Міністерство освіти і науки України

Національний університет «Львівська політехніка»



# Звіт

Про виконання

Лабораторної роботи № 2

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 ООП мови програмування Java та з Java

Collections..

## Завдання 1

### Умова

Написати ієрархію класів та продемонструвати порядок ініціалізації

полів класу. Використати конструктор та статичні блоки ініціалізації в усіх

класах (мінімум 3 класи)

### Код

// Figure.java

package Task1;

public class Figure {

public final double pi = 3.1415; // constanta 4yslo pi

protected String colour = "white";

public double findPerimetr(){

System.out.println("Perimetr!");

return 0;

};

public double findArea(){

System.out.println("Area!");

return 0;

};

public void setColour(String colour){

this.colour = colour;

};

public String toString(){

return ("Колiр:" + colour);

}

}

// Circle.java

package Task1;

public class Circle extends Figure {

private double r; // radius

@Override

public double findArea() {

return pi \* r \* r ;

}

@Override

public double findPerimetr() {

return 2 \* pi \*r;

}

@Override

public void setColour(String colour) {

this.colour = colour;

}

public String toString(){

return ("Колiр: " + colour + " " + "Радiус: " + r + " ");

}

Circle(double r){

this.r = r;

}

}

// Rectangle.java

package Task1;

public class Rectangle extends Figure {

private double a; //storony

private double b;

@Override

public double findArea() {

return a\*b;

}

@Override

public double findPerimetr() {

return (a+b)\*2;

}

@Override

public void setColour(String colour) {

this.colour = colour;

}

public String toString(){

return ("Колiр: " + colour + " " + "Сторона а: " + a + " " + "Сторона b: " + b + " ");

}

Rectangle(double a, double b){

this.a = a;

this.b = b;

}

}

// Square.java

package Task1;

public class Square extends Figure {

private double a; // storona

@Override

public double findArea() {

return a\*a;

}

@Override

public double findPerimetr() {

return a\*4;

}

@Override

public void setColour(String colour) {

this.colour = colour;

}

public String toString(){

return ("Колiр: " + colour + " " + "Сторона а: " + a + " ");

}

Square(double a){

this.a = a;

}

}

// Triangle.java

package Task1;

public class Triangle extends Figure {

private double a; // storony

private double b;

private double c;

private boolean isnue = false;

public void checkIsnue(){

if ((a+b)>c && (a+c)>b && (b+c)>a )

isnue = true;

else

System.out.println("Такого трикутника не iснує!");

}

@Override

public double findArea() {

double p = (a+b+c)/2; // poluperymetr

if (isnue == true)

return Math.sqrt(p \* (p - a) \* (p - b) \* (p - c)); // formula Gerona

else {

System.out.println("Помилка змiнiть значеня сторон");

return -1;}

}

@Override

public double findPerimetr() {

if (isnue == true)

return a+b+c;

else {

System.out.println("Помилка змiнiть значеня сторон");

return -1;}

}

@Override

public void setColour(String colour) {

super.setColour(colour);

}

public String toString(){

return ("Колiр: " + colour + " " + "Сторона а: " + a + " " + "Сторона b: " + b + " " + "Сторона с: " + c );

}

Triangle(double a,double b,double c){

this.a = a;

this.b = b;

this.c = c;

}

}

// Task1.java

package Task1;

public class Task1 {

public static void main(String[] args) {

Triangle t1 = new Triangle(1,2,3);

Triangle t2 = new Triangle(3,4,5);

System.out.println(t1.toString());

t1.checkIsnue();

t1.setColour("black");

System.out.println(t1.findArea());

System.out.println("");

t2.checkIsnue();

System.out.println(t2.toString());

System.out.println(t2.findArea());

System.out.println(t2.findPerimetr());

System.out.println("");

Circle c1 = new Circle(5);

System.out.println(c1.toString());

}

}

### Результат виконання на рисунку 1

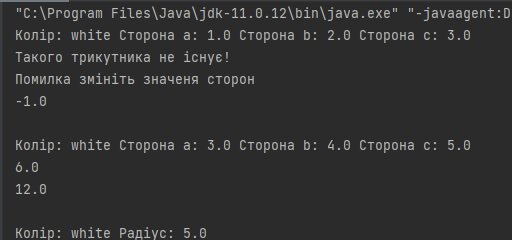


Рис. 1

## Завдання 2

### Умова

Реалізувати структуру даних “стек” на основі однозв’язного списку.

