---Câu1

public class Main {

public static void main(String[] args) {

Circle circle = new Circle(25.5f); //tạo một đối tượng hình tròn có bán kính 25.5

Rectangle rectangle = new Rectangle(5.0, 12.0); //tạo một đối tượng hình chữ nhật với chiều rộng 5, chiều dài 12

circle.showInfo(); //phương thức hiển thị thông tin của hình tròn

rectangle.showInfo(); //phương thức hiển thị thông tin của hình chữ nhật

}

}

interface IShape {

double PI=3.1416;

void showInfo(); //hàm hiển thị thông tin của hình

float getArea(); //phương thức tính diện tích

float getPerimeter(); //phương thức tính chu vi

}

class Circle implements IShape {

private float radius;

public Circle(float radius) { // hàm khởi tạo

this.radius = radius;

}

public float getRadius(){ //hàm lấy giá trị

return radius;

}

public void setRadius (float radius) { // hàm thay đổi giá trị

this.radius = radius;

}

@Override

public float getArea() {

return (float) (PI \* radius \* radius); // tính diện tích hình tròn

}

@Override

public float getPerimeter () {

return (float) (2 \* PI \* radius); // tính chu vi hình tròn

}

@Override

public void showInfo() { //hàm in ra thông tin hình tròn

System.out.printf("Circle with radius %.2f has area: %.2f and perimeter: %.2f\n", radius, getArea(), getPerimeter());

}

}

class Rectangle implements IShape {

public double width;

public double length;

public Rectangle(double width, double length) {

this.width = width;

this.length = length;

}

@Override

public float getArea() { //tính diện tích

return (float) (width \* length);

}

@Override

public float getPerimeter() { // tính chu vi

return (float) (2\*(width+length));

}

@Override

public void showInfo() { //hiển thị ra màn hình

System.out.printf("Rectangle with width %.2f and length %.2f has area: %.2f and perimeter: %.2f%n", width, length, getArea(), getPerimeter());

}

}

---câu 2

import java.io.\*;

public class Main {

public static void main(String[] args) {

String filename = "int.txt";

// Ghi 4 số nguyên vào file bằng DataOutputStream

try (DataOutputStream dos = new DataOutputStream(new FileOutputStream(filename))) {

dos.writeInt(10);

dos.writeInt(20);

dos.writeInt(30);

dos.writeInt(40);

} catch (IOException e) {

System.out.println("Error writing to file: " + e.getMessage());

}

// Đọc 4 số nguyên từ file bằng DataInputStream

int[] numbers = new int[4];

try (DataInputStream dis = new DataInputStream(new FileInputStream(filename))) {

for (int i = 0; i < 4; i++) {

numbers[i] = dis.readInt();

}

} catch (IOException e) {

System.out.println("Error reading from file: " + e.getMessage());

}

// Tính tổng 2 số nguyên đầu tiên

int sumFirstTwo = numbers[0] + numbers[1];

// In tổng các số nguyên ra màn hình

System.out.println("Tổng 2 số đầu tiên: " + sumFirstTwo);

int totalSum = 0;

for (int num : numbers) {

totalSum += num;

}

System.out.println("Tổng 4 số nguyên: " + totalSum);

}

}

---Câu3

import java.util.Random;

public class Main {

public static void main(String[] args) {

Thread thread1 = new Thread(() -> {

try {

while (true) {

long currentTimeMillis = System.currentTimeMillis();

String time = String.format("%tT", currentTimeMillis);

System.out.println("Thời gian hiện tại: " + time);

Thread.sleep(1000);

}

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

});

///Tạo thread2

Thread thread2 = new Thread(() -> {

Random random = new Random();

int randomNumber = random.nextInt(100) + 1;

System.out.println("Số ngẫu nhiên (1-100): " + randomNumber);

});

////Bắt đầu chạy

thread1.start();

thread2.start();

try {

thread2.join();

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

}

}

--Câu4

import java.util.\*;

public class Main {

public static void main(String[] args) {

Random random = new Random();

LinkedList<Integer> list = new LinkedList<>();

for (int i = 0; i < 8; i++) {

list.add(random.nextInt(100));

}

System.out.println("Danh sách ban đầu: " + list);

if (list.size() >= 2) {

list.add(2, 25);

} else {

list.add(25);

}

System.out.println("Sau khi thêm 25 vào vị trí số 2: " + list);

// Xóa các phần tử có giá trị < 30

list.removeIf(n -> n < 30);

System.out.println("Sau khi xóa phần tử < 30: " + list);

// Tìm và in ra vị trí của tất cả phần tử có giá trị từ 40 -> 80

System.out.print("Vị trí phần tử có giá trị từ 40 đến 80: ");

for (int i = 0; i < list.size(); i++) {

int value = list.get(i);

if (value >= 40 && value <= 80) {

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

}

}

}

}

---Câu 5:

// Functional Interface

@FunctionalInterface

interface Sorter {

void sort(int[] arr);

}

public class Main {

public static void main(String[] args) {

// Dùng biểu thức Lambda để sắp xếp tăng dần

Sorter ascendingSorter = (int[] arr) -> {

for (int i = 0; i < arr.length - 1; i++) {

for (int j = i + 1; j < arr.length; j++) {

if (arr[i] > arr[j]) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

};

// Khởi tạo mảng

int[] arr = {4, 2, 7, 1, 9, 5};

// Gọi phương thức sort để sắp xếp mảng

ascendingSorter.sort(arr);

// In ra mảng sau khi sắp xếp

System.out.print("Mảng sau khi sắp xếp: ");

for (int num : arr) {

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

}

}

}