,2}");
        Matcher mPhone = phonePattern.matcher(text);
        System.out.println("\nЗнайдені телефони:");
        while (mPhone.find()) System.out.println(" - " + mPhone.group());
        Pattern emailPattern = Pattern.compile(
            "[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.\\.[a-zA-Z]{2,}");
        Matcher mEmail = emailPattern.matcher(text);
        System.out.println("\nЗнайдені email:");
        while (mEmail.find()) System.out.println(" - " + mEmail.group());
        Pattern datePattern =
Pattern.compile("(\\b(\\d{2})\\.\\.(\\d{2})\\.(\\d{4}))\\b)");
        String textWithDatesReformatted = datePattern.matcher(text)
            .replaceAll("$4-$3-$2"); // yyyy-MM-dd
        System.out.println("\nТекст з переформатованими датами:\n" +
textWithDatesReformatted);
        String textPositionsUpdated = textWithDatesReformatted
            .replaceAll("\\bJunior Developer\\b", "Middle Developer")
            .replaceAll("\\bIntern\\b", "Junior Developer");
        System.out.println("\nТекст з оновленими посадами:\n" +
textPositionsUpdated);
        System.out.println("\n== TASK 3: Reflection API ==\n");
        Class<Staff> c1 = Staff.class;
        Class<?> c2 = new Staff().getClass();
        Class<?> c3 = Class.forName("com.education.ztu.Staff");
        System.out.println("Class via .class: " + c1.getName());
        System.out.println("Class via instance.getClass(): " + c2.getName());
        System.out.println("Class via Class.forName: " + c3.getName());
        System.out.println("\nDeclared fields:");
        for (Field f : c1.getDeclaredFields()) {
            System.out.printf(" - %s %s\n", f.getType().getSimpleName(),
f.getName());
        }
        System.out.println("\nDeclared constructors:");
        for (Constructor<?> ctor : c1.getDeclaredConstructors()) {
            System.out.print(" - " + ctor.getName() + "(");
            Class<?>[] params = ctor.getParameterTypes();
            for (int i = 0; i < params.length; i++) {
                System.out.print(params[i].getSimpleName() + (i < params.length-1
? ", " : ""));
            }
            System.out.println(")");
        }
        System.out.println("\nDeclared methods:");
        for (Method mm : c1.getDeclaredMethods()) {
            System.out.print(" - " + mm.getName() + "(");
            Class<?>[] p = mm.getParameterTypes();
            for (int i=0;i<p.length;i++) System.out.print(p[i].getSimpleName() +
(i<p.length-1? ", ":""));
            System.out.println(" : " + mm.getReturnType().getSimpleName());
        }
    }
}

```

		Ilyuk Ol.C.			Арк.
		Плюнгівський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	2

```

        }
        Constructor<Staff> ctor = c1.getDeclaredConstructor(String.class,
int.class);
        Staff staffInstance = ctor.newInstance("Ivan Petrenko", 29);
        System.out.println("\nСтворено об'єкт Staff: " + staffInstance);
        Method greet = c1.getDeclaredMethod("greet", String.class);
        greet.invoke(staffInstance, "Доброго дня");
        Field secretField = c1.getDeclaredField("secretNote");
        secretField.setAccessible(true);
        secretField.set(staffInstance, "Це приватна нотатка через reflection");
        String secretValue = (String) secretField.get(staffInstance);
        System.out.println("Отримано приватне поле secretNote: " + secretValue);
        Method privateMethod = c1.getDeclaredMethod("secretMethod");
        privateMethod.setAccessible(true);
        String secretMethodResult = (String) privateMethod.invoke(staffInstance);
        System.out.println("Результат виклику приватного методу: " +
secretMethodResult);
        System.out.println("\n==== TASK 4: Власна анотація ===\n");
        MyAnnotation ann = c1.getAnnotation(MyAnnotation.class);
        if (ann != null) {
            System.out.println("Анотація MyAnnotation присутня на Staff:");
            System.out.println(" - author(): " + ann.author());
            System.out.println(" - version(): " + ann.version());
            System.out.print(" - tags(): ");
            for (String t : ann.tags()) System.out.print(t + " ");
            System.out.println();
        } else {
            System.out.println("Анотація MyAnnotation відсутня.");
        }
        System.out.println("\n==== Кінець роботи програми ===");
    }
    private static String sampleEmployeesText() {
        return String.join("\n",
                "1) Іваненко Петро, 20.05.1995, Junior Developer, 2 years, Kyiv, ivan.pet@company.com, +380 (67) 123-45-67",
                "2) Олена Коваль, 03.03.1988, Senior Manager, 10 years, Lviv, o.koval@mail.ua, 032-222-33-44",
                "3) Сергій Бондар, 15.12.1999, Intern, 0 years, Odesa, s.bondar@example.com, +380671234568",
                "4) Марія Ткач, 01.01.1990, HR Specialist, 5 years, Kharkiv, maria.t@company.org, 057-555-11-22",
                "5) Андрій Шевченко, 30.06.1985, Junior Developer, 3 years, Vinnytsia, andriy.shev@work.ua, +38 050 444 55 66");
    }
}
// ===== TASK 3 & 4 supporting classes =====
@MyAnnotation(author = "Oleksandr", version = 2, tags = {"dev", "java"})
class Staff {
    public String name;
    private int age;
    private String secretNote = "initial secret";

    public Staff() {
        this.name = "Unknown";
        this.age = 0;
    }
    public Staff(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void greet(String msg) {
        System.out.println("GREET from Staff (" + name + "): " + msg + " - age=" + age);
    }
}

```

		Iлук Ол.С.			Арк.
		Плюнгівський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	3

