Міністерство освіти і науки України

Національний університет «Львівська політехніка»



# Звіт

Про виконання

Лабораторної роботи № 1

з дисципліни: «Прикладне програмування»

на тему:

«Основами мови програмування Java.»

**Виконав:**

cтудент групи ІТ-22

Степанчук. С. Т.

**Прийняв:**

асистент Тихонов В.А.

Львів

2021

## Мета роботи:

Ознайомитися із основами мови програмування Java.

## Завдання 1

### Умова

Реалізувати сортування масиву String за довжиною.

### Код

package Task1;

import java.util.Arrays;

import java.util.Comparator;

public class Task1

{

public static void main(final String[] args)

{

final String[] arr = {

"1",

"22",

"333",

"4444",

"55555",

"666666",

"7777777"

};

for (int i = 0; i < arr.length; i++)

System.out.println(arr[i]);

Arrays.sort(arr, Comparator.comparingInt(String::length).reversed());

System.out.println("\t\t\t");

for (int i = 0; i < arr.length; i++)

System.out.println(arr[i]);

}

}

### Результат виконання на рисунку 1

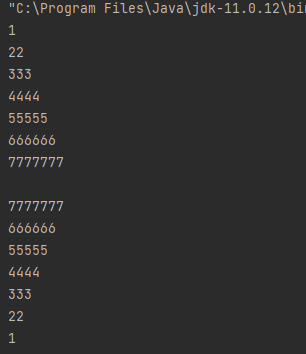


Рис. 1

## Завдання 2

### Умова

Написати програму для обчислення кількості заданого символу у масиві

String

### Код

package Task2;

public class Task2

{

public static void main(final String[] args) {

final char ch = 'a';

final String str = "aaabbaaa";

int counter = 0;

for (int i = 0; i < str.length(); i++)

if (str.charAt(i) == ch)

++counter;

System.out.format("%d '%c' in '%s'\n", counter, ch, str);

}

}

### Результат виконання на рисунку 2

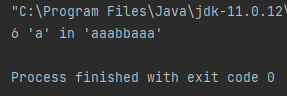


Рис. 2

## Завдання 3

### Умова

Почергово записати масив int та масив char в одну String. Приклад виводу,

"1e8t8c7b2"

### Код

package Task3;

import java.util.Arrays;

public class Task3

{

public static void main(final String[] args)

{

final char[] chArr = {'a', 'b', 'c', 'd'};

final int[] iArr = {1, 2, 3, 4};

System.out.format("Char array -\t%s\n", Arrays.toString(chArr));

System.out.format("Int array -\t\t%s\n", Arrays.toString(iArr));

System.out.format("Output -\t\t%s\n", combineIntArrayCharArrayToString(iArr, chArr));

}

public static String combineIntArrayCharArrayToString(final int[] iArr, final char[] chArr)

{

String outStr = "";

final int maxSize = chArr.length > iArr.length ? chArr.length : iArr.length;

for (int i = 0; i < maxSize; i++)

outStr += (i < iArr.length ? String.valueOf(iArr[i]) : "") +

(i < chArr.length ? String.valueOf(chArr[i]) : "");

return outStr;

}

}

### Результат зображений на рисунку 3

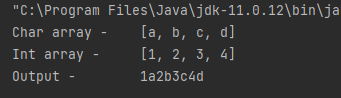


Рис. 3

## Завдання 4

### Умова

Вивести в консоль числа від 1 до 100. Замість чисел кратних 3 виводити

Fizz, замість чисел кратних 5 вивести Buzz. Замість чисел що кратні і 3, і 5

вивести FizzBuzz.

### Код

package Task4;

public class Task4

{

public static void main(final String[] args)

{

for (int i = 1; i <= 100; i++) {

if (i % 15 == 0)

System.out.print("FizzBuzz");

else if (i % 5 == 0)

System.out.print("Buzz");

else if (i % 3 == 0)

System.out.print("Fizz");

else

System.out.print(i);

System.out.print(" ");

if (i % 20 == 0)

System.out.print('\n');

}

System.out.print('\n');

}

}

### Результат виконання на рисунку 4

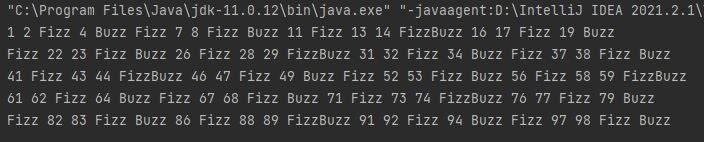


Рис. 4

## Завдання 5

### Умова

Транспонувати матрицю n X m.