Структура даних повинна надавати можливість обходу використовуючи

for-each.

### Код

// Stack.java

package Task2;

import java.util.Iterator;

public class Stack<Type> implements Iterable<Type>

{

private StackElement<Type> head;

public void push(Type value)

{

StackElement<Type> a = new StackElement<Type>();

a.value = value;

if (head == null) {

head = a;

} else {

a.next = head;

head = a;

}

}

void printStack()

{

StackElement<Type> t = head;

while (t != null)

{

System.out.print(t.value + " ");

t = t.next;

}

}

void pop()

{

if (head == null)

return;

if (head.next == null)

head = null;

else

head = head.next;

}

@Override

public Iterator<Type> iterator()

{

return new StackIterator<Type>(head);

}

}

// StackElement.java

package Task2;

public class StackElement<Type>

{

StackElement<Type> next = null;

Type value;

}

// StackIterator.java

package Task2;

import java.util.Iterator;

public class StackIterator<Type> implements Iterator<Type>

{

private StackElement<Type> cur;

public StackIterator(StackElement<Type> start)

{

cur = start;

}

@Override

public boolean hasNext()

{

return cur != null;

}

@Override

public Type next()

{

var val = cur.value;

cur = cur.next;

return val;

}

@Override

public void remove()

{

throw new UnsupportedOperationException();

}

}

//Task2.java

package Task2;

public class Task2

{

public static void main(String[] args)

{

Stack<String> a = new Stack<String>();

a.push("1");

a.push("2");

a.push("7");

a.push("4");

a.printStack();

System.out.println("");

a.pop();

a.pop();

a.printStack();

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

for (var el: a)

System.out.print(el + " ");

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

}

}

### Результат виконання на рисунку 2

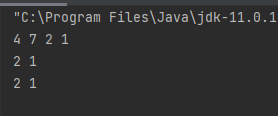


Рис. 2

## Завдання 3

### Умова

Створити ієрархію класів Студент, Оцінка та Предмет.

### Код

// Subject.java

package Task3;

public class Subject

{

private String name;

public Subject(String name)

{

this.name = name;

}

public String getName() {

return name;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (!(o instanceof Subject) && !(o instanceof String)) return false;

return name.equals(o instanceof String ? o : ((Subject)o).name);

}

@Override

public int hashCode() {

return name.hashCode();

}

@Override

public String toString() {

return name;

}

}

// Mark.java

package Task3;

public class Mark

{

private int value;

private Subject subject;

public Mark(int value, Subject subject)

{

this.value = value;

this.subject = subject;

}

@Override

public boolean equals(Object o)

{

if (this == o) return true;

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

Mark mark = (Mark) o;

return subject != null ? subject.equals(mark.subject) : mark.subject == null;

}

@Override

public int hashCode()

{

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

}

@Override

public String toString() {

return subject + ": " + value;

}

public int getValue() {

return value;

}

public String getSubjectName()

{

return subject.getName();

}

}

// Student.java

package Task3;

import java.util.HashSet;

import java.util.Set;

import java.util.stream.Collectors;

public class Student

{

private Set<Mark> marks;

private String id;

private String name,

surname,

lastname;

private char classChar;

private int numberOfClass;

public Student(String id, String name, String surname, String lastname, char classChar, int numberOfClass, Set<Mark> marks)

{

this.id = id;

this.name = name;

this.surname = surname;

this.lastname = lastname;

this.classChar = classChar;

this.numberOfClass = numberOfClass;

this.marks = marks;

}

public Student(String id, String name, String surname, String lastname, char classChar, int numberOfClass)