```
age);
    }
    private String secretMethod() {
        return "SECRET METHOD: '" + secretNote + "' for " + name;
    }
    @Override
    public String toString() {
        return String.format("Staff{name='%s', age=%d}", name, age);
    }
}
// ===== TASK 4: власна анотація =====
@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
@interface MyAnnotation {
    String author() default "Unknown";
    int version() default 1;
    String[] tags() default {};
}
```

Результат программы:

```
== TASK 2: Регулярні вирази ==

Оригінальний текст:
1) Іваненко Петро, 28.05.1995, Junior Developer, 2 years, Kyiv, ivan.pet@company.com, +380 (67) 123-45-67
2) Олена Коваль, 03.03.1988, Senior Manager, 10 years, Lviv, o.koval@mail.ua, 032-222-33-44
3) Сергій Бондар, 15.12.1999, Intern, 0 years, Odesa, s.bondar@example.com, +380671234568
4) Марія Ткач, 01.01.1990, HR Specialist, 5 years, Kharkiv, maria.t@company.org, 057-555-11-22
5) Андрій Шевченко, 30.06.1985, Junior Developer, 3 years, Vinnytsia, andriy.shev@work.ua, +38 050 444 55 66

Знайдені телефони:
- +380 (67) 123-45-67
- 032-222-33-44
- +380671234568
- 057-555-11-22
- +38 050 444 55 66

Знайдені email:
- ivan.pet@company.com
- o.koval@mail.ua
- s.bondar@example.com
- maria.t@company.org
- andriy.shev@work.ua
```

Текст з переформатованими датами:

- Іваненко Петро, 1995-05-20, Junior Developer, 2 years, Kyiv, ivan.pet@company.com, +380 (67) 123-45-67
- Олена Коваль, 1988-03-03, Senior Manager, 10 years, Lviv, o.koval@mail.ua, 032-222-33-44
- Сергій Бондар, 1999-12-15, Intern, 0 years, Odesa, s.bondar@example.com, +380671234568
- Марія Ткач, 1990-01-01, HR Specialist, 5 years, Kharkiv, maria.t@company.org, 057-555-11-22
- Андрій Шевченко, 1985-06-30, Junior Developer, 3 years, Vinnytsia, andriy.shev@work.ua, +38 050 444 55 66

Текст з оновленими посадами:

- Іваненко Петро, 1995-05-20, Middle Developer, 2 years, Kyiv, ivan.pet@company.com, +380 (67) 123-45-67
- Олена Коваль, 1988-03-03, Senior Manager, 10 years, Lviv, o.koval@mail.ua, 032-222-33-44
- Сергій Бондар, 1999-12-15, Junior Developer, 0 years, Odesa, s.bondar@example.com, +380671234568
- Марія Ткач, 1990-01-01, HR Specialist, 5 years, Kharkiv, maria.t@company.org, 057-555-11-22
- Андрій Шевченко, 1985-06-30, Middle Developer, 3 years, Vinnytsia, andriy.shev@work.ua, +38 050 444 55 66

		<i>Iцук Ол.С.</i>				Арк.
		<i>Піонтківський В.І.</i>				ДУ «Житомирська політехніка».25.121.00.000 – Пр9
Змн.	Арк.	№ докум.	Підпис	Дата		4

Рис. 1-2 Завдання 2

```
*** TASK 3: Reflection API ***

Class via .class: com.education.ztu.Staff
Class via instance.getClass(): com.education.ztu.Staff
Class via Class.forName: com.education.ztu.Staff

Declared fields:
- String name
- int age
- String secretNote

Declared constructors:
- com.education.ztu.Staff()
- com.education.ztu.Staff(String, int)

Declared methods:
- toString() : String
- secretMethod() : String
- greet(String) : void

Створено об'єкт Staff: Staff{name='Ivan Petrenko', age=29}
GREET from Staff (Ivan Petrenko): Доброго дня – age=29
Отримано приватне поле secretNote: Це приватна нотатка через reflection
Результат виклику приватного методу: SECRET METHOD: 'Це приватна нотатка через reflection' for Ivan Petrenko
```

Рис. 3 Завдання 3

```
==== TASK 4: Власна анотація ===

Анотація MyAnnotation присутня на Staff:
- author(): Oleksandr
- version(): 2
- tags(): dev java

==== Кінець роботи програми ===
```

Рис. 4 Завдання 4

Посилання на репозиторій: <https://github.com/Sasha1845/Java>

Висновок: Я практикував роботу з регулярними виразами, використання рефлексії, створення власних анотацій.

		<i>Іцуک Ол.С.</i>				<i>Арк.</i>
		<i>Піонтківський В.І.</i>				<i>ДУ «Житомирська політехніка».25.121.00.000 – Пр9</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>5</i>