**Week - 1. Design principles & Patterns**

**Exercise 1: Implementing the Singleton Pattern**

**What is Singleton Pattern *?***

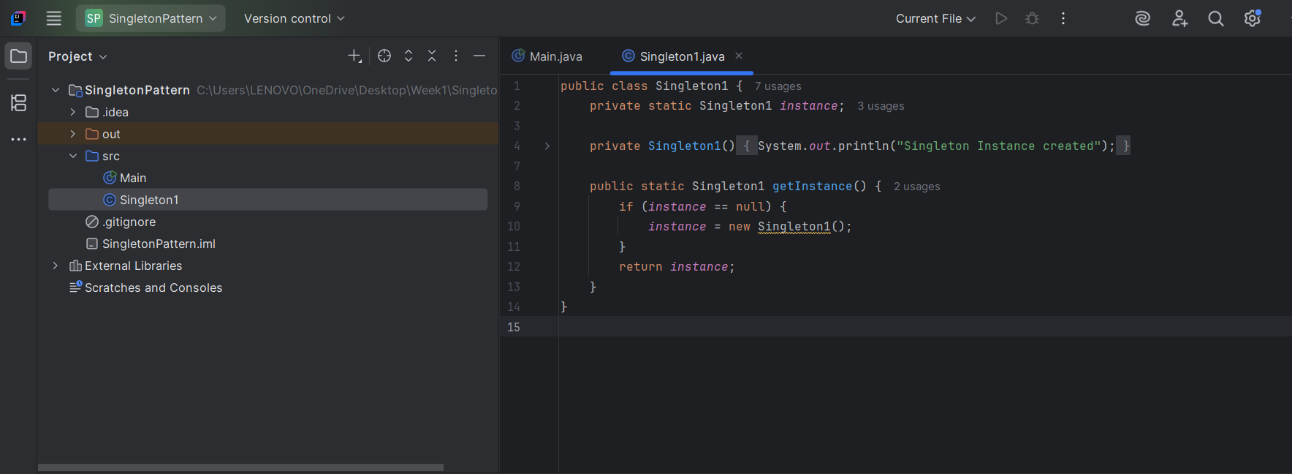
**Ans *:***

**The singleton pattern ensures that a *class* has "only 1 instance" throughout the application and providing a global access point to the instance**

