

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

ЗВІТ
З ЛАБОРАТОРНОЇ РОБОТИ №1

дисципліна: «Крос-платформне програмування»

Виконав: студент групи КС21
Барбінов Віктор Андрійович

Перевірив: доцент кафедри ШІтаПЗ
Споров Олександр Євгенович

Харків
2024

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.

Результати виконання завдання №1:

```
1. Enter full class name
2. Exit
Enter your choice: 1
Enter full class name: java.lang.String
Analyzing class: java.lang.String

Package: java.lang
public final class String implements Serializable, Comparable, CharSequence, Constable, ConstantDesc {

// Fields
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;

// Constructors
public String(StringBuffer arg0);
public String(StringBuilder arg0);
    String(char[] arg0, int arg1, int arg2, Void arg3);
    String(AbstractStringBuilder arg0, Void arg1);
public String(byte[] arg0, int arg1, int arg2, Charset arg3);
public String(byte[] arg0, String arg1);
public String(byte[] arg0, Charset arg1);
public String(byte[] arg0, int arg1, int arg2);
```

Рисунок 1 – Результат виконання консольної програми

```

// Methods
byte[] value();
public boolean equals(Object arg0);
public int length();
public String toString();
static void checkIndex(int arg0, int arg1);
public int hashCode();
public void getChars(int arg0, int arg1, char[] arg2, int arg3);
public int compareTo(String arg0);
public volatile int compareTo(Object arg0);
public int indexOf(String arg0);
static int indexOf(byte[] arg0, byte arg1, int arg2, String arg3, int arg4);
public int indexOf(String arg0, int arg1);
public int indexOf(int arg0);
public int indexOf(int arg0, int arg1);
public static String valueOf(char arg0);
public static String valueOf(char[] arg0, int arg1, int arg2);
public static String valueOf(char[] arg0);
public static String valueOf(Object arg0);
public static String valueOf(boolean arg0);
public static String valueOf(double arg0);
public static String valueOf(long arg0);
public static String valueOf(int arg0);
public static String valueOf(float arg0);
byte coder();
private static void rangeCheck(char[] arg0, int arg1, int arg2);
static void checkBoundsOffCount(int arg0, int arg1, int arg2);
private static Charset lookupCharset(String arg0);
private static char decode2(int arg0, int arg1);
private static int decodeUTF8_UTF16(byte[] arg0, int arg1, int arg2, byte[] arg3, int arg4, boolean arg5);