### Код

package Task5;

import java.util.Random;

public class Task5

{

public static final Random rand = new Random();

public static void main(final String[] args)

{

final int n = 5;

final int m = 8;

final int maxNum = 100;

final int[][] matrix = createAndFillMatrix(n, m, maxNum);

final int[][] matrixT = transposeMatrix(matrix);

// out

final int max\_num\_length = String.valueOf(maxNum).length() + 1;

System.out.format("Start matrix[%d][%d] - \n", matrix.length, matrix[0].length);

printMatrix(matrix, max\_num\_length);

System.out.format("Transposed matrix[%d][%d] - \n", matrixT.length, matrixT[0].length);

printMatrix(matrixT, max\_num\_length);

}

public static int[][] createAndFillMatrix(final int n, final int m, final int max\_num)

{

final int[][] matrix = new int[n][m];

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

matrix[i][j] = rand.nextInt(max\_num);

}

}

return matrix;

}

public static void printMatrix(final int[][] matrix, final int maxNumLength)

{

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

System.out.format("%" + maxNumLength + "d", matrix[i][j]);

}

System.out.print('\n');

}

}

public static int[][] transposeMatrix(final int[][] matrix)

{

final int[][] matrixT = new int[matrix[0].length][matrix.length];

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

matrixT[j][i] = matrix[i][j];

}

}

return matrixT;

}

}

### Результат виконання на рисунку 5

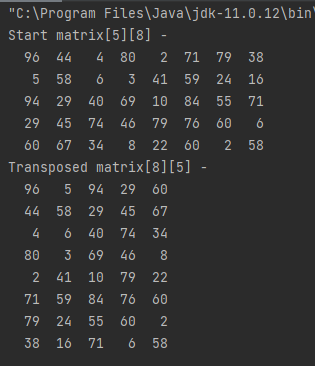


Рис. 5

## Завдання 6

### Умова

Написати програму для обчислення множення матриць

### Код

package Task6;

public class Task6

{

public static void main(final String[] args)

{

final int maxNumLength = 3;

final int[][] matrix\_1 = {

{1, 3, 5},

{2, 3, 8},

{1, 2, 2}

};

final int[][] matrix\_2 = {

{6, 6, 8},

{1, 5, 2},

{8, 4, 3}

};

final int[][] matrix\_out = multiplyMatrix(matrix\_1, matrix\_2);

// out

System.out.format("matrix\_1[%d][%d] - \n", matrix\_1.length, matrix\_1[0].length);

printMatrix(matrix\_1, maxNumLength);

System.out.format("matrix\_2[%d][%d] - \n", matrix\_2.length, matrix\_2[0].length);

printMatrix(matrix\_2, maxNumLength);

if (matrix\_out != null)

{

System.out.format("matrix\_out[%d][%d] - \n", matrix\_out.length, matrix\_out[0].length);

printMatrix(matrix\_out, maxNumLength);

}

else

System.out.println("Matrix multiplication is not possible");

}

public static int[][] multiplyMatrix(final int[][] m1, final int[][] m2) {

final int m1ColLength = m1[0].length; // m1 columns length

final int m2RowLength = m2.length; // m2 rows length

if(m1ColLength != m2RowLength) return null; // matrix multiplication is not possible

final int mRRowLength = m1.length; // m result rows length

final int mRColLength = m2[0].length; // m result columns length

final int[][] mResult = new int[mRRowLength][mRColLength];

for(int i = 0; i < mRRowLength; i++) { // rows from m1

for(int j = 0; j < mRColLength; j++) { // columns from m2

for(int k = 0; k < m1ColLength; k++) { // columns from m1

mResult[i][j] += m1[i][k] \* m2[k][j];

}

}

}

return mResult;

}

public static void printMatrix(final int[][] matrix, final int max\_num\_length)

{

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++) {

System.out.format("%" + max\_num\_length + "d", matrix[i][j]);

}

System.out.print('\n');

}

}

}

### Результат виконання на рисунку 6

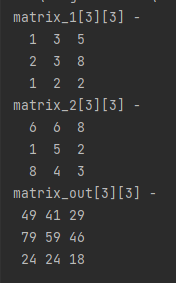


Рис. 6

## Завдання 7

### Умова

Реалізувати клас User з полями firstName, lastName, age та email.