{

this(id, name, surname, lastname, classChar, numberOfClass, new HashSet<Mark>());

}

public String getName() {

return name;

}

public String getSurname() {

return surname;

}

public String getLastname() {

return lastname;

}

public char getClassChar() {

return classChar;

}

public int getNumberOfClass() {

return numberOfClass;

}

public String getId() {

return id;

}

@Override

public String toString() {

return name + " " + surname + " " + lastname + "\n\t" +

marks.stream().map(mr -> mr.toString()).collect(Collectors.joining("\n\t")) +

"\n\tAvarage: " + getAvarageMark();

}

public void addMark(Mark mark)

{

marks.add(mark);

}

public double getAvarageMark()

{

return marks.stream().mapToInt(mark -> mark.getValue()).average().getAsDouble();

}

public Mark getMarkBySubject(Subject subject)

{

return marks.stream().filter(mark -> subject.equals(mark.getSubjectName())).findFirst().get();

}

}

// Task3.java

package Task3;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public class Task3

{

public static void main(String[] args) {

final Set<Subject> subjects = new HashSet<Subject>(Arrays.asList(

new Subject("Фізика"),

new Subject("Англійська мова"),

new Subject("Математика")

));

final var sb\_phythics = subjects.stream().filter(el -> el.getName() == "Фізика").findFirst().get();

final var sb\_english = subjects.stream().filter(el -> el.getName() == "Англійська мова").findFirst().get();

final var sb\_math = subjects.stream().filter(el -> el.getName() == "Математика").findFirst().get();

final Set<Student> students = new HashSet<Student>(Arrays.asList(

new Student("AAAAAA", "Serhii", "Stepanchuk", "Tarasovych", 'A', 11, new HashSet<Mark>(Arrays.asList(

new Mark(80, sb\_phythics),

new Mark(100, sb\_english),

new Mark(90, sb\_math)

))),

new Student("BBBBBB","Maria", "Gryziak", "Lybomirivna", 'B', 10, new HashSet<Mark>(Arrays.asList(

new Mark(80, sb\_phythics),

new Mark(100, sb\_english),

new Mark(90, sb\_math)

))),

new Student("CCCCCC","Andrii", "Sobor", "Vasylivych", 'B', 11, new HashSet<Mark>(Arrays.asList(

new Mark(80, sb\_phythics),

new Mark(100, sb\_english),

new Mark(90, sb\_math)

)))

));

for (var student : students)

{

System.out.println(student);

}

}

}

### Результат виконання на рисунку 3

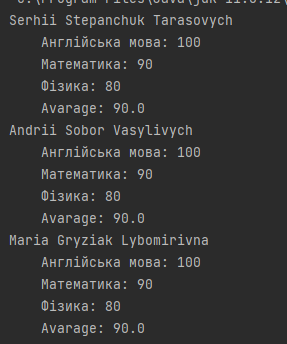


Рис. 3

## Завдання 4

### Умова

Створити List студентів з оцінками з трьох предметів (Фізика,

Математика, Англійська мова). Завдання:

● Написати метод для добавлення нового предмету усім студентам.

● Написати метод для виведення студентів з усіма їхніми оцінками.

● Вивести дані студента з найвищим середнім балом

● Додати до класу Студент унікальний ідентифікатор.

● Вивести Map де ключем буде унікальний ідентифікатор, а значенням

- оцінки із заданого предмету (предмет задається через аргумент

методу)

### Код

// StudentList.java

package Task4;

import Task3.Mark;

import Task3.Student;

import Task3.Subject;

import java.util.\*;

import java.util.stream.Collectors;

public class StudentList

{

private Set<Subject> subjects;

private Map<String, Student> students;

public StudentList(Set<Subject> subjects, Student[] students)

{

this.subjects = subjects;

this.students = new HashMap<String, Student>();

for (var student : students)

this.students.put(student.getId(), student);

}

public void addSubject(Subject subject)

{

subjects.add(subject);

