

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ

ЛАБОРАТОРНА РОБОТА № 1
З дисципліни «Крос-платформне програмування»
з теми «Java Reflection»

Виконав:

студент 2 курсу, групи КС24

Бондаренко Олег

Перевірив:

Споров Олександр Євгенович

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).

1. На рисунку 1 бачимо результат виконання коду першого завдання:

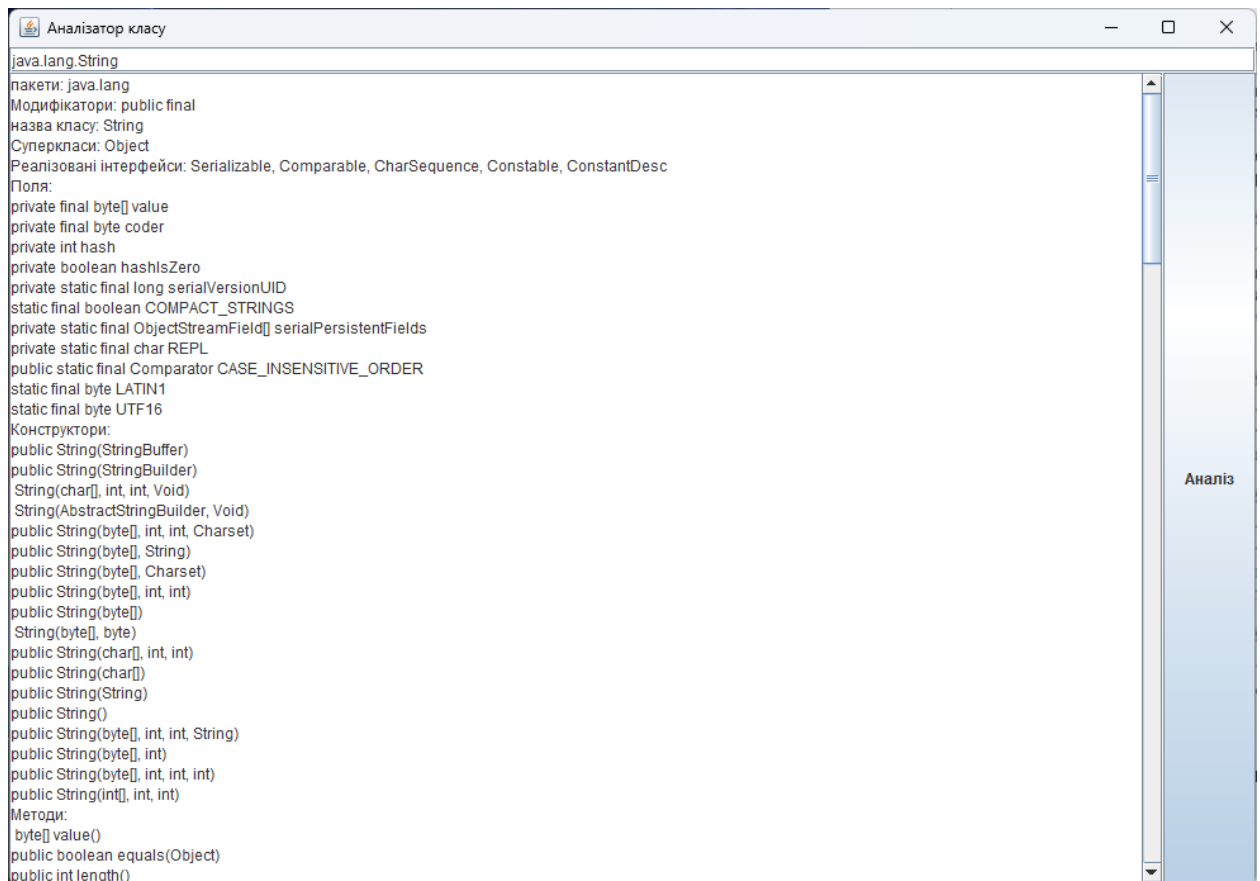


Рисунок 1. Завдання 1

2. На рисунку 2 бачимо результат виконання коду другого завдання:

```
Реальний тип об'єкта: task2.TypeInfoAnalyzer
Стан об'єкта:
Список відкритих методів:
1). public static java.lang.String task2.TypeInfoAnalyzer.getTypeInfo(java.lang.String) throws java.lang.C
2). public static java.lang.String task2.TypeInfoAnalyzer.getTypeInfo(java.lang.Class)
3). public boolean java.lang.Object.equals(java.lang.Object)
4). public java.lang.String java.lang.Object.toString()
5). public native int java.lang.Object.hashCode()
6). public final native java.lang.Class java.lang.Object.getClass()
7). public final native void java.lang.Object.notify()
8). public final native void java.lang.Object.notifyAll()
9). public final void java.lang.Object.wait(long) throws java.lang.InterruptedException
10). public final void java.lang.Object.wait(long,int) throws java.lang.InterruptedException
11). public final void java.lang.Object.wait() throws java.lang.InterruptedException
Введіть порядковий номер методу [1 ,11]: 3
Обраний метод має параметри. Введіть інший метод.
```

Рисунок 2. Завдання 2

3. На рисунку 3 бачимо результат виконання коду третього завдання:

```
Результати виклику методу:  
Типи: [double], значення: [1]  
Метод method з параметрами [class java.lang.Double] не знайдено.  
  
Process finished with exit code 0
```

Рисунок 3. Завдання 3

4. На рисунку 4 бачимо результат виконання коду четвертого завдання:

```
[[0, 0]]  
[[null, null, null]]  
[[null, null, null, null, null]]  
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]  
[[0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0]]  
[[0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0]]  
[[0, 0], [0, 0]]  
  
Process finished with exit code 0
```

Рисунок 4. Завдання 4

5. На рисунку 5 бачимо результат виконання коду п'ятого завдання:

```
[eval] took 8327400 ns  
F1: 0.1373577165473844  
[Function2.eval](1.0)  
F2: 1.0  
  
Process finished with exit code 0
```

Рисунок 5. Завдання 5