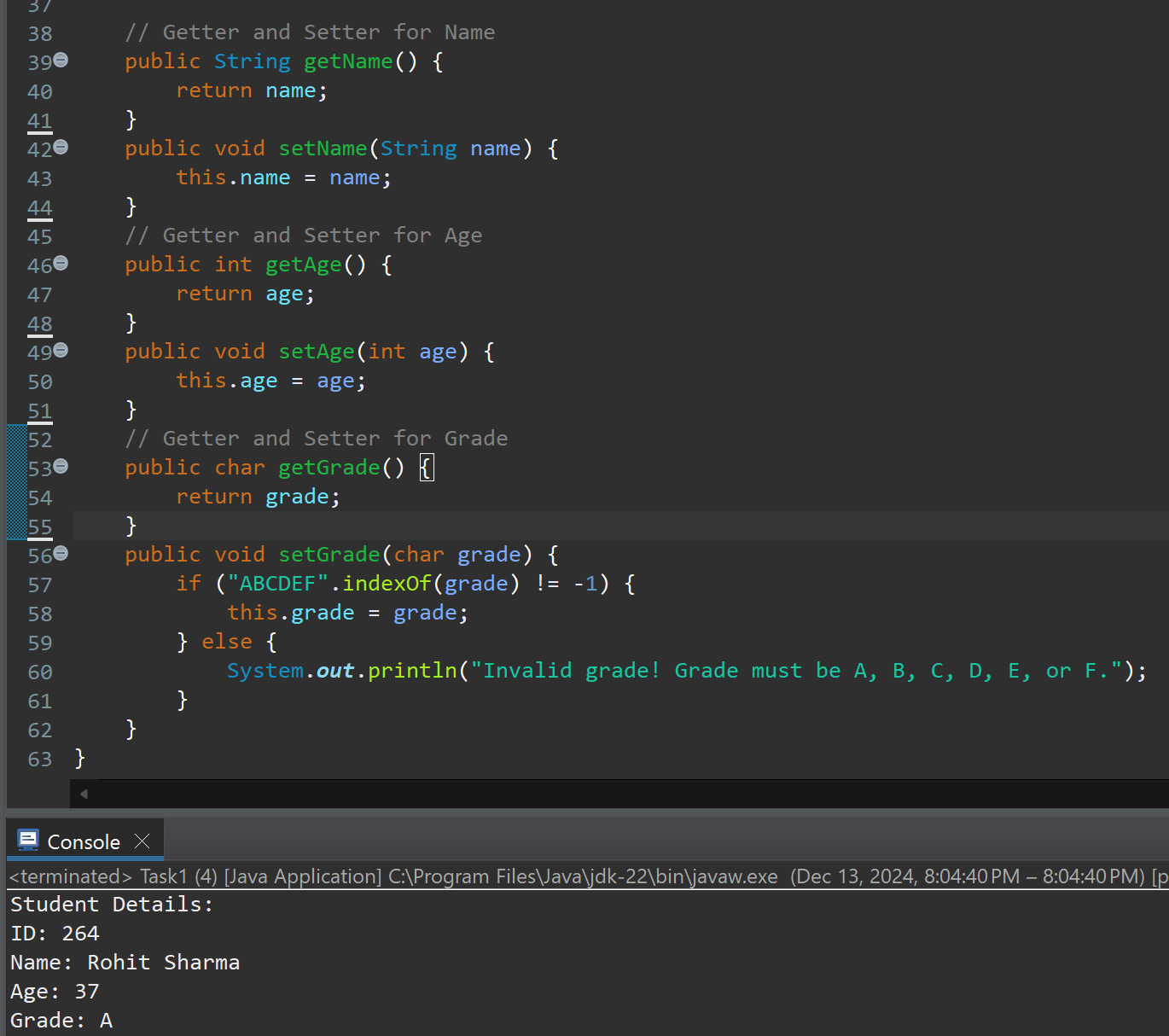