Перевизначити стандартні методи equals() та hashCode();

### Код

// User.java

package Task7;

public class User {

public String firstName;

public String lastName;

public int age;

public String email;

@Override

public boolean equals(final Object o) {

if (this == o) return true;

if (!(o instanceof User)) return false;

final User user = (User) o;

if (age != user.age) return false;

if (firstName != null ? !firstName.equals(user.firstName) : user.firstName != null) return false;

if (lastName != null ? !lastName.equals(user.lastName) : user.lastName != null) return false;

return email != null ? email.equals(user.email) : user.email == null;

}

@Override

public int hashCode() {

int result = firstName != null ? firstName.hashCode() : 0;

result = 31 \* result + (lastName != null ? lastName.hashCode() : 0);

result = 31 \* result + age;

result = 31 \* result + (email != null ? email.hashCode() : 0);

return result;

}

}

package Task7;

public class Task7

{

public static void main(final String[] args)

{

final User us = new User();

final User us2 = new User();

final User us3 = new User();

us3.age = 22;

System.out.format("us and us2 %b\n", us.equals(us2));

System.out.format("us and us3 %b\n", us.equals(us3));

System.out.format("us and us2 %b\n", us.equals(us2));

System.out.format("us and us3 %b\n", us.equals(us3));

}

}

### Результат виконання на рисунку 7

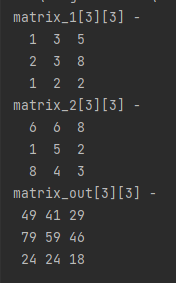


Рис. 7

## Завдання 8

### Умова

Реалізувати клас CustomDouble з двома полями: типу int(ціла частина) та

double (дробова частина). Перевизначити операції суми, різниці,

порівняння, equals() і hashCode(). Реалізувати метод toDouble()

### Код

// CustomDouble.java

package Global;

public class CustomDouble

{

public double whole = 0;

public double fractional = 0;

public CustomDouble(final double whole, final double fractional) {

this.whole = whole;

this.fractional = fractional;

}

@Override

public boolean equals(final Object o) {

if (this == o) return true;

if (!(o instanceof CustomDouble)) return false;

final CustomDouble that = (CustomDouble) o;

if (Double.compare(that.whole, whole) != 0) return false;

return Double.compare(that.fractional, fractional) == 0;

}

@Override

public int hashCode() {

int result;

long temp;

temp = Double.doubleToLongBits(whole);

result = (int) (temp ^ (temp >>> 32));

temp = Double.doubleToLongBits(fractional);

result = 31 \* result + (int) (temp ^ (temp >>> 32));

return result;

}

public double toDouble() {

return whole + fractional;

}

}

package Task8;

import Global.CustomDouble;

public class Task8

{

public static void main(final String[] args) {

final CustomDouble cs = new CustomDouble(1, 0.3);

final CustomDouble cs2 = new CustomDouble(1, 0.3);

final CustomDouble cs3 = new CustomDouble(22, 0.11);

System.out.format("cs and cs2 %b\n", cs.equals(cs2));

System.out.format("cs and cs3 %b\n", cs.equals(cs3));

System.out.format("Output - %,.4f\n", cs3.toDouble());

}

}

### Результат виконання на рисунку 8

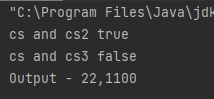


Рис. 8

## Завдання 9

### Умова

Реалізувати калькулятор. Якщо вхідними даними є дробові числа -

використати клас CustomDouble (при використанні методів добутку і

ділення - привести CustomDouble до Double).

### Код

// CustomCalc.java

package Task9;

import Global.CustomDouble;

public class CustomCalc

{

public static double multiplication(final CustomDouble f, final CustomDouble s)

{

return f.toDouble() \* s.toDouble();

}

public static double division(final CustomDouble f, final CustomDouble s)

{

return f.toDouble() / s.toDouble();

}

public static double addition(final CustomDouble f, final CustomDouble s)

{

return f.toDouble() + s.toDouble();

}

public static double subtraction(final CustomDouble f, final CustomDouble s)

{

return f.toDouble() - s.toDouble();

}

public static int multiplication(final int f, final int s)

{

return f \* s;

}

public static int division(final int f, final int s)

{

return f / s;

}

public static int addition(final int f, final int s)

{

return f + s;

}

public static int subtraction(final int f, final int s)

{

return f - s;

}

}

// Task9.java

package Task9;

import Global.CustomDouble;

public class Task9

{

public static void main(final String[] args)

{

final String[] iResultsMap = {

"\*",

"/",

"+",

"-"