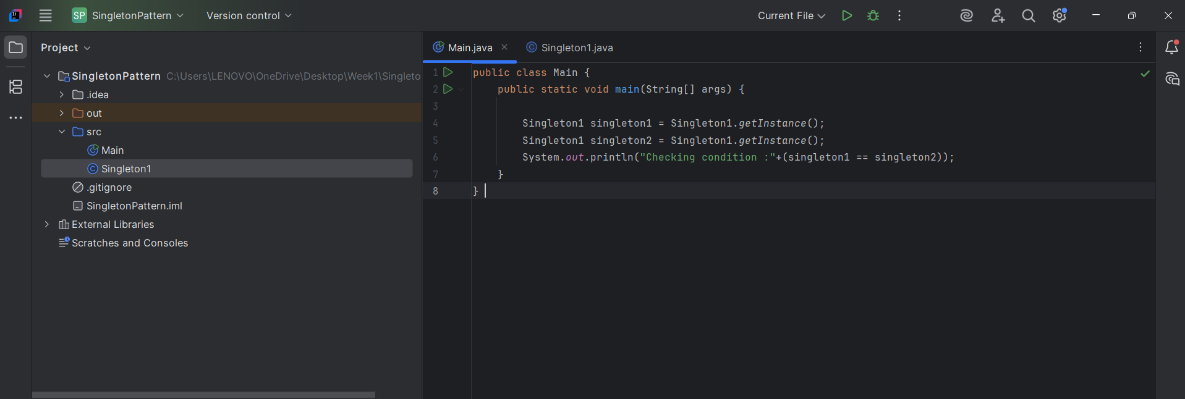
**Disadvantage:**

**Not Thread safe but we can counter that to using the different approaches**

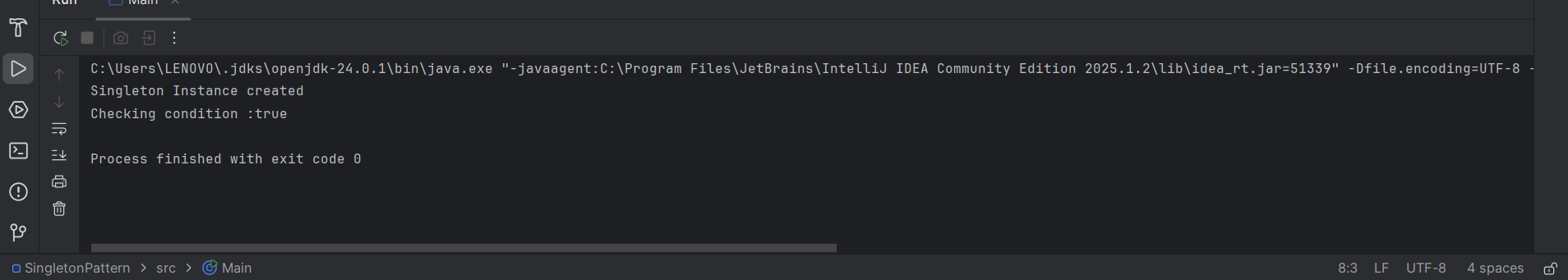
**Singleton Class**



**Main Class**



**Output:**

****

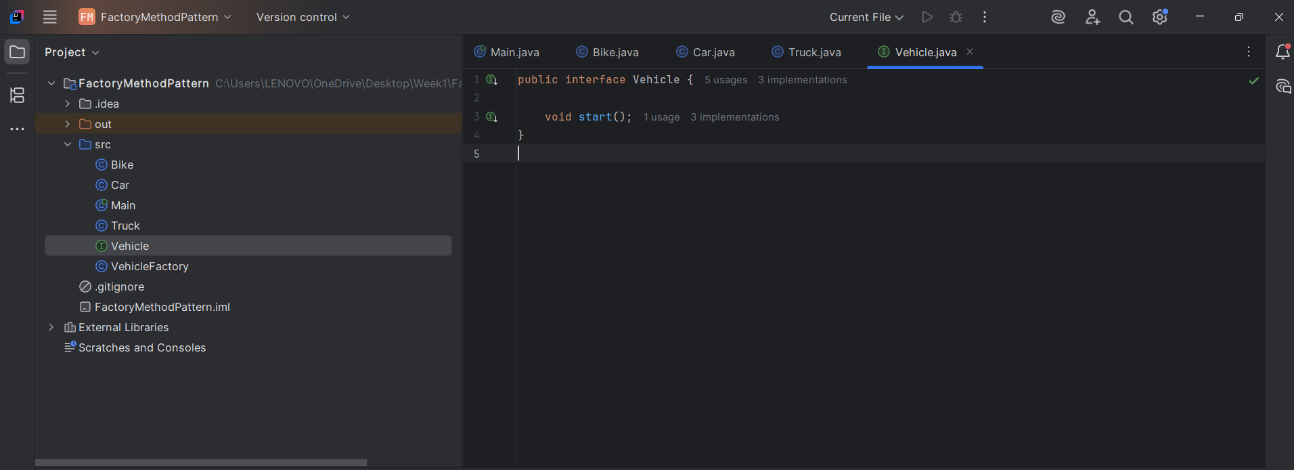
**Exercise 2: Implementing the Factory Method Pattern**

**What is Factory Method*?***

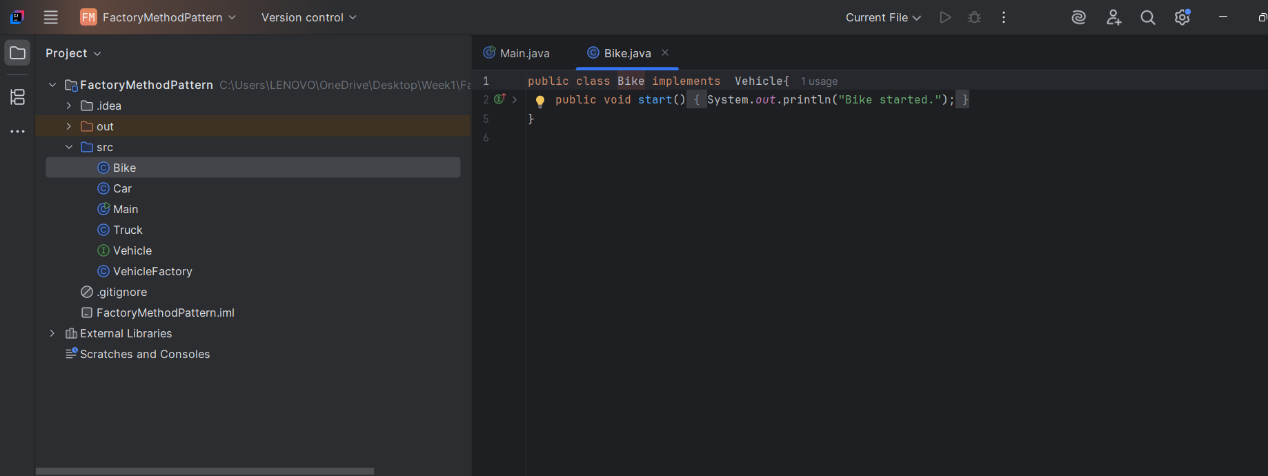
**It helps us create objects without exposing the object creation logic to the client. *The* client just asks the Factory, and the factory returns the correct object.**

