Міністерство освіти і науки, молоді та спорту України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

**до лабораторної роботи №2**

**Виконав:**

Ст. гр. КН-203 Павлишин  Д. А.

**Перевірив:**

Скрибайло-Леськів Д.Ю.

**Завдання**

Exercise 2: (1) Create a class containing a float and use it to demonstrate aliasing.

Exercise 3: (1) Create a class containing a float and use it to demonstrate aliasing during

method calls.

Exercise 4: (2) Write a program that calculates velocity using a constant distance and a

constant time.

Exercise 5: (2) Create a class called Dog containing two Strings: name and says. In main(

), create two dog objects with names “spot” (who says, “Ruff!”) and “scruffy” (who says, “Wurf!”).

Then display their names and what they say.

Exercise 6: (3) Following Exercise 5, create a new Dog reference and assign it to spot’s object.

Test for comparison using == and equals( ) for all references.

Exercise 7: (3) Write a program that simulates coin-flipping

Exercise 8: (2) Show that hex and octal notations work with long values. Use

Long.toBinaryString( ) to display the results.

Exercise 9: (1) Display the largest and smallest numbers for both float and double

exponential notation.

Exercise 10: (3) Write a program with two constant values, one with alternating binary

ones and zeroes, with a zero in the least-significant digit, and the second, also alternating,

with a one in the least-significant digit (hint: It’s easiest to use hexadecimal constants for

this). Take these two values and combine them in all possible ways using the bitwise

operators, and display the results using Integer.toBinaryString( ).

Exercise 11: (3) Start with a number that has a binary one in the most significant position

(hint: Use a hexadecimal constant). Using the signed right-shift operator, right shift it all the

way through all of its binary positions, each time displaying the result using

Integer.toBinaryString( ).

Exercise 12: (3) Start with a number that is all binary ones. Left shift it, then use the

unsigned right-shift operator to right shift through all of its binary positions, each time

displaying the result using Integer.toBinaryString( ).

Exercise 13: (1) Write a method that displays char values in binary form. Demonstrate it

using several different characters

Exercise 14: (3) Write a method that takes two String arguments and uses all the

boolean comparisons to compare the two Strings and print the results. For the == and !=,

also perform the equals( ) test. In main( ), call your method with some different String

objects.

**Код програми:**

**LabTwp.java:**

package com.pavlyshyn;  
  
public class LabTwo {  
  
 public static void main(String[] args) {  
 // Task2  
 System.*out*.println("Task 2");  
 Aliasing aliasing1 = new Aliasing(3);  
 Aliasing aliasing2 = new Aliasing(4);  
 System.*out*.println("aliasing1"+aliasing1+" "+"aliasing2"+aliasing2);  
 aliasing1 = aliasing2;  
 System.*out*.println("aliasing1"+aliasing1+" "+"aliasing2"+aliasing2);  
 aliasing1.aFloat = 5;  
 System.*out*.println("aliasing1"+aliasing1+" "+"aliasing2"+aliasing2);  
 //Task3  
 System.*out*.println("Task 3");  
 Letter letter = new Letter('a');  
 System.*out*.println("letter1"+letter);  
 *ChangeLetterToZ*(letter);  
 System.*out*.println("letter1"+letter);  
 //Task4  
 System.*out*.println("Task 4");  
 Velocity velocity = new Velocity();  
 System.*out*.println(velocity.CalculateVelocity());  
 //Task5  
 System.*out*.println("Task 5");  
 Dog dog1 = new Dog("Spot", "Ruff!");  
 Dog dog2 = new Dog("scruffy", "Wurf!");  
 System.*out*.println(dog1+" "+dog2);  
 //Task6  
 System.*out*.println("Task 6");  
 Dog dog3 = dog1;  
 System.*out*.println("dog1 == dog2: " + (dog1 == dog2) );  
 System.*out*.println("dog1 == dog3: " + (dog1 == dog3) );  
 System.*out*.println("dog2 == dog3: " + (dog2 == dog3) );  
 System.*out*.println("dog1.equals(dog2): " + dog1.equals(dog2) );  
 System.*out*.println("dog1.equals(dog3): " + dog1.equals(dog3) );  
 System.*out*.println("dog2.equals(dog3): " + dog2.equals(dog3) );  
 //Task7  
 System.*out*.println("Task 7");  
 Coin coin = new Coin();  
 coin.FlipCoin();  
 System.*out*.println(coin);  
 //Task8  
 System.*out*.println("Task8");  
 long heximal = 0xffaa;  
 long octimal = 07701;  
 System.*out*.println("heximal: " + Long.*toBinaryString*(heximal));  
 System.*out*.println("octimal: " + Long.*toBinaryString*(octimal));  
 //Task9  
 System.*out*.println("Task 9");  
 System.*out*.println(Float.*MIN\_EXPONENT*);  
 System.*out*.println(Float.*MAX\_EXPONENT*);  
 System.*out*.println(Double.*MIN\_EXPONENT*);  
 System.*out*.println(Double.*MAX\_EXPONENT*);  
 //Task10  
 System.*out*.println("Task 10");  
 int Val1 = 0xAA, Val2 = 0x55;  
 System.*out*.println(Integer.*toBinaryString*(Val1));  
 System.*out*.println(Integer.*toBinaryString*(Val2));  
 System.*out*.println("Val1 & Val2 = " + Integer.*toBinaryString*(Val1 & Val2));  
 System.*out*.println("Val1 | Val2 = " + Integer.*toBinaryString*(Val1 | Val2));  
 System.*out*.println("Val1 ^ Val2 = " + Integer.*toBinaryString*(Val1 ^ Val2));  
 System.*out*.println("~Val1 = " + Integer.*toBinaryString*(~Val1));  
 System.*out*.println("~Val2 = " + Integer.*toBinaryString*(~Val2));  
  
