

**The course “Application development with Java”**

Practical assignment 1

Student name: Roberts

Student surname: Dubovskis

St. code: 62085

**RIGA**

**2022**

Contents

[Exercise 1 3](#_Toc97740283)

[Exercise 2 3](#_Toc97740284)

[Exercise 3 4](#_Toc97740285)

[Exercise 4 5](#_Toc97740286)

[Exercise 5 5](#_Toc97740287)

[Exercise 6 6](#_Toc97740288)

[Exercise 7 7](#_Toc97740289)

# Exercise 1



Img.1 (result)

package com.java;  
  
*//x (t) = (t ^ 2) \* a \* 0.5 + v0 \* t + x0*public class GravityCalculator {  
 static final double *GRAVITY\_CONSTANT* = 9.81;  
  
 double fallingTime = 0;  
 int initialPosition = 0;  
 int initialVelocity = 0;  
  
  
 public GravityCalculator(double fallingTime, int initialPosition, int initialVelocity) {  
 this.fallingTime = fallingTime;  
 this.initialPosition = initialPosition;  
 this.initialVelocity = initialVelocity;  
 }  
  
 public void calculate() {  
 *//4. Convert the received initialPosition and initialVelocity parameters to type double.  
 // Convert the fallingTime parameter to type int. (0.5p.)* double fallingTime\_double = fallingTime;  
 int initialPosition\_int = initialPosition;  
 int initialVelocity\_int = initialVelocity;  
  
 double finalPosition = (Math.*pow*(fallingTime\_double, 2))  
 \* (*GRAVITY\_CONSTANT* \* 0.5)  
 + (initialVelocity\_int \* fallingTime\_double)  
 + initialPosition\_int;  
  
 System.*out*.println("Position in " + fallingTime + " seconds:" + finalPosition);  
 }  
}

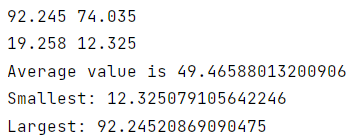
# Exercise 2



Img.2 (result)

package com.java;  
  
public class FactorialCalculator {  
  
 long factorial = 0;  
  
 public FactorialCalculator(int factorial) {  
 this.factorial = factorial;  
 }  
  
 public long factorial\_calculations() {  
 long val = factorial;  
 for (long i = factorial - 1; i > 0; i--) {  
 val = val \* i;  
 }  
 return val;  
 }  
  
  
 public long calculate\_factorial\_recursive() {  
 return factorial\_calculations\_recursive(factorial);  
 }  
  
 public long factorial\_calculations\_recursive(long factorial) {  
 if (factorial == 0) {  
 return 1;  
 }  
 return factorial \* factorial\_calculations\_recursive(factorial - 1);  
 }  
}

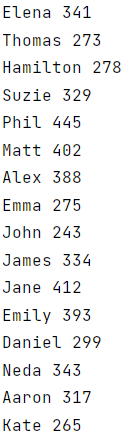
# Exercise 3



Img.3 (result)

package com.java;  
  
public class OneDimensionalArray {  
  
 void createRandomArray(int size) {  
 double[][] array = new double[size][size];  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 array[i][j] = Math.*random*() \* 100; *// not said to include 0 and 100* }  
 }  
 arrOutput(array, size);  
 averageVal(array, size);  
 largestSmallestVal(array, size);  
 }  
  
 void arrOutput(double[][] array, int size) {  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 double val = Math.*round*(array[i][j] \* 1000.0) / 1000.0;  
 System.*out*.print(val + " ");  
 }  
 System.*out*.println();  
 }  
 }  
  
 void averageVal(double[][] array, int size) {  
 double buf = 0;  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 buf += array[i][j];  
 }  
 }  
 double averageVal = buf / (Math.*pow*(size, 2));  
 System.*out*.println("Average value is " + averageVal);  
 }  
  
 void largestSmallestVal(double[][] array, int size) {  
 double largest = array[0][0];  
 double smallest = array[0][0];  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 if (array[i][j] > largest) {  
 largest = array[i][j];  
 }  
 if (array[i][j] < smallest) {  
 smallest = array[i][j];  
 }  
 }  
 }  
 System.*out*.println("Smallest: " + smallest);  
 System.*out*.println("Largest: " + largest);  
 }  
  
}

# Exercise 4



Img.4 (result)