**Example for Factory Method is Vehicle**

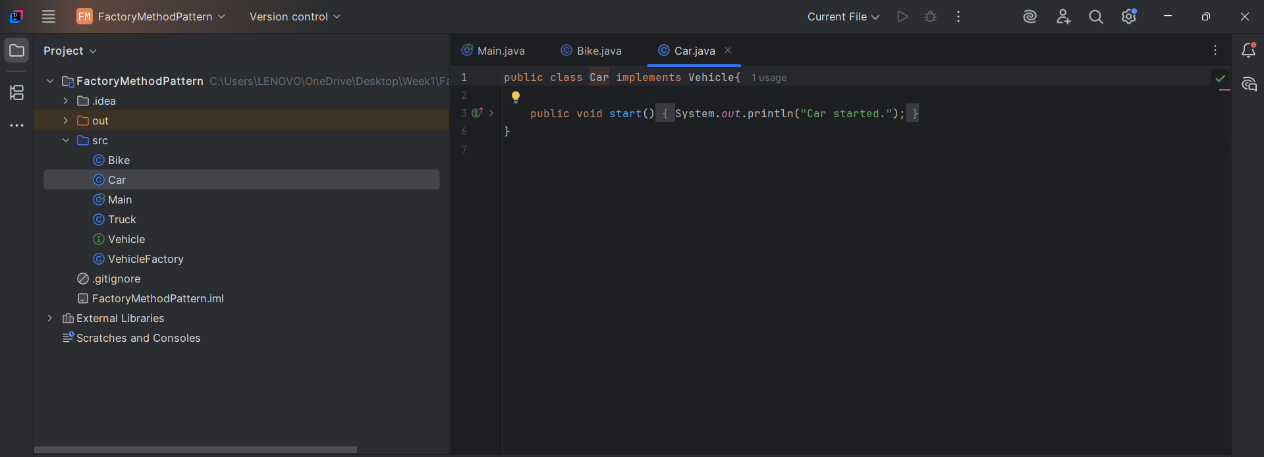
**Vehicle Interface:**

****

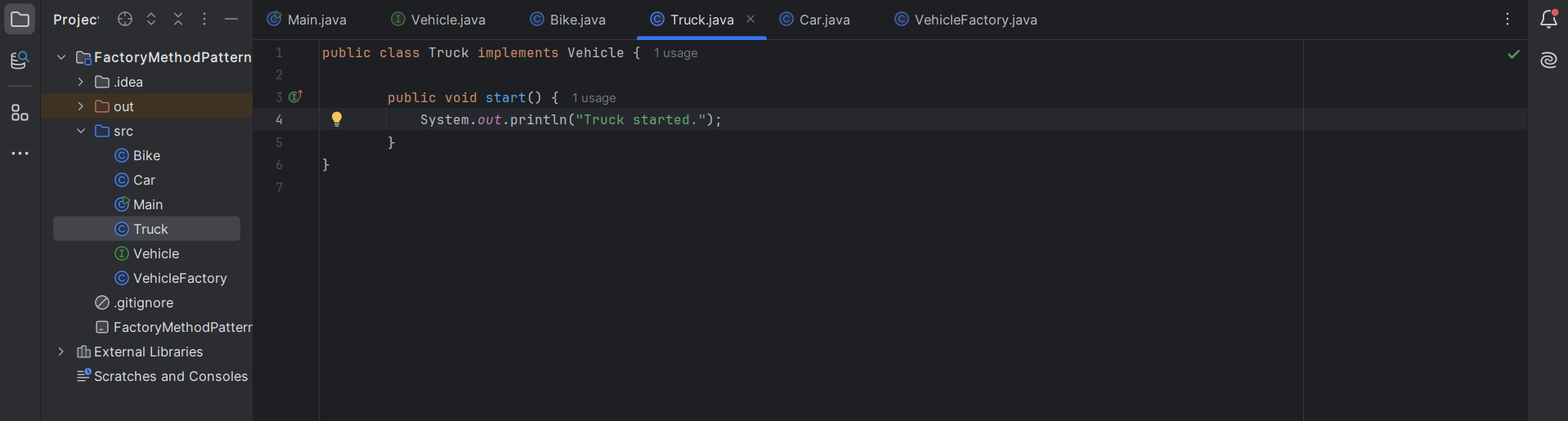
**Bike Class:**

****

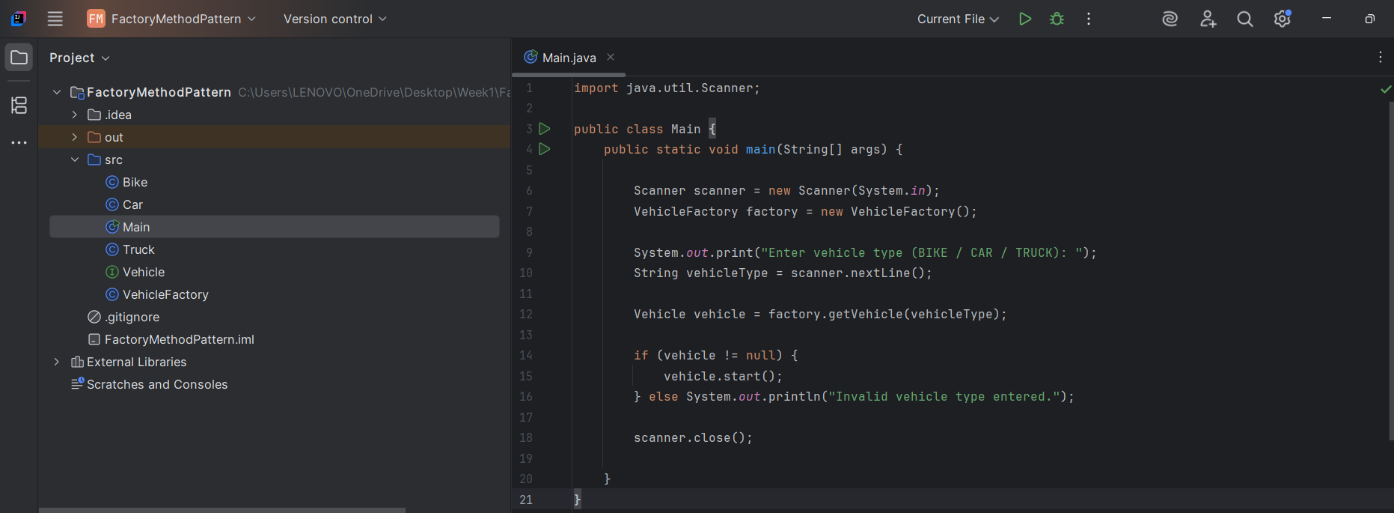
