

Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №1

ДИСЦИПЛІНА: «КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ»

Виконав: студентка групи КС22
Мазуренко Анжеліки
Перевірив: Споров Олександр
Євгенович

Харків
2024

Лабораторна робота No1 Java Reflection

Git Repository: https://github.com/miorezu/LB1_JavaReflection

Основні завдання

Завдання 1

Напишіть метод, що по повному імені типу, заданому у вигляді рядка, або по об'єкту типу Class, що створений попередньо, повертає рядок з його повним описом: ім'я пакета, в якому клас визначено, модифікатори та ім'я аналізованого класу, його базовий клас, список реалізованих інтерфейсів, а також список усіх полів, конструкторів та методів, оголошених у класі, та їх характеристики.

При розв'язку задачі потрібно передбачити, що в програму для аналізу можуть бути передані як примітивні типи, так і типи-посилання (reference types): масиви, класи та інтерфейси. Для перевірки роботи напишіть консольну програму та програму з графічним інтерфейсом користувача.

У Main я зразу викликаю метод для task1.University.Student та java.io.Console, що аналізує класи та виводить інформацію про нього. Тому у концолі бачимо інформацію про них на рис.1-2.

```
Package: package task1.University
Name: Student
Modifiers: public
Base class: class java.lang.Object
Implemented interfaces: java.io.Serializable
                        javax.xml.crypto.Data

Declared fields: [private java.lang.String task1.University.Student.name, private java.lang.String task1.University.Student.surname, public java.lang.String task1.University.

Declared methods: public void task1.University.Student.rate()
public java.lang.String task1.University.Student.getGroup()
public java.lang.String task1.University.Student.getName()

Constructors: [public task1.University.Student(), public task1.University.Student(java.lang.String,java.lang.String,java.lang.String)]
```

```
Package: package java.io
Name: Console
Modifiers: public final
Base class: class java.lang.Object
Implemented interfaces: java.io.Flushable

Declared fields: [private java.lang.Object java.io.Console.readLock, private java.lang.Object java.io.Console.writeLock, private java.io.Reader java.io.Console.reader, private

Declared methods: public void java.io.Console.flush()
public java.io.Console java.io.Console.format(java.lang.String,java.lang.Object[])
public java.nio.charset.Charset java.io.Console.charset()
public java.io.Console java.io.Console.printf(java.lang.String,java.lang.Object[])
public java.lang.String java.io.Console.readLine()
public java.lang.String java.io.Console.readLine(java.lang.String,java.lang.Object[])
public java.io.Reader java.io.Console.reader()
public java.io.PrintWriter java.io.Console.writer()
private static native java.lang.String java.io.Console.encoding()
private char[] java.io.Console.grow()
private char[] java.io.Console.readLine(boolean) throws java.io.IOException
private void java.io.Console.installShutdownHook()
private static native boolean java.io.Console.echo(boolean) throws java.io.IOException
public char[] java.io.Console.readPassword(java.lang.String,java.lang.Object[])
public char[] java.io.Console.readPassword()
private static native boolean java.io.Console.isatty()

Constructors: []
```

Рисунок 1-2 – вивід інформація у консолі

Також реалізован графічний інтерфейс. У вверхньому полі вводимо повну назву класу, інформацію про який хочемо отримати та натискаємо кнопку “Analysis”(рис.3).

