**Exercise 1: Inventory Management System**

class Product {

String name;

int quantity;

double price;

Product(String name, int quantity, double price) {

this.name = name;

this.quantity = quantity;

this.price = price;

}

}

class Inventory {

Product[] products = new Product[100];

int size = 0;

void addProduct(Product p) {

products[size++] = p;

}

void displayInventory() {

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

System.*out*.println(products[i].name + " " + products[i].quantity + " " + products[i].price);

}

}

public static void main(String[] args) {

Inventory inv = new Inventory();

inv.addProduct(new Product("Pen", 10, 5.0));

inv.addProduct(new Product("Book", 5, 20.0));

inv.displayInventory();

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 2: E-commerce Platform Search Function**

class SearchItem {

String name;

double price;

SearchItem(String name, double price) {

this.name = name;

this.price = price;

}

}

class SearchProduct {

SearchItem[] items;

SearchProduct() {

items = new SearchItem[3];

items[0] = new SearchItem("Laptop", 50000);

items[1] = new SearchItem("Mobile", 25000);

items[2] = new SearchItem("Tablet", 15000);

}

void search(String key) {

boolean found = false;

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

if (items[i].name.equalsIgnoreCase(key)) {

System.*out*.println(items[i].name + " " + items[i].price);

found = true;

break;

}

}

if (!found) {

System.*out*.println("Product not found");

}

}

public static void main(String[] args) {

SearchProduct sp = new SearchProduct();

sp.search("Mobile");

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 3: Sorting Customer Orders**

class Order {

int orderId;

double amount;

Order(int orderId, double amount) {

this.orderId = orderId;

this.amount = amount;

}

}

class SortOrders {

public static void main(String[] args) {

Order[] orders = new Order[3];

orders[0] = new Order(101, 500.0);

orders[1] = new Order(102, 300.0);

orders[2] = new Order(103, 700.0);

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

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

if (orders[j].amount > orders[j + 1].amount) {

Order temp = orders[j];

orders[j] = orders[j + 1];

orders[j + 1] = temp;

}

}

}

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

System.*out*.println(orders[i].orderId + " " + orders[i].amount);

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 4: Employee Management System**

class EmployeeDetail {

int id;

String name;

String position;

double salary;

EmployeeDetail(int id, String name, String position, double salary) {

this.id = id;

this.name = name;

this.position = position;

this.salary = salary;

}

}

class EmployeeList {

public static void main(String[] args) {

EmployeeDetail[] emp = new EmployeeDetail[3];

emp[0] = new EmployeeDetail(1, "Arun", "Manager", 50000);

emp[1] = new EmployeeDetail(2, "Divya", "Developer", 40000);

emp[2] = new EmployeeDetail(3, "Kumar", "Tester", 30000);

