

Лабораторна робота №9

Регулярні вирази. Рефлексія. Анотації.

Мета роботи: 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 (ПБ, 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.

Лістинг програми:

```
package com.education.ztu;

import java.util.regex.*;

public class Task_2 {
    public static void main(String[] args) {
        String text =
            "ПІБ: Іваненко Іван Іванович, Дата народження: 20.05.1995, Посада: Менеджер, Досвід: 5 років, Адреса: Київ, Email: ivan.ivanenko@example.com, Телефон: +380(67)123-45-67\n" +
            "ПІБ: Петров Петро Петрович, Дата народження: 11.12.1988, Посада: Програміст, Досвід: 10 років, Адреса: Харків, Email: petro.petrov@gmail.com, Телефон: +380931234568\n" +
            "ПІБ: Коваленко Олена Сергіївна, Дата народження: 03.07.1992, Посада: Дизайнер, Досвід: 7 років, Адреса: Львів, Email: olena.design@ukr.net, Телефон: 067-222-33-44\n" +
            "ПІБ: Мироненко Андрій Олегович, Дата народження: 15.01.1990, Посада: Бухгалтер, Досвід: 8 років, Адреса: Дніпро, Email: andrii.miron@gmail.com, Телефон: +38050-555-66-77\n" +
            "ПІБ: Савченко Марія Василівна, Дата народження: 29.09.1998, Посада: HR, Досвід: 3 роки, Адреса: Одеса, Email: maria.hr@example.com, Телефон: 380631112233";
```

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.00.000–Пр9				
Розроб.	Іцуку О.С.				Звіт з лабораторної роботи	Літ.	Арк.	Аркушів	
Перевір.	Піонтківський В.І.							1	
Керівник								6	
Н. контр.									
Зав. каф.						ФІКТ Гр. ІПЗ-23-1			

```

        Pattern phonePattern = Pattern.compile("\\+?\\d{\\d()}-\\s{8,16}\\d");
        Pattern emailPattern = Pattern.compile("[\\w.-]+@[\\w.-]+\\.\\p{Alpha-Zero}]{2,6}");
        Pattern datePattern = Pattern.compile("(\\d{2})\\.(\\d{2})\\.(\\d{4})");

        Matcher phones = phonePattern.matcher(text);
        Matcher emails = emailPattern.matcher(text);

        System.out.println("Телефони:");
        while (phones.find()) System.out.println(phones.group());

        System.out.println("\nЕмайли:");
        while (emails.find()) System.out.println(emails.group());

        String updatedDates = datePattern.matcher(text).replaceAll("$3-$2-$1");

        String updatedPositions = updatedDates
            .replace("Менеджер", "Керівник проекту")
            .replace("Дизайнер", "UI/UX спеціаліст")
            .replace("HR", "HR-менеджер");

        System.out.println("\nОновлений текст:");
        System.out.println(updatedPositions);
    }
}

```

Результат виконання:

```

Телефони:
+380(67)123-45-67
+380931234568
067-222-33-44
+38050-555-66-77
380631112233
|
Емайли:
ivanivanenko@example.com
petro.petrov@gmail.com
oleena.design@ukr.net
andrui.miron@gmail.com
maria.hr@example.com

Оновлений текст:
ПІБ: Іваненко Іван Іванович, Дата народження: 1995-05-20, Посада: Керівник проекту, Досвід: 5 років, Адреса: Київ, Email: ivanivanenko@example.com, Телефон: +380(67)123-45-67
ПІБ: Петров Петро Петрович, Дата народження: 1988-12-11, Посада: Програміст, Досвід: 10 років, Адреса: Харків, Email: petro.petrov@gmail.com, Телефон: +380931234568
ПІБ: Коваленко Олена Сергіївна, Дата народження: 1992-07-03, Посада: UI/UX спеціаліст, Досвід: 7 років, Адреса: Львів, Email: olena.design@ukr.net, Телефон: 067-222-33-44
ПІБ: Мироненко Андрій Олегович, Дата народження: 1990-01-15, Посада: Бухгалтер, Досвід: 8 років, Адреса: Дніпро, Email: andrii.miron@gmail.com, Телефон: +38050-555-66-77
ПІБ: Савченко瑪麗亞 Василівна, Дата народження: 1998-09-29, Посада: HR-менеджер, Досвід: 3 роки, Адреса: Одеса, Email: maria.hr@example.com, Телефон: 380631112233

```

Рис. 1 Результат виконання завдання 2

Завдання 3. Робота з користувачьким класом методами Reflection API:

- Створити власний клас в якому міститимуться публічні та приватні поля, коструктори 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.

		Iuцук O.C.			ДУ «Житомирська політехніка».25.121.00.000 – Пр9	Арк.
		Плюнгіківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

- Викликати метод класу.
- Попрацювати з приватним полем (встановити та отримати значення) • Результати роботи відобразити в консолі.