**Car Class:**

****

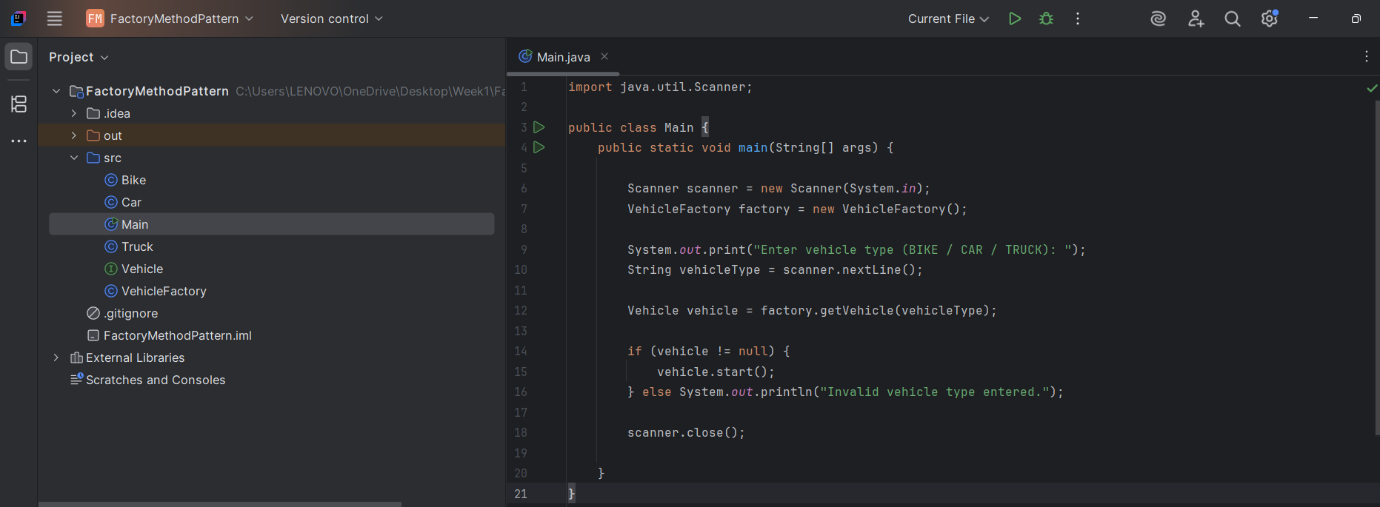
**Truck Class:**

****

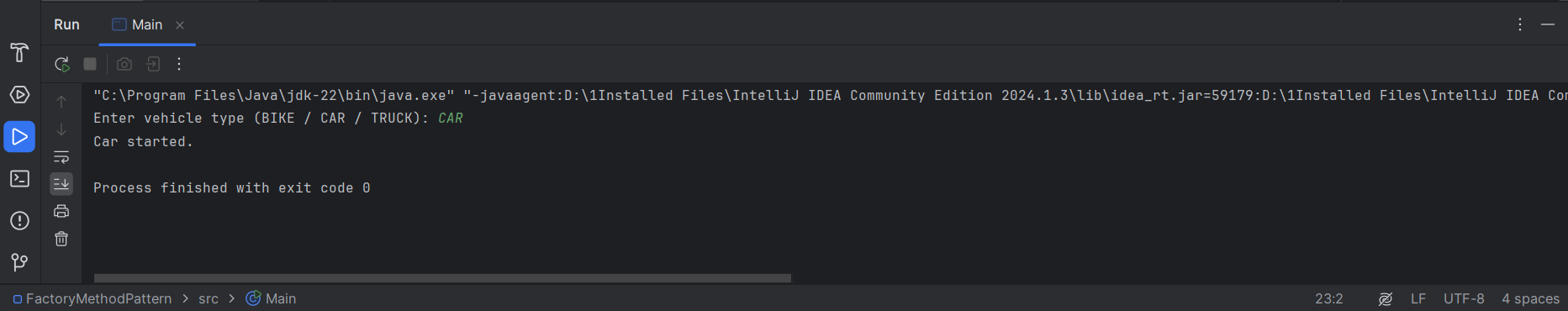
**VehicleFactory Class:**

****

**Main Class:**

****

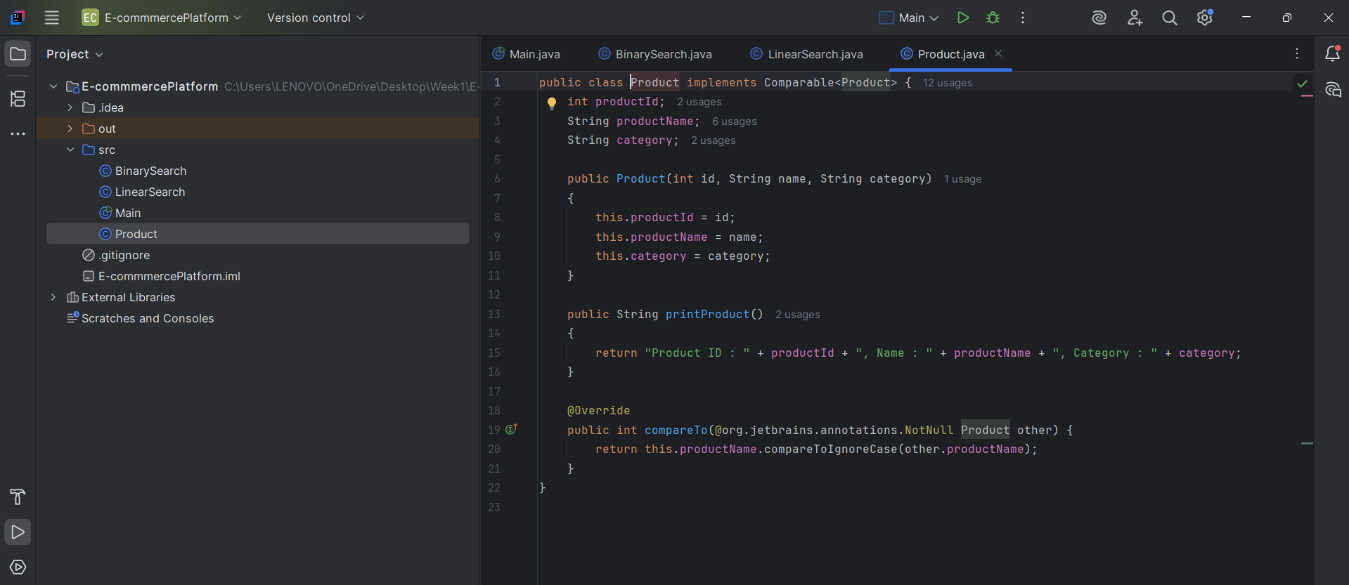