 //Task11  
 System.*out*.println("Task 11");  
 long heximal2 = Integer.*MAX\_VALUE*;  
 while(heximal2!=0) {  
 System.*out*.println(Long.*toBinaryString*(heximal2));  
 heximal2>>=1;  
 }  
 //Task12  
 System.*out*.println("Task 12");  
 heximal2 =-1;  
 System.*out*.println(Long.*toBinaryString*(heximal2));  
 heximal2 <<= 10;  
 System.*out*.println(Long.*toBinaryString*(heximal2));  
 while (heximal2!=0){  
 heximal2 >>>= 1;  
 System.*out*.println(Long.*toBinaryString*(heximal2));  
 }  
 //Task13  
 System.*out*.println("Task 13");  
 for(char i ='a'; i< 26+'a'; i++){  
 System.*out*.println("Char:" + i + " Binary code:" + Integer.*toBinaryString*(i));  
 }  
 //Task14  
 System.*out*.println("Task 14");  
 *сompareStringAndPrint*("Test14", "Test14");  
 *сompareStringAndPrint*("Test14", "Test1");  
 }  
  
 public static void сompareStringAndPrint(String s1, String s2) {  
 System.*out*.println(s1 + "==" + s2 +" :" + (s1 == s2));  
 System.*out*.println(s1 + "!=" + s2 +" :" + (s1 != s2));  
 System.*out*.println(s1 + ".equals(" + s2 +") :" + s1.equals(s2));  
 }  
 public static void ChangeLetterToZ(Letter letter){  
 letter.symbol = 'z';  
 }  
}

**Aliasing.java:**

package com.pavlyshyn;  
  
public class Aliasing {  
 float aFloat;  
 public Aliasing(float aFloat) {  
 this.aFloat = aFloat;  
 }  
 @Override  
 public String toString() {  
 return "Aliasing{" +  
 "aFloat=" + aFloat +  
 '}';  
 }  
}

**Letter.java:**

package com.pavlyshyn;  
  
public class Letter {  
 char symbol;  
  
 public Letter(char symbol) {  
 this.symbol = symbol;  
 }  
  
 @Override  
 public String toString() {  
 return "Letter{" +  
 "symbol=" + symbol +  
 '}';  
 }  
}

**Velocity.java:**

package com.pavlyshyn;  
  
public class Velocity {  
 private static final int *time* = 3;  
 private static final double *distance* = 1.5;  
 public double CalculateVelocity(){  
 return *distance*/*time*;  
 }  
}

**Dog.java:**

package com.pavlyshyn;  
  
import java.util.Objects;  
  
public class Dog {  
 String name;  
 String says;  
  
 public Dog(String name, String says) {  
 this.name = name;  
 this.says = says;  
 }  
  