}

public void addSubjectWithMarks(Subject subject, Map<String, Integer> marks)

{

addSubject(subject);

for (var mark : marks.entrySet())

students.get(mark.getKey()).addMark(new Mark(mark.getValue(), subject));

}

public Student getStudentById(String id)

{

return students.get(id);

}

@Override

public String toString() {

return students.values().stream().map(st -> st.toString()).collect(Collectors.joining("\n"));

}

public Student getBestStudent()

{

var arr = new ArrayList<Student>(students.values());

arr.sort(Comparator.comparingDouble(Student::getAvarageMark).reversed());

return arr.get(0);

}

public Map<String, Integer> getStudentsMarksBySubject(Subject subject)

{

var resMap = new HashMap<String, Integer>();

for (var student: students.entrySet())

resMap.put(student.getKey(), student.getValue().getMarkBySubject(subject).getValue());

return resMap;

}

}

// Task4.java

package Task4;

import Task3.\*;

import java.util.\*;

public class Task4

{

public static void main(String[] args) {

final var sb\_phythics = new Subject("Фізика");

final var sb\_english = new Subject("Англійська мова");

final var sb\_math = new Subject("Математика");

final Set<Subject> subjects = new HashSet<Subject>(Arrays.asList(

sb\_phythics,

sb\_english,

sb\_math

));

var studentList = new StudentList(subjects, new Student[] {

new Student("AAAAAA", "Serhii", "Stepanchuk", "Tarasovych", 'A', 11, new HashSet<Mark>(Arrays.asList(

new Mark(80, sb\_phythics),

new Mark(100, sb\_english),

new Mark(90, sb\_math)

))),

new Student("BBBBBB","Maria", "Gryziak", "Lybomirivna", 'B', 10, new HashSet<Mark>(Arrays.asList(

new Mark(89, sb\_phythics),

new Mark(100, sb\_english),

new Mark(90, sb\_math)

))),

new Student("CCCCCC","Andrii", "Sobor", "Vasylivych", 'B', 11, new HashSet<Mark>(Arrays.asList(

new Mark(99, sb\_phythics),

new Mark(100, sb\_english),

new Mark(90, sb\_math)

)))

});

System.out.println(studentList);

studentList.addSubjectWithMarks(new Subject("Географія"), Map.of(

"AAAAAA", 80,

"BBBBBB", 90,

"CCCCCC", 100)

);

System.out.println("\nAfter add new subject: ");

System.out.println(studentList);

System.out.println("\nBest student: ");

System.out.println(studentList.getBestStudent());

System.out.println("\nPhysic marks: ");

for (var entry : studentList.getStudentsMarksBySubject(sb\_phythics).entrySet())

System.out.println(entry.getKey() + ":" + entry.getValue());

}

}

### Результат виконання на рисунках 4 та 5

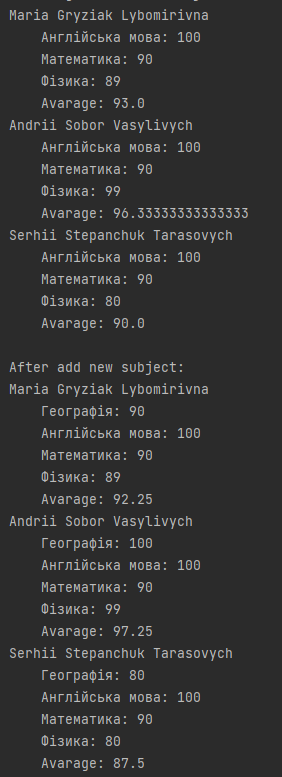


Рис. 4

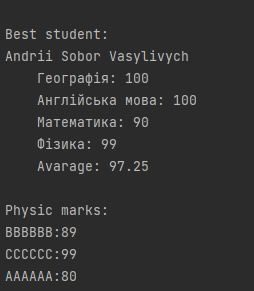


Рис. 5

## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/APP_Lab_2>

## Висновок

Ознайомився із основами ООП мови програмування Java та з Java Collections.