```

Рисунок 2 – Результат виконання консольної програми

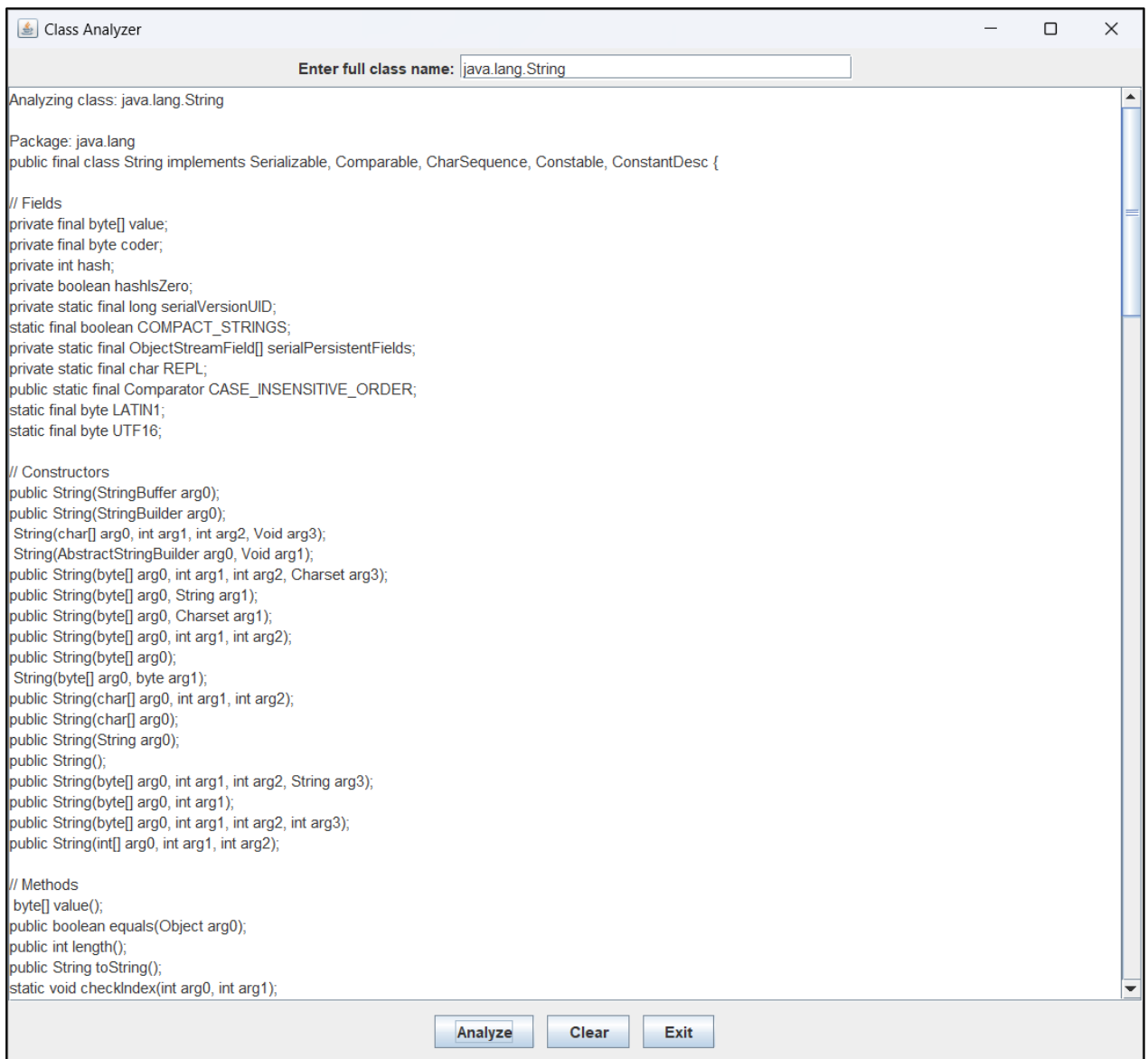


Рисунок 3 – Результат виконання програми з графічним інтерфейсом

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.

Результати виконання завдання №2:

```

Тип об'єкта: Check
Стан об'єкта:
double x = 3.0
double y = 4.0
java.lang.String name = Object1
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
1
-----
Результат виклику методу getName: Object1
-----

```

Рисунок 4 – Результат виконання методу getName

```

-----
Тип об'єкта: Check
Стан об'єкта:
double x = 3.0
double y = 4.0
java.lang.String name = Object1
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
2
-----
Результат виклику методу: Check{x=3.0, y=4.0, name='Object1'}
-----

```

Рисунок 5 – Результат виконання методу toString

```

Тип об'єкта: Check
Стан об'єкта:
double x = 3.0
double y = 4.0
java.lang.String name = Object1
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
5
-----
Результат виклику методу: 5.0
-----

```

Рисунок 6 – Результат виконання методу Dist

```

Тип об'єкта: Check
Стан об'єкта:
double x = 3.0
double y = 4.0
java.lang.String name = Object1
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
6
-----
Метод встановлення випадкових даних був успішно викликаний.
-----
Тип об'єкта: Check
Стан об'єкта:
double x = 64.699589109991637
double y = 39.393608720305075
java.lang.String name = Object1

```

Рисунок 7 – Результат виконання методу setRandomData

```

Тип об'єкта: Check
Стан об'єкта:
double x = 64.69958910991637
double y = 39.393608720305075
java.lang.String name = Object1
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
3
-----
Введіть нове значення для імені:
Object2
Метод setName був успішно викликаний.
-----
Тип об'єкта: Check
Стан об'єкта:
double x = 64.69958910991637
double y = 39.393608720305075
java.lang.String name = Object2

```

Рисунок 8 – Результат виконання методу setName

```

-----
Введіть нове значення для імені:
Object2
Метод setName був успішно викликаний.
-----
Тип об'єкта: Check
Стан об'єкта:
double x = 64.69958910991637
double y = 39.393608720305075
java.lang.String name = Object2
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
2
-----
Результат виклику методу: Check{x=64.69958910991637, y=39.393608720305075, name='Object2'}
-----

```

Рисунок 9 – Результат виконання методу toString після встановлення
НОВИХ ЗНАЧЕНЬ

```

Тип об'єкта: Check
Стан об'єкта:
double x = 64.69958910991637
double y = 39.393608720305075
java.lang.String name = Object2
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
4
-----
Введіть значення для x:
2
Введіть значення для y:
5
Метод setData був успішно викликаний.
-----

```

Рисунок 10 – Результат виконання методу setData

```

Тип об'єкта: Check
Стан об'єкта:
double x = 2.0
double y = 5.0
java.lang.String name = Object2
Список відкритих методів:
1). public java.lang.String Check.getName()
2). public java.lang.String Check.toString()
3). public void Check.setName(java.lang.String)
4). public void Check.setData(double,double)
5). public double Check.Dist()
6). public void Check.setRandomData()
Введіть порядковий номер методу для виклику, або 0 для виходу:
5
-----
Результат виклику методу: 5.385164807134504
-----

```

Рисунок 11 – Результат виконання методу Dist після встановлення нових значень

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`.