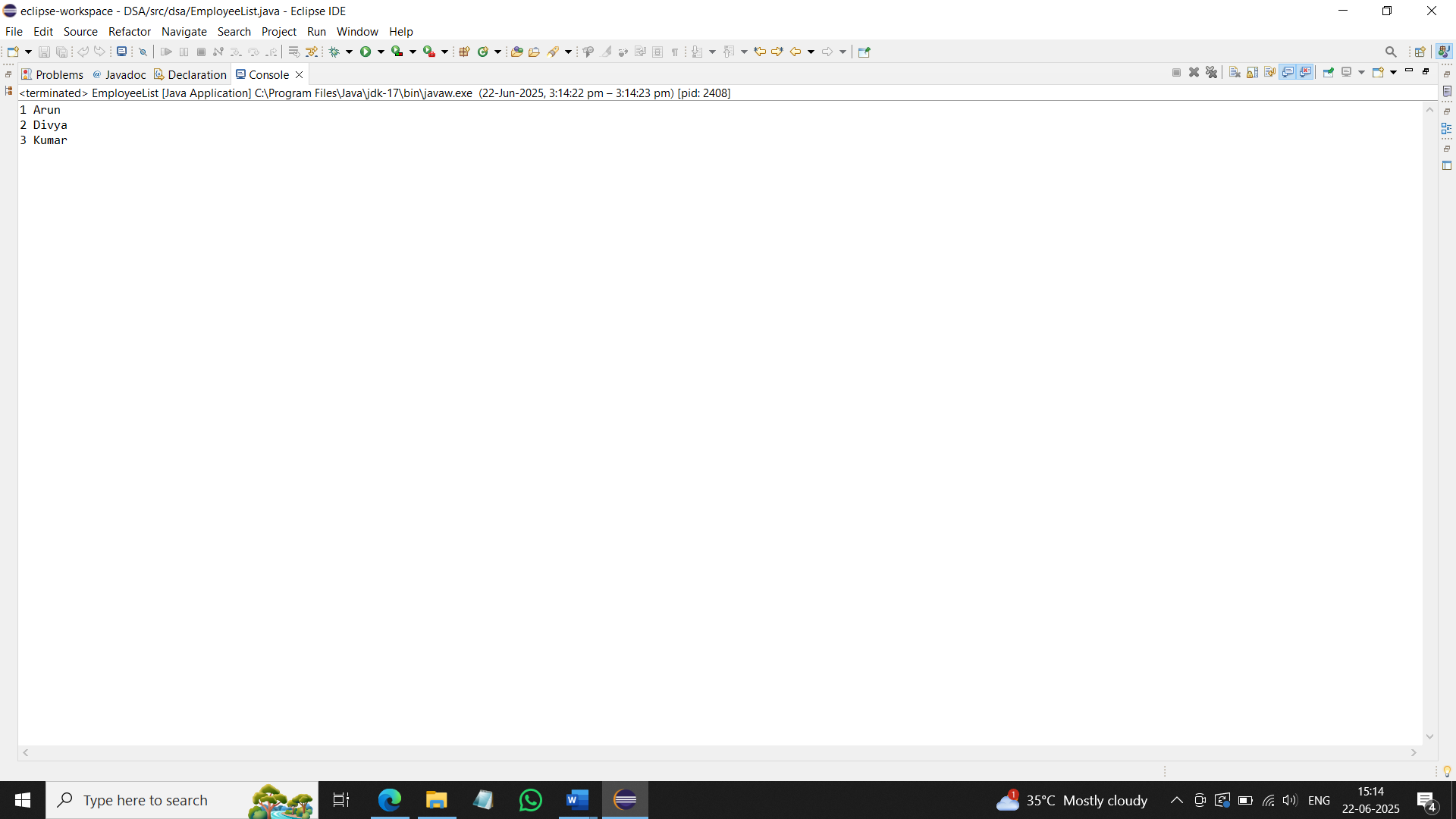
for (int i = 0; i < emp.length; i++) {

System.*out*.println(emp[i].id + " " + emp[i].name);

}

}

}



**Exercise 5: Task Management System**

class TaskDetail {

String title;

String dueDate;

TaskDetail(String title, String dueDate) {

this.title = title;

this.dueDate = dueDate;

}

}

class TaskDisplay {

public static void main(String[] args) {

TaskDetail[] tasks = new TaskDetail[3];

tasks[0] = new TaskDetail("Submit Report", "2025-06-30");

tasks[1] = new TaskDetail("Fix Bugs", "2025-07-05");

tasks[2] = new TaskDetail("Client Meeting", "2025-07-01");

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

System.*out*.println(tasks[i].title + " " + tasks[i].dueDate);

}

}

}

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 6: Library Management System**

class BookInfo {

String title;

String author;

BookInfo(String title, String author) {

this.title = title;

this.author = author;

}

}

class LibrarySearch {

public static void main(String[] args) {

BookInfo[] books = new BookInfo[3];

books[0] = new BookInfo("Wings of Fire", "A.P.J. Abdul Kalam");

books[1] = new BookInfo("The Alchemist", "Paulo Coelho");

books[2] = new BookInfo("Ignited Minds", "A.P.J. Abdul Kalam");

String search = "A.P.J. Abdul Kalam";