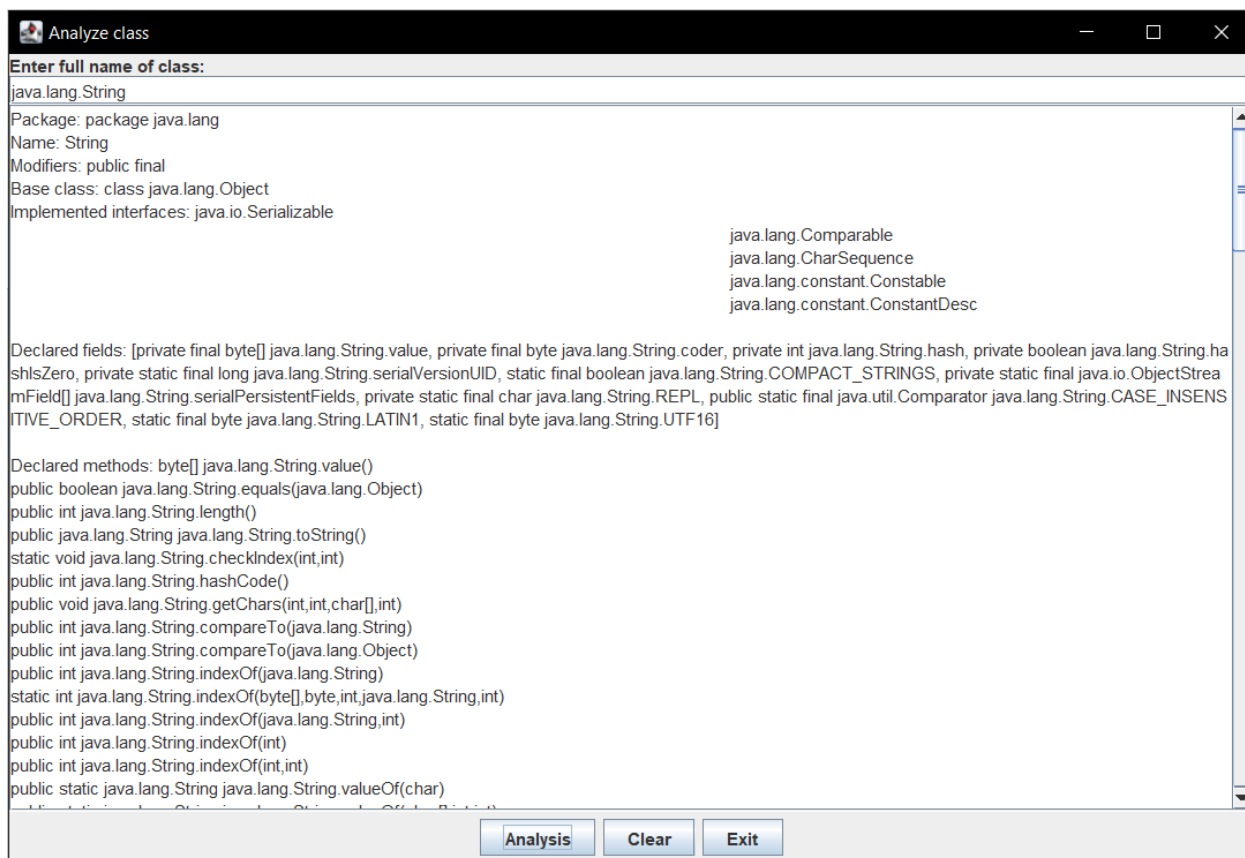


Рисунок 3 – вивід інформації у графічному інтерфейсі

Якщо клас існує то ми побачимо вивід його інформації. Кнопка “Clear” очищує поле вводу та поле виводу. Якщо клас не існує, то ми побачимо таке повідомлення (рис.4).

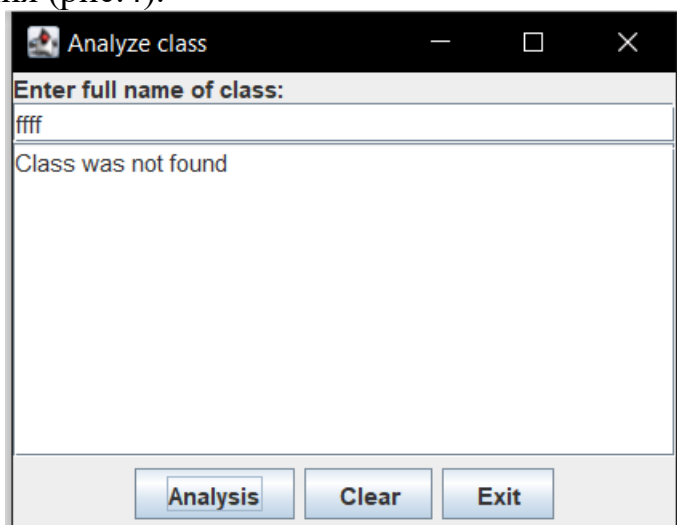


Рисунок 4 – повідомлення про не знайдений клас

Кнопка “Exit” завершує роботу програми.

