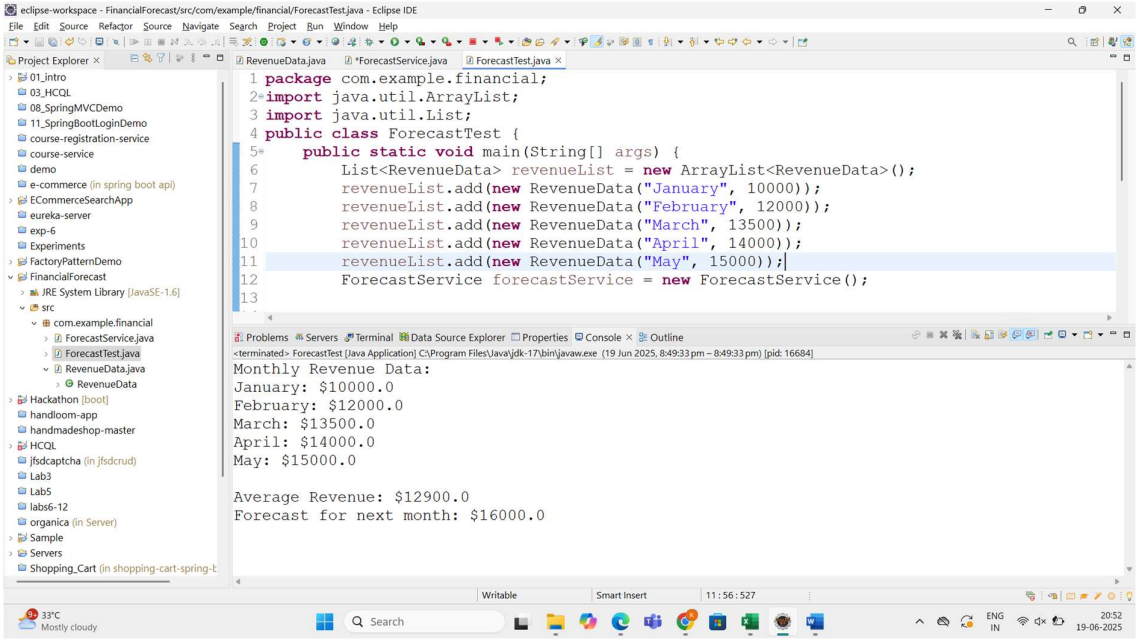
Exercise 7: Financial Forecasting

```
package com.example.financial;
public class RevenueData {
    private String month;
    private double revenue;

    public RevenueData(String month, double revenue) {
        this.month = month;
        this.revenue = revenue;
    }

    public String getMonth() {
        return month;
    }

    public double getRevenue() {
        return revenue;
    }

    @Override
    public String toString() {
        return month + ": $" + revenue;
    }
}
package com.example.financial;

import java.util.List;

public class ForecastService {
```

```
    public double calculateAverage(List<RevenueData> revenues) {
        double sum = 0.0;
        for (RevenueData r : revenues) {
            sum += r.getRevenue();
        }
        return revenues.isEmpty() ? 0.0 : sum / revenues.size();
    }



    public double forecastNextMonth(List<RevenueData> revenues) {
        int size = revenues.size();
        if (size < 2) {
            return calculateAverage(revenues);
        }

        double last = revenues.get(size - 1).getRevenue();
        double secondLast = revenues.get(size - 2).getRevenue();
        double trend = last - secondLast;

        return last + trend;
    }
}
package com.example.financial;

import java.util.ArrayList;
import java.util.List;
```

```
public class ForecastTest {
    public static void main(String[] args) {
        List<RevenueData> revenueList = new ArrayList<RevenueData>();

        revenueList.add(new RevenueData("January", 10000));
        revenueList.add(new RevenueData("February", 12000));
        revenueList.add(new RevenueData("March", 13500));
        revenueList.add(new RevenueData("April", 14000));
        revenueList.add(new RevenueData("May", 15000));

        ForecastService forecastService = new ForecastService();



        System.out.println("Monthly Revenue Data:");
        for (RevenueData r : revenueList) {
            System.out.println(r);
        }



        double avg = forecastService.calculateAverage(revenueList);
        System.out.println("\nAverage Revenue: $" + avg);



        double forecast = forecastService.forecastNextMonth(revenueList);
        System.out.println("Forecast for next month: $" + forecast);
    }
}
```

OUTPUT:


eclipse-workspace - FinancialForecast/src/com/example/financial/ForecastTest.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Project Explorer
01_intro
03_HCQL
08_SpringMVCDemo
11_SpringBootLoginDemo
course-registration-service
course-service
demo
e-commerce (in spring boot api)
ECommerceSearchApp
eureka-server
exp-6
Experiments
FactoryPatternDemo
FinancialForecast
JRE System Library [JavaSE-1.6]
src
com.example.financial
ForecastService.java
ForecastTest.java
RevenueData.java
RevenueData
Hackathon [boot]
handloom-app
handmadeshop-master
HCQL
jfsdcaptcha (in jfsdcrud)
Lab3
Lab5
labs6-12
organica (in Server)
Sample
Servers
Shopping_Cart (in shopping-cart-spring-b
RevenueData.java
*ForecastService.java
ForecastTest.java
1 package com.example.financial;
2 import java.util.ArrayList;
3 import java.util.List;
4 public class ForecastTest {
5 public static void main(String[] args) {
6 List<RevenueData> revenueList = new ArrayList<RevenueData>();
7 revenueList.add(new RevenueData("January", 10000));
8 revenueList.add(new RevenueData("February", 12000));
9 revenueList.add(new RevenueData("March", 13500));
10 revenueList.add(new RevenueData("April", 14000));
11 revenueList.add(new RevenueData("May", 15000));
12 ForecastService forecastService = new ForecastService();
13
Problems Servers Terminal Data Source Explorer Properties Console Outline
<terminated> ForecastTest [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (19 Jun 2025, 8:49:33 pm – 8:49:33 pm) [pid: 16684]
Monthly Revenue Data:
January: $10000.0
February: $12000.0
March: $13500.0
April: $14000.0
May: $15000.0
Average Revenue: $12900.0
Forecast for next month: $16000.0
Writable
Smart Insert
11 : 56 : 527
33°C
Mostly cloudy
Search
ENG
IN
20:52
19-06-2025