};

final int[] iResults = {

CustomCalc.multiplication(1, 2), // addition

CustomCalc.division(1, 2),

CustomCalc.addition(1, 2),

CustomCalc.subtraction(1, 2)

};

final double[] dResults = {

CustomCalc.multiplication(new CustomDouble(1, 0), new CustomDouble(2, 0)), // addition

CustomCalc.division(new CustomDouble(1, 0), new CustomDouble(2, 0)),

CustomCalc.addition(new CustomDouble(1, 0), new CustomDouble(2, 0)),

CustomCalc.subtraction(new CustomDouble(1, 0), new CustomDouble(2, 0))

};

System.out.format("Int calc results - \n");

for (int i = 0; i < iResultsMap.length; i++)

System.out.format("1 %s 2 = %d\n", iResultsMap[i], iResults[i]);

System.out.format("Double calc results - \n");

for (int i = 0; i < iResultsMap.length; i++)

System.out.format("1.0 %s 2.0 = %,.04f\n", iResultsMap[i], dResults[i]);

}

}

### Результат виконання на рисунку 9

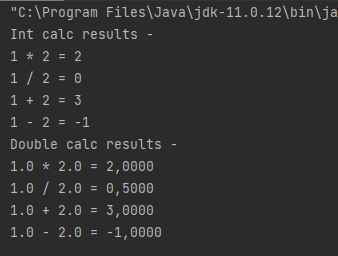


Рис. 9

## Завдання 10

### Умова

Реалізувати класи День і Погода. Створити ієрархію класів та використати

структуру даних "масив", щоб отримати календар погоди за тиждень.

Самостійно визначити та реалізувати необхідні методи, та додаткові класи

### Код

// Weather.java

package Task10;

public class Weather

{

private final double temperature; // температура

private final double pressure; // тиск

private final double humidity; // вологість

public Weather(final double temperature, final double pressure, final double humidity) {

this.temperature = temperature;

this.pressure = pressure;

this.humidity = humidity;

}

public double getTemperature() {

return temperature;

}

public double getPressure() {

return pressure;

}

public double getHumidity() {

return humidity;

}

@Override

public String toString() {

return temperature + "°\t\t" + pressure + "мм.рт.ст.\t\t" + humidity + "%";

}

}

// WeatherDay.java

package Task10;

import java.util.Arrays;

public class WeatherDay

{

public enum Time {

Night,

Morning,

Afternoon,

Evening

}

private Weather[] weather;

public WeatherDay(final Weather[] weather)

{

this.weather = weather;

}

public Weather getWeather(final Time dayTime)

{

return weather[dayTime.ordinal()];

}

@Override

public String toString() {

String outStr = "";

for (int i = 0; i < weather.length; i++)

outStr += String.format("%.7s\t\t- %s\n", Time.values()[i].toString(), weather[i].toString());

return outStr;

}

}

// WeekDay.java

package Task10;

public enum WeekDay {

Mounday,

Tuesday,

Wednesday,

Thursday,

Friday,

Saturday,

Sunday

}

// Task10.java

package Task10;

public class Task10

{

public static void main(final String[] args)

{

final WeatherDay[] weekWeather = {

new WeatherDay(new Weather[]{ // Mounday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

}),

new WeatherDay(new Weather[]{ // Tuesday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

}),

new WeatherDay(new Weather[]{ // Wednesday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

}),

new WeatherDay(new Weather[]{ // Thursday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

}),

new WeatherDay(new Weather[]{ // Friday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

}),

new WeatherDay(new Weather[]{ // Saturday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

}),

new WeatherDay(new Weather[]{ // Sunday

new Weather(11, 764, 76), // Night

new Weather(12, 755, 66), // Morning

new Weather(13, 711, 55), // Afternoon

new Weather(14, 714, 50) // Evening

})

};

for (int i = 0; i < WeekDay.values().length; i++)

System.out.println(WeekDay.values()[i].toString() + ":\n" + weekWeather[i].toString());