Завдання 2

Напишіть метод, що по отриманому об'єкту виводить його реальний тип та стан — список всіх полів, оголошених у класі, разом з їх значеннями, а також список оголошених у класі відкритих методів. Користувач може переглянути цей список, вибрати для виклику лише відкриті методи без параметрів, викликати їх на цьому об'єкті та переглянути результат виклику.

У консолі виводиться інформація про клас та список методів, які ми можемо викликати. Вводимо його номер та викликаємо, бачимо результат виклику(рис.5-6)

```
Name of class: Student

Declared fields: [private java.lang.String task2.Student.name, private java.lang.String task2.Student.surname, public java.lang.String task2.Student.group]
Anzhelika Mazurenko CS22
Choose method to call:

1) public java.lang.String task2.Student.getGroup()
2) public void task2.Student.rate()
3) public java.lang.String task2.Student.getSName()
Print number of method:
3
What rating would you like to give?
90
You rated it - 90

Declared fields: [private java.lang.String task2.Student.name, private java.lang.String task2.Student.surname, public java.lang.String task2.Student.group]
Anzhelika Mazurenko CS22
Choose method to call:

1) public void task2.Student.rate()
2) public java.lang.String task2.Student.getGroup()
3) public java.lang.String task2.Student.getSName()
Print number of method:
1
Anzhelika Mazurenko

Process finished with exit code 0
```

Рисунок 5-6 – результат виконання 2 завдання

Завдання 3

Напишіть метод, що отримує об'єкт, ім'я методу у вигляді рядка та список необхідних для виклику методу параметрів. Якщо цей метод може бути викликаний на заданому об'єкті, то вивести результат, інакше викинути виключення *FunctionNotFoundException*.

```
Result(int): 97
Result(double): 52.0
Result(String): 2021

Process finished with exit code 0
```

Рисунок 7 – результати виконання 3 завдання

Завдання 4

Напишіть програму, що дозволяє створювати одновимірні масиви та матриці як примітивних, так і типів посилань (reference types), що будуть вказані під час роботи програми. Програма повинна вміти змінювати розміри масиву та матриці зі збереженням значень та перетворювати масиви та матриці на рядок.

```
Your array(int): [40, 15, 40, 78, 48, 67, 64, 47, 63, 36]
Your array after change size: [40, 15, 40, 78, 48, 67, 64, 47, 63, 36, null, null, null, null, null]

Your array(double): [31, 67, 86, 69, 81, 50, 14, 14, 21, 84]
Your array after change size: [31, 67, 86, 69, 81, 50, 14, 14, 21, 84, null, null, null, null, null]

Your array(String): [Hello, Me, Java, Array, Manipulation]
Your array after change size: [Hello, Me, Java, Array, Manipulation, null, null, null, null, null, null, null, null, null, null]
```

Рисунок 8 - результати виконання 4 завдання

Завдання 5

Напишіть програму, що демонструє особливості застосування «універсальних» динамічних об'єктів проксі для профілювання методу (виводить на екран час обчислення методу) та для трасування методу (виводить на екран ім'я, параметри методу та обчислене значення).

```
Package: package task5
Name: Function
Modifiers: public
Base class: class java.lang.Object
Implemented interfaces: task5.Calculate

Declared fields: []

Declared methods: public double task5.Function.funcCos(double)
public double task5.Function.funcExp(double)
public double task5.Function.funcSin(double)

Constructors: [public task5.Function()]

Our number is
31

Exponent of the number:
Elapsed time: 5437200.0 ns
res: 2.9048849665247426E13

Sin of the number:
Elapsed time: 139900.0 ns
res: -0.404037645323065

Cos of the number:
Elapsed time: 103700.0 ns
res: 0.9147423578045313
```

```
Name: Function
Modifiers: public
Base class: class java.lang.Object
Implemented interfaces: task5.Calculate

Declared fields: []

Declared methods: public double task5.Function.funcExp(double)
public double task5.Function.funcCos(double)
public double task5.Function.funcSin(double)

Constructors: [public task5.Function()]

Our number is
97

Exponent of the number:
Elapsed time: 8842900.0 ns
res: 1.3383347192042695E42

Sin of the number:
Elapsed time: 236600.0 ns
res: 0.3796077390275217

Cos of the number:
Elapsed time: 95100.0 ns
res: -0.9251475365964139
```

Рисунок 9-10 – результати виконання 5 завдання