Лістинг програми:

```

package com.education.ztu;

import java.lang.reflect.*;

class Employee {
    public String name;
    private int age;
    private String position;

    public Employee() {
        this.name = "Default";
        this.age = 20;
        this.position = "None";
    }

    public Employee(String name, int age, String position) {
        this.name = name;
        this.age = age;
        this.position = position;
    }

    public void printInfo() {
        System.out.println(name + ", " + age + ", " + position);
    }

    private String secretCode() {
        return "SC-" + age;
    }
}

public class Task_3 {
    public static void main(String[] args) throws Exception {
        Class<?> c1 = Employee.class;
        Class<?> c2 = Class.forName("com.education.ztu.Employee");
        Class<?> c3 = new Employee().getClass();

        System.out.println("Поля:");
        Field[] fields = c1.getDeclaredFields();
        for (Field f : fields) System.out.println(f.getName() + " : " +
f.getType().getSimpleName());

        System.out.println("\nМетоди:");
        Method[] methods = c1.getDeclaredMethods();
        for (Method m : methods) {
            System.out.print(m.getName() + "(");
            Class<?>[] params = m.getParameterTypes();
            for (int i = 0; i < params.length; i++) {
                System.out.print(params[i].getSimpleName());
                if (i < params.length - 1) System.out.print(", ");
            }
            System.out.println(") -> " + m.getReturnType().getSimpleName());
        }

        System.out.println("\nКонструктори:");
        Constructor<?>[] constructors = c1.getDeclaredConstructors();
    }
}

```

		Iлуць О.С.			Арк.
		Плюнгівський В.І.			
Змн.	Арк.	№ докум.	Підпис	Дата	3

```

        for (Constructor<?> ctor : constructors) {
            System.out.print(ctor.getName() + "(");
            Class<?>[] params = ctor.getParameterTypes();
            for (int i = 0; i < params.length; i++) {
                System.out.print(params[i].getSimpleName());
                if (i < params.length - 1) System.out.print(", ");
            }
            System.out.println(")");
        }

        System.out.println("\nСтворення екземпляра:");
        Object obj = c1.getConstructor(String.class, int.class, String.class)
            .newInstance("Олег", 20, "Програміст");
        Method printInfo = c1.getMethod("printInfo");
        printInfo.invoke(obj);

        System.out.println("\nРобота з приватним полем age:");
        Field ageField = c1.getDeclaredField("age");
        ageField.setAccessible(true);
        ageField.set(obj, 30);
        System.out.println("Нове значення age: " + ageField.get(obj));
    }
}

```

Результат виконання:

```

Поля:
name : String
age : int
position : String

Методи:
printInfo() -> void
secretCode() -> String

Конструктори:
com.education.ztu.Employee()
com.education.ztu.Employee(String, int, String)

Створення екземпляра:
Олег, 20, Програміст

Робота з приватним полем age:
Нове значення age: 30

```

Рис. 2 Результат виконання завдання 3

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.

		Iлуцк О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр9	Арк.
		Піонітківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Лістинг програми:

```
package com.education.ztu;

import java.lang.annotation.*;
import java.lang.reflect.*;

enum Level {
    LOW, MEDIUM, HIGH
}

@Target({ElementType.TYPE, ElementType.FIELD, ElementType.METHOD})
@Retention(RetentionPolicy.RUNTIME)
@Repeatable(Infos.class)
@interface Info {
    String author() default "Unknown";
    String date() default "2025-01-01";
    String description() default "No description";
    int version() default 1;
    Level level() default Level.MEDIUM;
    String[] reviewers() default {};
}

@Target({ElementType.TYPE, ElementType.FIELD, ElementType.METHOD})
@Retention(RetentionPolicy.RUNTIME)
@interface Infos {
    Info[] value();
}

@Info(author = "Oleg", date = "2025-12-01", description = "Person class", version = 2, level = Level.HIGH, reviewers = {"Anna", "Ivan"})
@Info(author = "Maria", description = "Second class annotation")
class Person {

    @Info(description = "Name field", level = Level.LOW)
    public String name;

    @Info(description = "Print method", version = 3, reviewers = {"Oleg"})
    public void print() {}
}

public class Task_4 {
    public static void main(String[] args) throws Exception {
        Class<Person> cls = Person.class;

        Info[] classInfos = cls.getAnnotationsByType(Info.class);
        for (Info i : classInfos) {
            System.out.println("CLASS: " + i.author() + ", " + i.date() + ", " +
i.description() + ", version: " + i.version() + ", level: " + i.level());
            System.out.println("Reviewers: " + String.join(" | ", i.reviewers()));
            System.out.println();
        }
        Field field = cls.getDeclaredField("name");
        Info fieldInfo = field.getAnnotation(Info.class);
        System.out.println("FIELD: " + fieldInfo.description() + ", level: " +
fieldInfo.level());

        Method method = cls.getDeclaredMethod("print");
        Info methodInfo = method.getAnnotation(Info.class);
        System.out.println("METHOD: " + methodInfo.description() + ", version: " +
methodInfo.version());
        System.out.println("Reviewers: " + String.join(" | ",
methodInfo.reviewers()));
    }
}
```

		Iuцук О.С.			ДУ «Житомирська політехніка».25.121.00.000 – Пр9	Арк.
		Плюнгівський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

```
}
```

Результат виконання:

```
CLASS: Oleg, 2025-12-01, Person class, version: 2, level: HIGH
Reviewers: Anna | Ivan

CLASS: Maria, 2025-01-01, Second class annotation, version: 1, level: MEDIUM
Reviewers:

FIELD: Name field, level: LOW
METHOD: Print method, version: 3
Reviewers: Oleg
```

Рис. 3 Результат виконання завдання 4

Посилання на репозиторій: <https://github.com/Oleg-Ischuk/Java-IPZ-23-1>

Висновок: попрактикувався працювати з регулярними виразами, використанням рефлексії, створенням власних анотацій.

		Iuцук O.C.			ДУ «Житомирська політехніка».25.121.00.000 – Пр9	Арк.
		Піонітківський В.І.				
Змн.	Арк.	№ докум.	Підпис	Дата		6