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

if (books[i].author.equalsIgnoreCase(search)) {

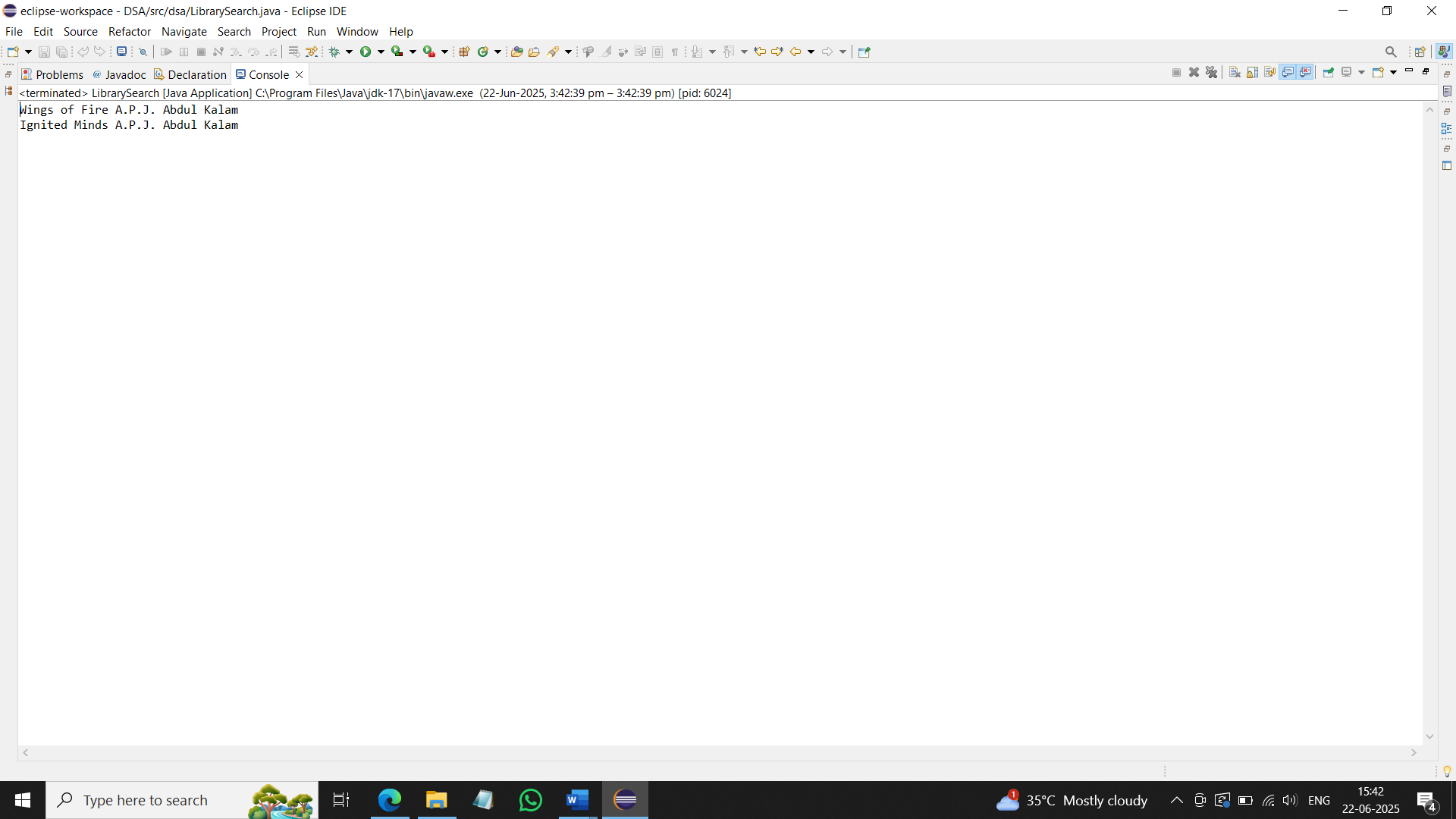
System.*out*.println(books[i].title + " " + books[i].author);

}

}

}

}



**Exercise 7: Financial Forecasting**

class RecursiveRevenueAverage {

static double totalRevenue(double[] revenue, int index) {

if (index == revenue.length) {

return 0;

}

return revenue[index] + *totalRevenue*(revenue, index + 1);

}

public static void main(String[] args) {

double[] revenue = {25000.5, 30000.75, 28000.0, 35000.25, 27000.0};

double total = *totalRevenue*(revenue, 0);

double average = total / revenue.length;

System.*out*.println(average);

}

}