**Output:**



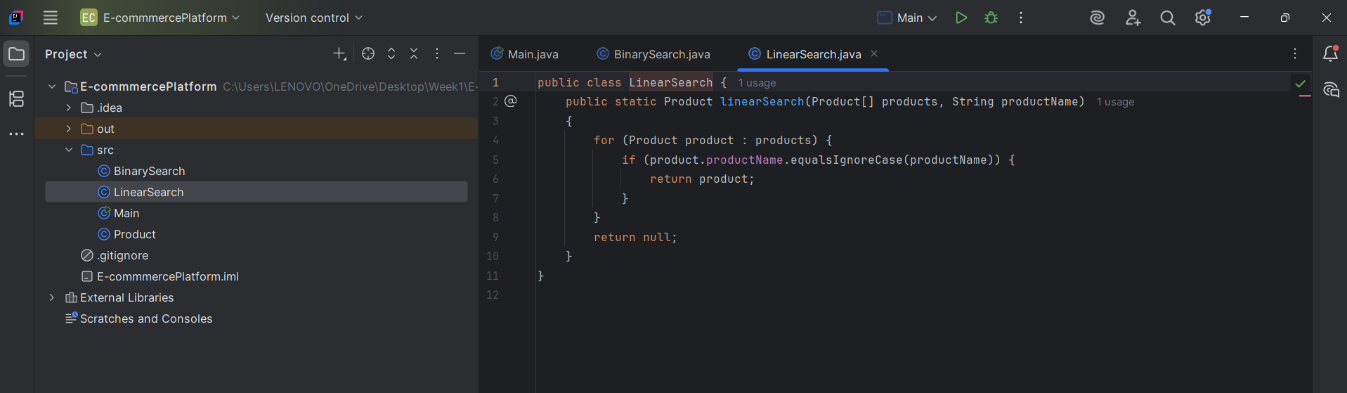
**2.Algorithms\_Data Structures**

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

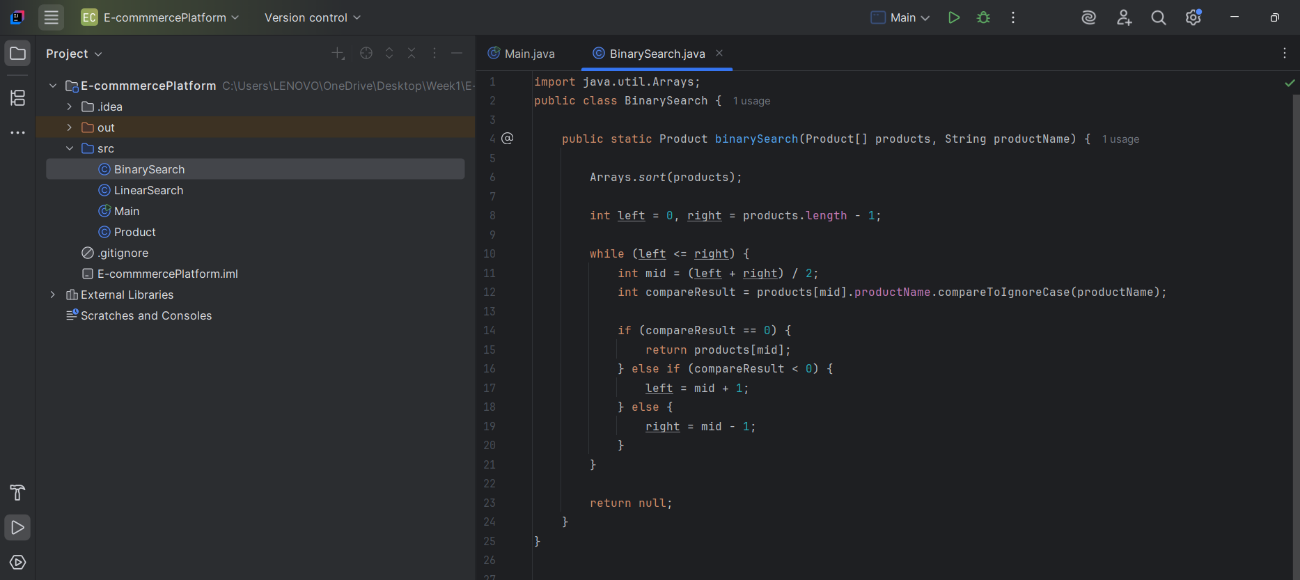