 @Override  
 public String toString() {  
 return "Dog{" +  
 "name='" + name + '\'' +  
 ", says='" + says + '\'' +  
 '}';  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Dog dog = (Dog) o;  
 return Objects.*equals*(name, dog.name) &&  
 Objects.*equals*(says, dog.says);  
 }  
  
}

**Coin.java:**

package com.pavlyshyn;  
  
import java.util.Random;  
  
public class Coin {  
 String side;  
 void FlipCoin(){  
 side = new Random().nextBoolean()? "Head":"Tail";  
 }  
  
 @Override  
 public String toString() {  
 return "Coin{" +  
 "side='" + side + '\'' +  
 '}';  
 }  
}

**Реалізація програми:**

"C:\Program Files\Java\jdk1.8.0\_181\bin\java.exe" -agentlib:jdwp=transport=dt\_socket,address=127.0.0.1:55276,suspend=y,server=n -javaagent:C:\Users\dimap\.IdeaIC2018.2\system\captureAgent\debugger-agent.jar=file:/C:/Users/dimap/AppData/Local/Temp/capture.props -Dfile.encoding=UTF-8 -classpath "C:\Program Files\Java\jdk1.8.0\_181\jre\lib\charsets.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\deploy.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\access-bridge-64.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\cldrdata.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\dnsns.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\jaccess.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\jfxrt.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\localedata.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\nashorn.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\sunec.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\sunjce\_provider.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\sunmscapi.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\sunpkcs11.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\ext\zipfs.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\javaws.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\jce.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\jfr.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\jfxswt.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\jsse.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\management-agent.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\plugin.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\resources.jar;C:\Program Files\Java\jdk1.8.0\_181\jre\lib\rt.jar;E:\Navchannya\2course\Applied Programming\LabAP\Lab 2\out\production\Lab2;C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2018.2.3\lib\idea\_rt.jar" com.pavlyshyn.LabTwo

Connected to the target VM, address: '127.0.0.1:55276', transport: 'socket'

Task 2

aliasing1Aliasing{aFloat=3.0} aliasing2Aliasing{aFloat=4.0}

aliasing1Aliasing{aFloat=4.0} aliasing2Aliasing{aFloat=4.0}

aliasing1Aliasing{aFloat=5.0} aliasing2Aliasing{aFloat=5.0}

Task 3

letter1Letter{symbol=a}

letter1Letter{symbol=z}

Task 4

0.5

Task 5

Dog{name='Spot', says='Ruff!'} Dog{name='scruffy', says='Wurf!'}

Task 6

dog1 == dog2: false

dog1 == dog3: true

dog2 == dog3: false

dog1.equals(dog2): false

dog1.equals(dog3): true

dog2.equals(dog3): false

Task 7

Coin{side='Tail'}

Task8

heximal: 1111111110101010

octimal: 111111000001

Task 9

-126

127

-1022

1023

Task 10

10101010

1010101

Val1 & Val2 = 0

Val1 | Val2 = 11111111

Val1 ^ Val2 = 11111111

~Val1 = 11111111111111111111111101010101

~Val2 = 11111111111111111111111110101010

Task 11

1010101010101010

101010101010101

10101010101010

1010101010101

101010101010

10101010101

1010101010

101010101

10101010

1010101

101010

10101

1010

101

10

1

Task 12

11111111111111111111111111111111

11111111111111111111110000000000

1111111111111111111111000000000

111111111111111111111100000000

11111111111111111111110000000

1111111111111111111111000000

111111111111111111111100000

11111111111111111111110000

1111111111111111111111000

111111111111111111111100

11111111111111111111110

1111111111111111111111

111111111111111111111

11111111111111111111

1111111111111111111

111111111111111111

11111111111111111

1111111111111111

111111111111111

11111111111111

1111111111111

111111111111

11111111111

1111111111

111111111

11111111

1111111

111111

11111

1111

111

11

1

0

Task 13

Char:a Binary code:1100001

Char:b Binary code:1100010

Char:c Binary code:1100011

Char:d Binary code:1100100

Char:e Binary code:1100101

Char:f Binary code:1100110

Char:g Binary code:1100111

Char:h Binary code:1101000

Task 14

Test14==Test14 :true

Test14!=Test14 :false

Test14.equals(Test14) :true

Test14==Test1 :false

Test14!=Test1 :true

Test14.equals(Test1) :false

Disconnected from the target VM, address: '127.0.0.1:55276', transport: 'socket'

Process finished with exit code 0