}

}

### Результат виконання на рисунку 10

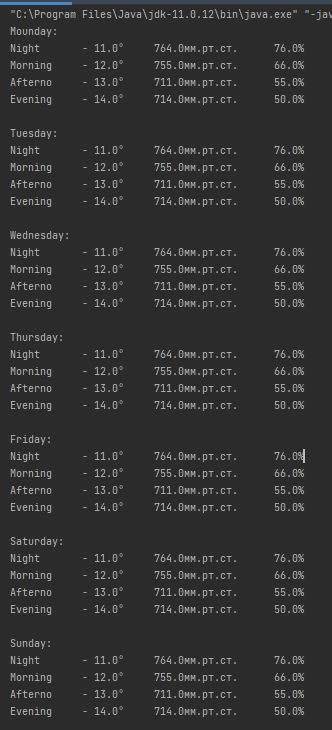


Рис. 10

## Завдання 11

### Умова

Реалізувати ієрархію класів системи "Таксі". Перевизначити equals() i

hashCode(). Одна машина може мати кількох водіїв. Додати клас

"Замовлення (Order)" в ієрархію

### Код

// Car.java

package Task11;

public class Car {

private Driver driver;

private String brand;

private int numberplate;

private String color;

public Car(final String brand, final int numberplate, final String colour ){

this.brand = brand;

this.numberplate = numberplate;

this.color = colour;

}

@Override

public boolean equals(final Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

final Car car = (Car) o;

if (numberplate != car.numberplate) return false;

if (driver != null ? !driver.equals(car.driver) : car.driver != null) return false;

return brand != null ? brand.equals(car.brand) : car.brand == null;

}

@Override

public int hashCode() {

int result = driver != null ? driver.hashCode() : 0;

result = 31 \* result + (brand != null ? brand.hashCode() : 0);

result = 31 \* result + numberplate;

return result;

}

public Driver getDriver() {

return driver;

}

public String getBrand() {

return brand;

}

public int getNumberplate() {

return numberplate;

}

public String getColor() {

return color;

}

public void setDriver(final Driver driver) {

this.driver = driver;

}

}

// Driver.java

package Task11;

public class Driver

{

private String name;

private boolean ready;

public Driver(final String name, final Boolean ready)

{

this.name = name;

this.ready = ready;

}

@Override

public boolean equals(final Object o)

{

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

final Driver driver = (Driver) o;

return name != null ? name.equals(driver.name) : driver.name == null;

}

@Override

public int hashCode()

{

return name != null ? name.hashCode() : 0;

}

public String getName()

{

return name;

}

public boolean isReady()

{

return ready;

}

public void setReady(final boolean ready)

{

this.ready = ready;

}

}

//Order.java

package Task11;

public class Order {

Car car;

private final double costkm = 50;

private double price;

private double way;

public void getPrice()

{

price = costkm \* way;

}

public void order (final Car car )

{

final var driver = car.getDriver();

if (driver.isReady())

{

System.out.println("Order started!\n" +"Price = " + price +"\n" + "Your driver is " + driver.getName() +"." +

" Wait for " + car.getColor() +" "+ car.getBrand() + " with numberplate: " + car.getNumberplate() + ".\n");

driver.setReady(false);

}

else

System.out.println("This driver is busy!\n" + "choose another or wait if all busy.\n");

}

public void endOrder (final Car car)

{

final var driver = car.getDriver();

if (!driver.isReady())

{

driver.setReady(true);

System.out.println("Order canceled!\n");

}

}

public static void main(final String[] args)

{

final Car car1 = new Car("Mazda", 4321 ,"red");

final Car car2 = new Car("Ford",3252, "white");

final Car car3 = new Car("Subaru",7890,"lightblue");

final Driver d1 = new Driver("John",true);

final Driver d2 = new Driver("Steve",true);

final Driver d3 = new Driver("Juzo",true);

car1.setDriver(d1);

car2.setDriver(d3);

final Order o1 = new Order();

final Order o2 = new Order();

o1.way = 20;

o1.getPrice();

o1.order(car1);

o2.way = 30;

o2.getPrice();

o2.order(car1);

o1.endOrder(car1);

car1.setDriver(d2);

o2.way = 15;

o2.getPrice();

o2.order(car1);

o1.way = 12;

o1.getPrice();

o1.order(car2);

}

}

### Результат виконання на рисунку 11

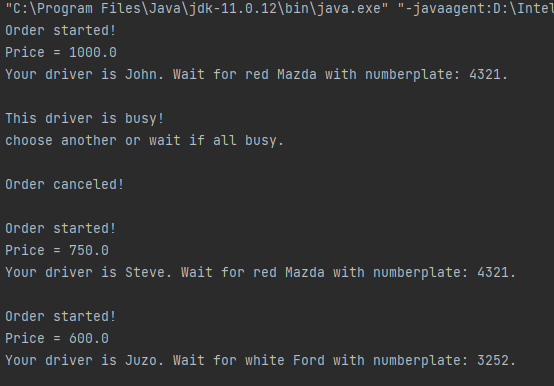


Рис. 11

## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/APP_Lab_1>

## Висновок

Ознайомився із основами мови програмування Java.