Результати виконання завдання №3:

```
TestClass [a=1.0, exp(-abs(a)*x)*sin(x)]  
Типи: [double], значення: [1.0]  
Результат виклику: 0.3095598756531122  
TestClass [a=1.0, exp(-abs(a)*x)*sin(x)]  
Типи: [double, int], значення: [1.0, 1]  
Результат виклику: 0.3095598756531122  
  
Process finished with exit code 0
```

Рисунок 12 – Результат виконання програми

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.

Результати виконання завдання №4:

```

Введіть кількість вимірів масиву (1/2), або введіть 'exit' для виходу: 1

=> Введіть дані 1: 5

Введіть повну назву класу або примітивний тип:
int
-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
0 0 0 0 0

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
3
-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
89 44 61 98 43

```

Рисунок 13 – Результат створення та заповнення одновимірного масиву

```

1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
1
Введіть новий розмір масиву:
=> Введіть дані 1: 7

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
89 44 61 98 43 0 0

```

Рисунок 14 – Результат збільшення розміру одновимірного масиву

```

Введіть новий розмір масиву:
=> Введіть дані 1: 4

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
89 44 61 98

```

Рисунок 15 – Результат зменшення розміру одновимірного масиву

```

Введіть кількість вимірів масиву (1/2), або введіть 'exit' для виходу: 2

=> Введіть дані 1: 5

=> Введіть дані 2: 5

Введіть повну назву класу або примітивний тип:
int
-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
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

```

Рисунок 16 – Результат створення матриці

```

1
Введіть нові розміри масиву:
=> Введіть дані 1: 4

```

```

=> Введіть дані 2: 2

```

```

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----

```

```

2
0 0
0 0
0 0
0 0

```

Рисунок 17 – Результат зменшення розміру матриці

```

2
51 84
74 93
76 30
36 86

```

```

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----

```

```

4
[51, 84], [74, 93], [76, 30], [36, 86]

```

```

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----

```

Рисунок 18 – Результат заповнення матриці та перетворення її на рядок

```

Введіть нові розміри масиву:
=> Введіть дані 1: 5

=> Введіть дані 2: 3

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
51 84 0
74 93 0
76 30 0
36 86 0
0 0 0

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
4
[51, 84, 0], [74, 93, 0], [76, 30, 0], [36, 86, 0], [0, 0, 0]
-----

```

Рисунок 18 – Результат збільшення розміру матриці

```

-----
1
Введіть нові розміри масиву:
=> Введіть дані 1: 2

=> Введіть дані 2: 2

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
2
51 84
74 93

-----
1 - Змінити розмір
2 - Показати
3 - Заповнити
4 - Перетворити в рядок
0 - Вихід
-----
4
[51, 84], [74, 93]
-----

```

Рисунок 19 – Результат зменшення розміру заповненої матриці

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).

Результати виконання завдання №5:

```

[Exp(-|2.5| * x) * sin(x)]: eval(1.0) = 0.06907214463000695
[[Exp(-|2.5| * x) * sin(x)]].eval took 9508800 ns
F1: 0.06907214463000695
-----
[x * x]: eval(1.0) = 1.0
[[x * x]].eval took 29700 ns
F2: 1.0

Process finished with exit code 0

```

Рисунок 19 – Результат виконання програми

Додаткове 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).

Результати виконання завдання:

```
Test Object Creation
Object: CheckNext:
origin: (0, 0)
position: (0, 0)
Let Create the same with reflection...
The beginning of creation of the CheckNext object
1). public CheckNext(Pair,Pair)
2). public CheckNext(int,int,int,int)
3). public CheckNext()
Input the Number of Constructor [1, 3]:
2
The beginning of creation of the int object
Input int value:
4
The end of creation of the int object
The beginning of creation of the int object
Input int value:
5
The end of creation of the int object
The beginning of creation of the int object
Input int value:
6
The end of creation of the int object
The beginning of creation of the int object
Input int value:
9
The end of creation of the int object
CheckNext@3eb07fd3
CheckNext:
origin: (4, 5)
position: (6, 9)

Process finished with exit code 0
```

Рисунок 20 – Результат виконання програми