**Product Class:**

****

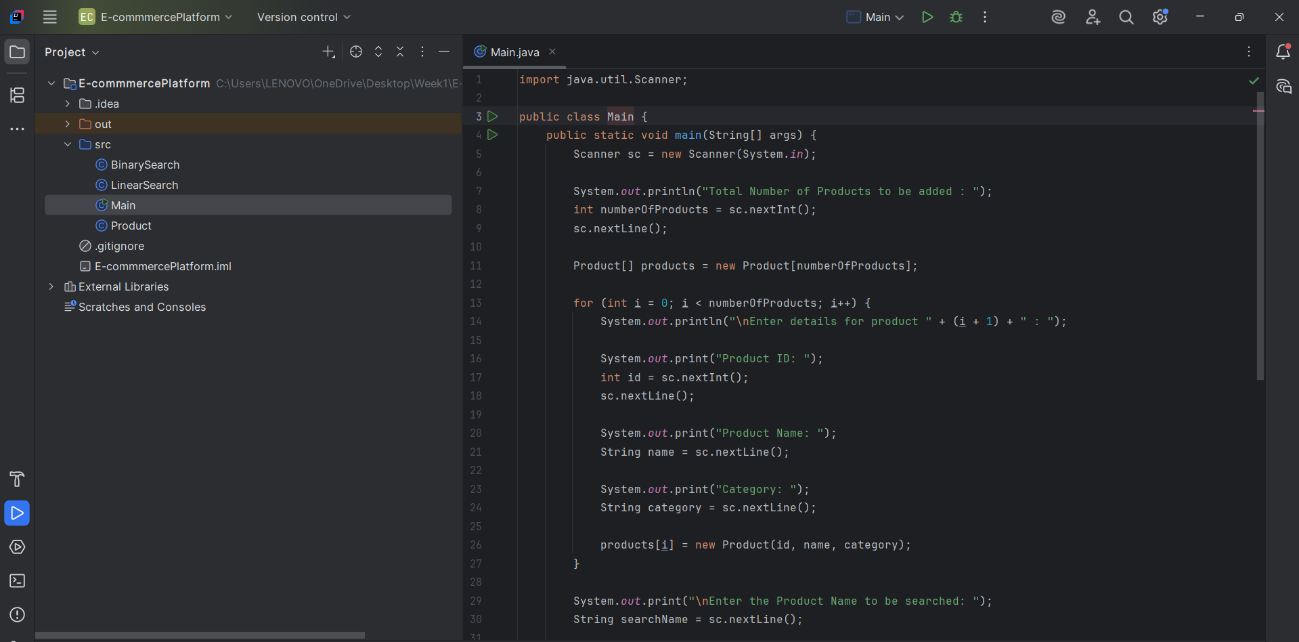
**LinearSearch Class:**

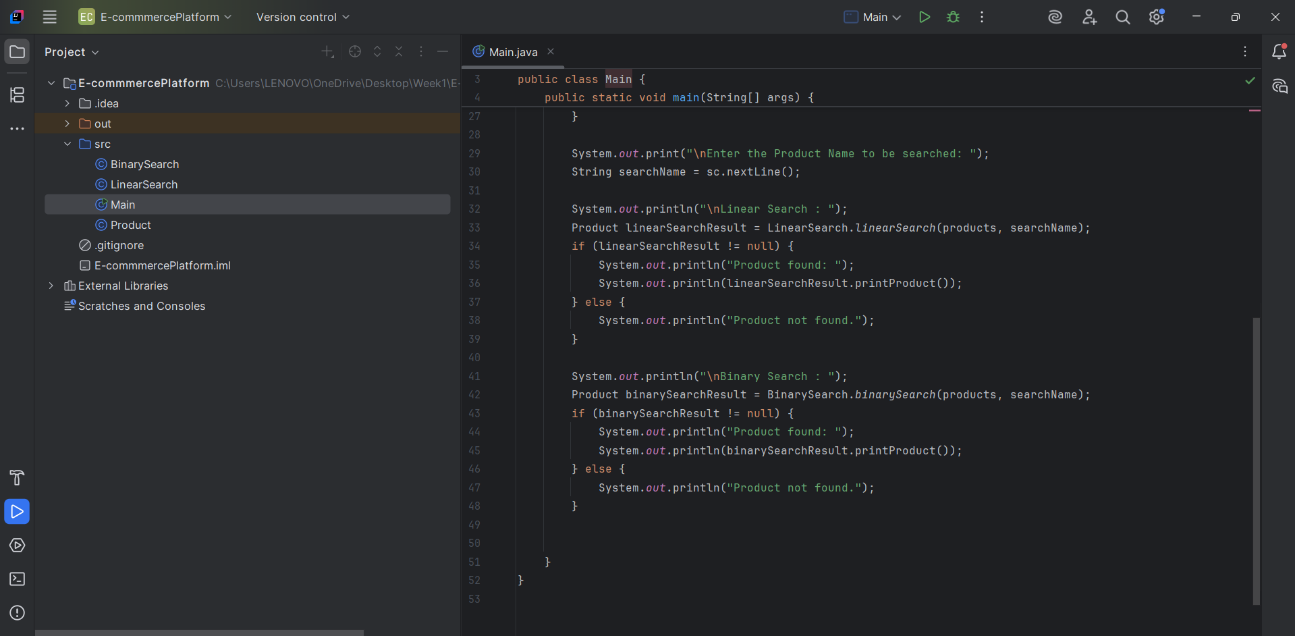
****

**BinarySearch Class:**

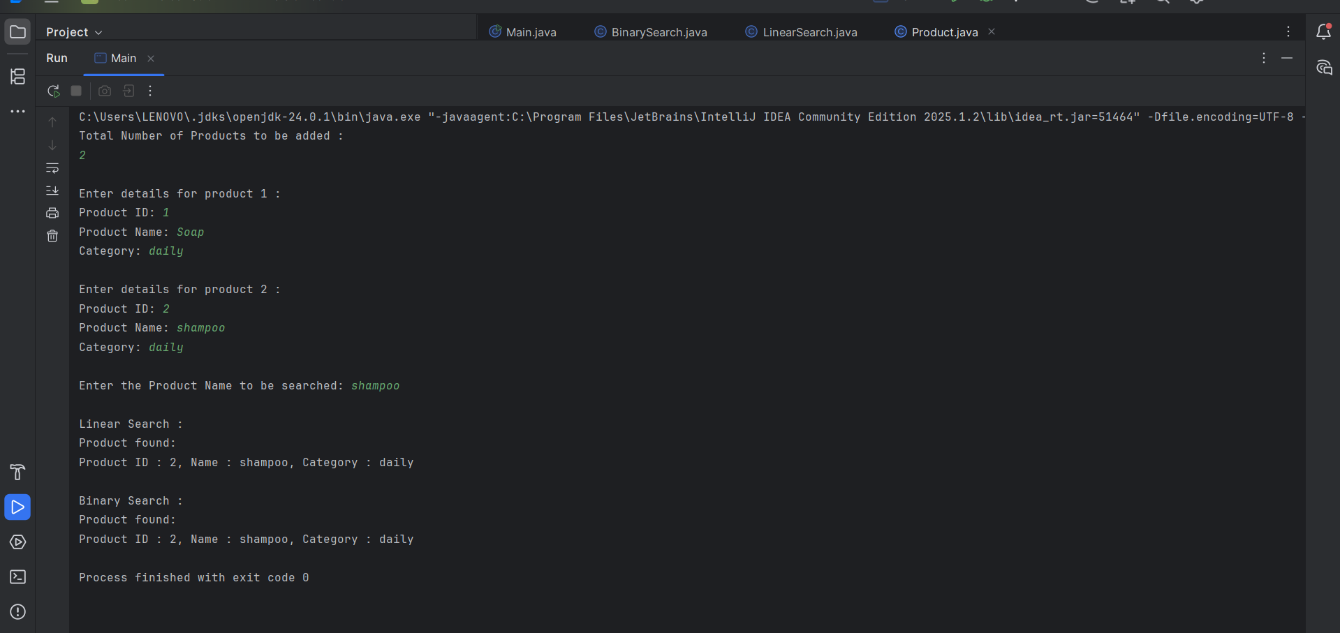
