

Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра штучного інтелекту та програмного забезпечення

ЗВІТ  
З ЛАБОРАТОРНОЇ РОБОТИ №1  
«Java Reflection»  
дисципліна: «Крос-платформне програмування»

Виконала: студентка групи КС21

Пушкіна Олеся

Перевірив: доцент кафедри ШПЗ

Споров Олександр Євгенович

Харків

2024

## **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.

### **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.

### **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`.

#### **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.

#### **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).

#### **Додаткові завдання**

##### **Завдання №1**

Напишіть програму, що дозволяє переглянути список конструкторів заданого під час роботи програми класу, вибрати потрібний конструктор та створити об'єкт цього класу, потім переглянути список усіх методів класу та вибрати потрібний метод. На кожному етапі програма повинна аналізувати та виводити на екран стан об'єкта (імена та значення його полів).

##### **Завдання №2**

Напишіть програму, що дозволяє перетворювати об'єкт у рядок із зазначенням усієї необхідної для зворотного відновлення інформації, а також відновлювати об'єкт із такого рядка.

## Результати виконання завдань:

### Завдання №1:

```
Enter the name of the class (or type 'exit' to quit): java.lang.String
Package: java.lang
public final class String implements Serializable, Comparable, CharSequence, Constable, ConstantDesc {
    private final byte[] value;
    private final byte coder;
    private int hash;
    private boolean hashIsZero;
    private static final long serialVersionUID;
    static final boolean COMPACT_STRINGS;
    private static final ObjectStreamField[] serialPersistentFields;
    private static final char REPL;
    public static final Comparator CASE_INSENSITIVE_ORDER;
    static final byte LATIN1;
    static final byte UTF16;
    public java.lang.String(StringBuilder);
    java.lang.String(char[], int, int, Void);
    java.lang.String(AbstractStringBuilder, Void);
    private java.lang.String(Charset, byte[], int, int);
    public java.lang.String(byte[], int, int, Charset);
    public java.lang.String(byte[], String);
```

Рис. 1 — скріншот роботи консольної програми ClassAnalyzer

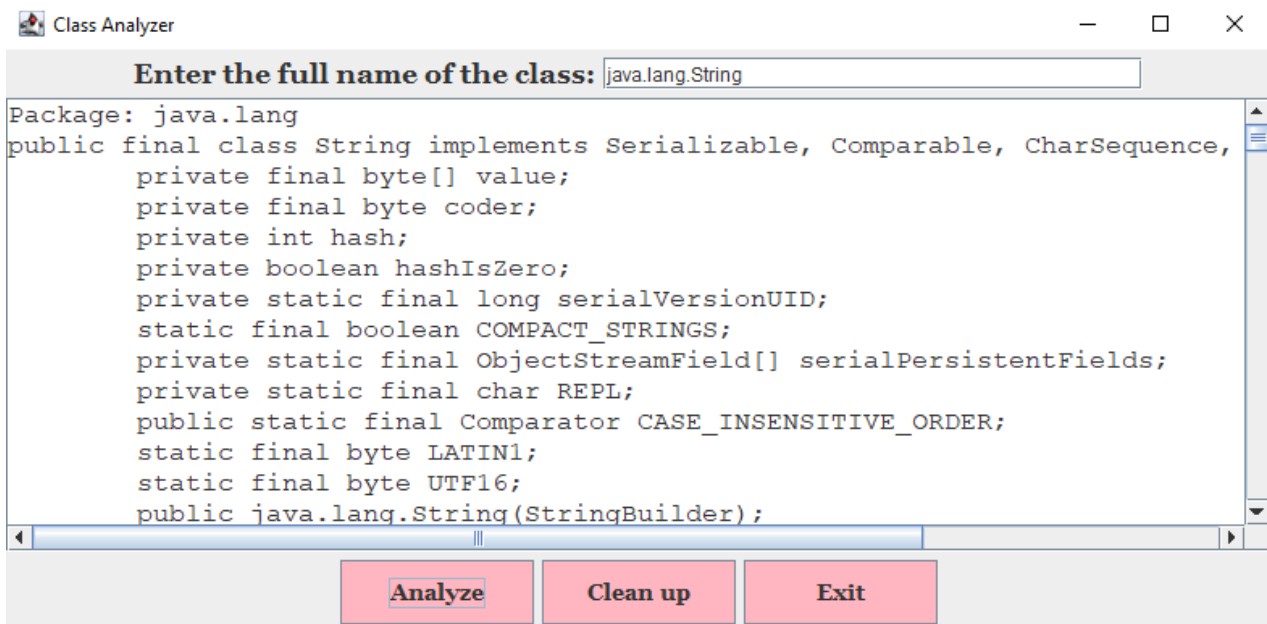


Рис. 2 — скріншот роботи програми з графічним інтерфейсом користувача ClassAnalyzerGUI