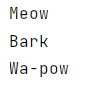
package com.java;  
  
public class Marathon {  
  
 String[] names = {"Elena", "Thomas", "Hamilton", "Suzie", "Phil", "Matt", "Alex",  
 "Emma", "John", "James", "Jane", "Emily", "Daniel", "Neda", "Aaron", "Kate"};  
  
 int[] times = {341, 273, 278, 329, 445, 402, 388, 275, 243, 334, 412, 393, 299,  
 343, 317, 265};  
  
 void printResults() {  
 int sizeN = names.length;  
 int sizeT = times.length;  
 if (sizeN != sizeT) {  
 System.*out*.println("Error");  
 return;  
 }  
 for (int i = 0; i < sizeN; i++) {  
 System.*out*.print(names[i] + " " + times[i] + "\n");  
 }  
 }  
  
}

# Exercise 5

Img.5 (result)

package com.java;  
  
import java.util.Random;  
  
public class TwoDimensionalArray {  
  
 void createRandomArray(int size) {  
 int[][] array = new int[size][size];  
 Random rand = new Random();  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 array[i][j] = rand.nextInt(101);  
 }  
 }  
 arrOutput(array, size);  
 readOneDimensionalArray(array,size);  
  
 }  
  
 void arrOutput(int[][] array, int size) {  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size; j++) {  
 System.*out*.print(array[i][j] + " ");  
 }  
 System.*out*.println();  
 }  
 for (int i = 0; i <size; i++) {  
 colAverage(array, size,i);  
 }  
 }  
  
 *//Create a print function that will print to the console: Array;  
 // Average values of each column and column.* void colAverage(int[][] array, int size, int col){  
 int buf=0;  
 for (int i = 0; i < size; i++) {  
 buf+=array[i][col];  
 }  
 System.*out*.print(buf/size + " ");  
 }  
  
 void readOneDimensionalArray(int[][] array,int size){  
 int buf=0;  
 for (int i = 0; i < size; i++) {  
 buf+=array[0][i];  
 }  
 buf/=size;  
 }  
}

# Exercise 6



Img.6 (result)

package animals;  
  
public interface Animal {  
 public void sound();  
}

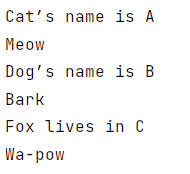
package animals;  
  
public class Animals {  
 public static void main(String[] args) {  
 Cat cat = new Cat();  
 *//cat.printName();* cat.sound();  
  
 Dog dog = new Dog();  
 *//dog.printName();* dog.sound();  
  
 Fox fox = new Fox();  
 *//fox.printHabitat();* fox.sound();  
 }  
}

package animals;  
  
public class Cat implements Animal{  
 @Override  
 public void sound()  
 {  
 System.*out*.println("Meow");  
 }  
}

package animals;  
  
public class Dog implements Animal{  
 @Override  
 public void sound()  
 {  
 System.*out*.println("Bark");  
 }

package animals;  
  
public class Fox implements Animal {  
 public void sound()  
 {  
 System.*out*.println("Wa-pow");  
 }  
}

# Exercise 7



Img.7 (result)

package animals;  
  
public class Animals {  
 public static void main(String[] args) {  
 Cat cat = new Cat("");  
 cat.printName();  
 cat.sound();  
  
 Dog dog = new Dog("");  
 dog.printName();  
 dog.sound();  
  
 Fox fox = new Fox("");  
 fox.printHabitat();  
 fox.sound();  
 }  
}

package animals;  
  
public interface Domestic {  
 public void printName () ;  
}

package animals;  
  
public interface Wild {  
 public void printHabitat();  
}

package animals;  
  
public class Cat implements Animal, Domestic {  
 @Override  
 public void sound()  
 {  
 System.*out*.println("Meow");  
 }  
  
 public String name=null;  
 public Cat(){  
 this.name = "CAT";  
 }  
 public Cat(String name) {  
 this.name = name;  
 }  
  
 public void printName() {  
 System.*out*.println("Cat’s name is " + name);  
 }  
}

package animals;  
  
public class Dog implements Animal, Domestic{  
 @Override  
 public void sound()  
 {  
 System.*out*.println("Bark");  
 }  
  
 public String name=null;  
 public Dog(){  
 this.name = "DOG";  
 }  
 public Dog(String name) {  
 this.name = name;  
 }  
  
 public void printName() {  
 System.*out*.println("Dog’s name is " + name);  
 }  
}

package animals;  
  
public class Fox implements Animal,Wild{  
 public void sound()  
 {  
 System.*out*.println("Wa-pow");  
 }  
 public String name=null;  
 public Fox(){  
 this.name = "Worldwide";  
 }  
 public Fox(String name) {  
 this.name = name;  
 }  
  
 public void printHabitat() {  
 System.*out*.println("Fox lives in " + name);  
 }  
}