****

**Main Class:**

****

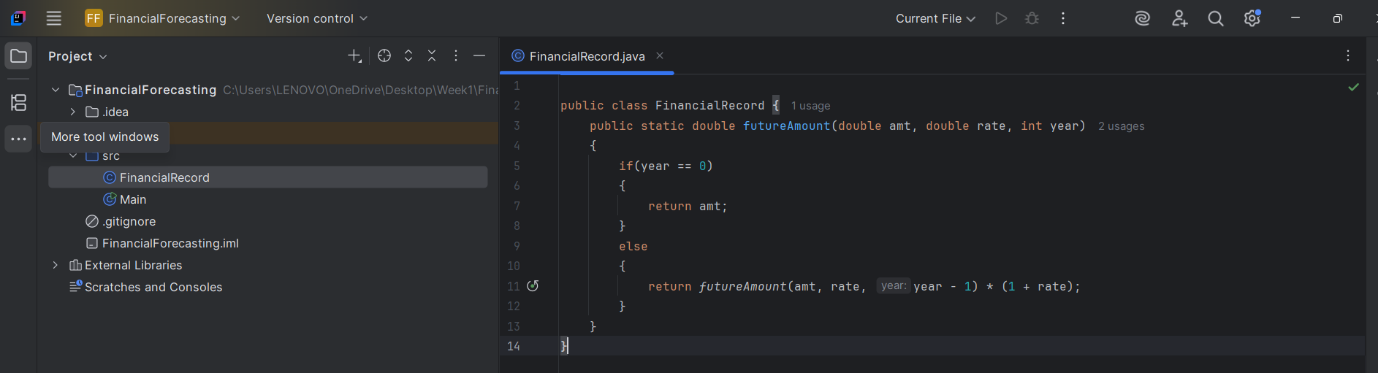


**Output:**

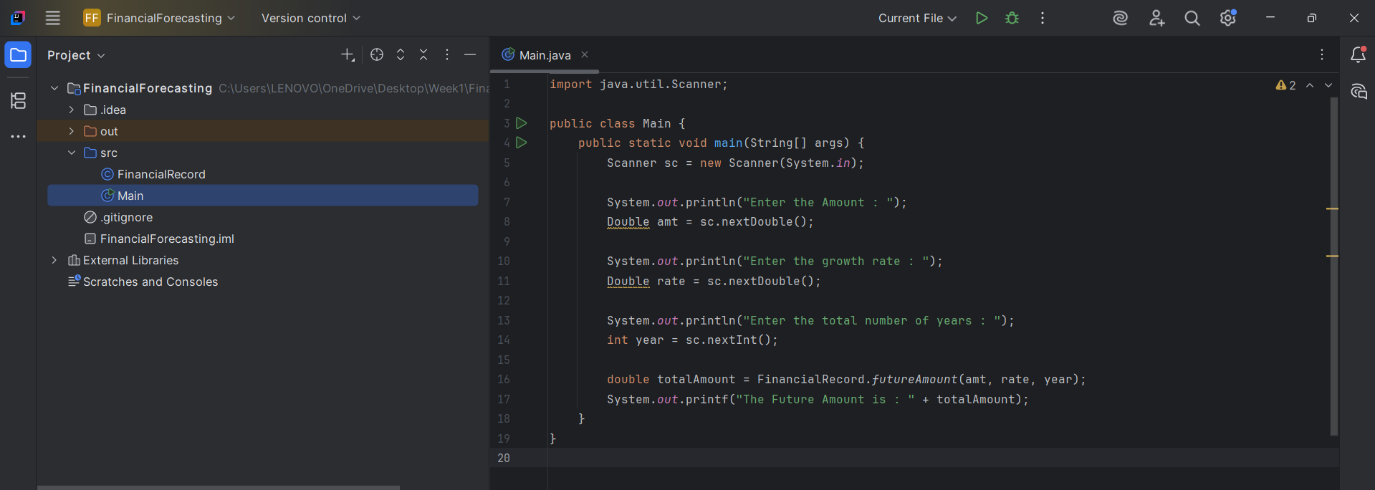
****

**Exercise 7:** **Financial Forecasting**

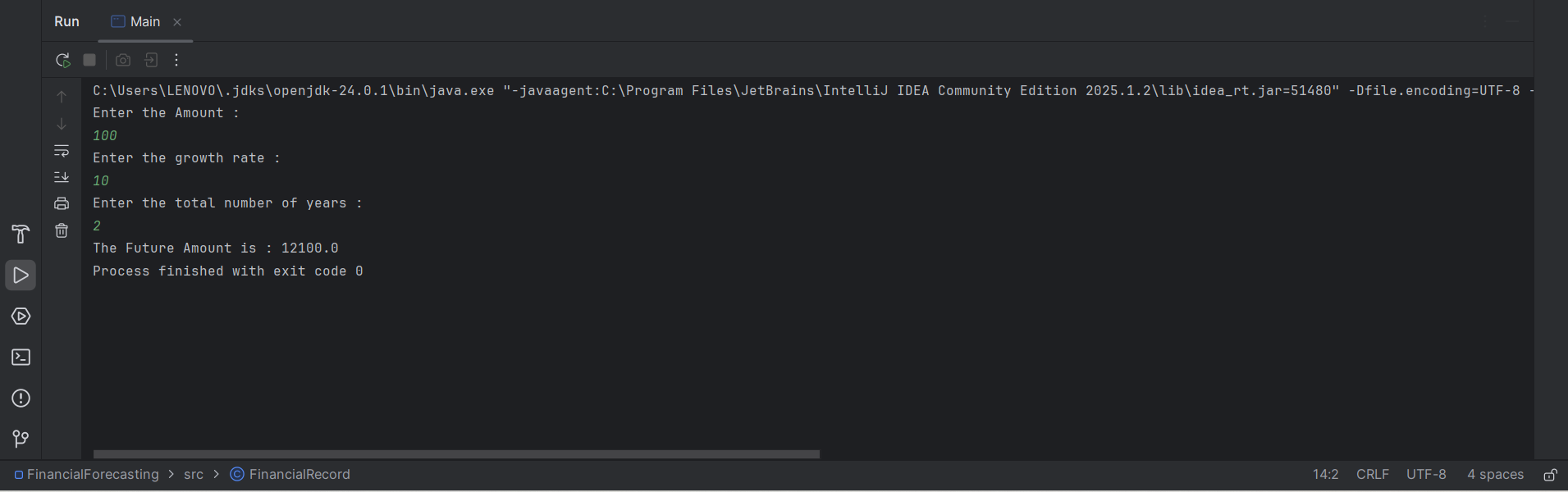
**FinancialRecord Class:**

****

**Main Class:**

****

**Output:**

**[](../OneDrive/Desktop/Week1)**

**Folder:** <https://drive.google.com/drive/folders/10Z_y-9v-elKOn0DHK1-sLEMkivaRzgDU?usp=drive_link>