## Завдання №2

```
Тип об'єкту: Task2.Check
Стан об'єкту:
x = 3.0
y = 4.0
Список всіх відкритих методів:
1). public java.lang.String Task2.Check.toString()
2). public void Task2.Check.setRandomData()
3). public double Task2.Check.Dist()
4). public void Task2.Check.setData(double,double)
Введіть порядковий номер методу [1, 4] або ('exit' щоб вийти):
1
Результат виклику методу: Task2.Check{x=3.0, y=4.0}
Тип об'єкту: Task2.Check
Стан об'єкту:
x = 3.0
y = 4.0
Список всіх відкритих методів:
1). public java.lang.String Task2.Check.toString()
2). public void Task2.Check.setRandomData()
3). public double Task2.Check.Dist()
4). public void Task2.Check.setData(double,double)
Введіть порядковий номер методу [1, 4] або ('exit' щоб вийти):
3
Результат виклику методу: 5.0
Тип об'єкту: Task2.Check
Стан об'єкту:
x = 3.0
y = 4.0
Список всіх відкритих методів:
1). public java.lang.String Task2.Check.toString()
2). public void Task2.Check.setRandomData()
3). public double Task2.Check.Dist()
4). public void Task2.Check.setData(double,double)
Введіть порядковий номер методу [1, 4] або ('exit' щоб вийти):
```

Рис. 3 — скріншот роботи програми ObjectInspector

### Завдання №3

```
Task3.TestClass [a=1.0, exp(-abs(a)*x)*sin(x)]
Типи: double, значення: [1.0]
Результат виклику: 0.3095598756531122
Типи: double, int, значення: [1.0, 1]
Результат виклику: 0.3095598756531122
Типи: int, double, значення: [3, 3.0]
Результат виклику: 9.0
Типи: String, значення: [CS21]
Результат виклику: I Love CS21
```

Рис. 4 — скріншот роботи програми MethodCaller

### Завдання №4

```
java.lang.Integer[2] = {null, null}
java.lang.String[3] = {null, null, null}
java.lang.Double[5] = {null, null, null, null, null}
java.lang.Integer[3][5] = {
    {null, null, null, null, null}
    {null, null, null, null, null}
    {null, null, null, null, null}
}
java.lang.Integer[4][6] = {
    {null, null, null, null, null, null}
    {null, null, null, null, null, null}
    {null, null, null, null, null, null}
    {null, null, null, null, null, null}
}
java.lang.Integer[3][7] = {
    {null, null, null, null, null, null, null}
    {null, null, null, null, null, null, null}
    {null, null, null, null, null, null, null}
}
java.lang.Integer[2][2] = {
    {null, null}
    {null, null}
}
```

Рис. 5 — скріншот роботи програми MethodCaller

## Завдання №5

```
[Profiling] eval took 6974400 ns
F1: 0.06907214463000695
[Tracing] eval:
  Arguments: 1.0
  eval(1.0) = 1.0
  [Tracing] eval took 21000 ns
F2: 1.0
```

Рис. 6 — скіншот роботи програми ProxiDemo

## Додаткові завдання:

### Завдання №1

```
Enter class name to instantiate or 'exit' to quit: ExtraTasks.Task1.CheckNext
Available constructors for CheckNext:
1: public ExtraTasks.Task1.CheckNext(ExtraTasks.Task1.Pair,ExtraTasks.Task1.Pair)
2: public ExtraTasks.Task1.CheckNext(ExtraTasks.Task1.Pair,int,int)
3: public ExtraTasks.Task1.CheckNext(int,int)
4: public ExtraTasks.Task1.CheckNext()
Choose a constructor by number: 3
Input value for parameter of type int: 8
Input value for parameter of type int: 9
Current state of the object:
origin: (8, 9)
position: (8, 9)
Would you like to invoke a method? (yes/no): yes
Available methods for CheckNext:
1: public java.lang.String ExtraTasks.Task1.CheckNext.toString()
Select a method by number: 1
Method result: CheckNext:
origin: (8, 9)
position: (8, 9)
Current state of the object:
origin: (8, 9)
position: (8, 9)
Would you like to invoke a method? (yes/no): no
Enter class name to instantiate or 'exit' to quit: ExtraTasks.Task1.Pair
Available constructors for Pair:
1: public ExtraTasks.Task1.Pair()
2: public ExtraTasks.Task1.Pair(int,int)
Choose a constructor by number: 1
Current state of the object:
x: 0
y: 0
```

Рис. 7 — скіншот роботи програми ReflectionInteractive

## Завдання №2

```
Serialized form: r00ABXNyABVFeHRyYVRhc2tzLlRhc2syLLVzZXIA  
Deserialized User: User{name='Pushkina Lesia', age=19}  
  
Process finished with exit code 0
```

Рис. 8 — скріншот роботи програми