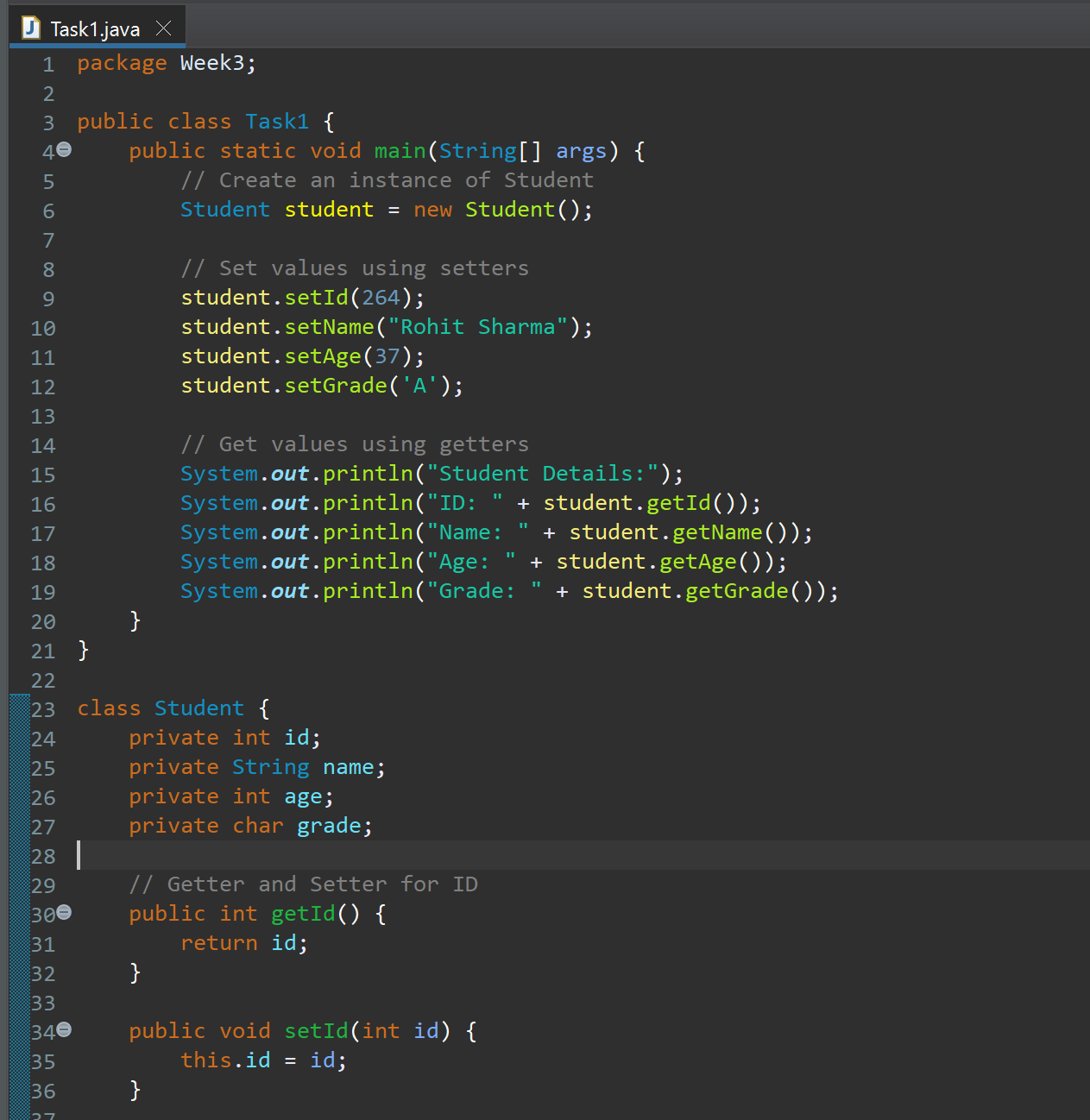
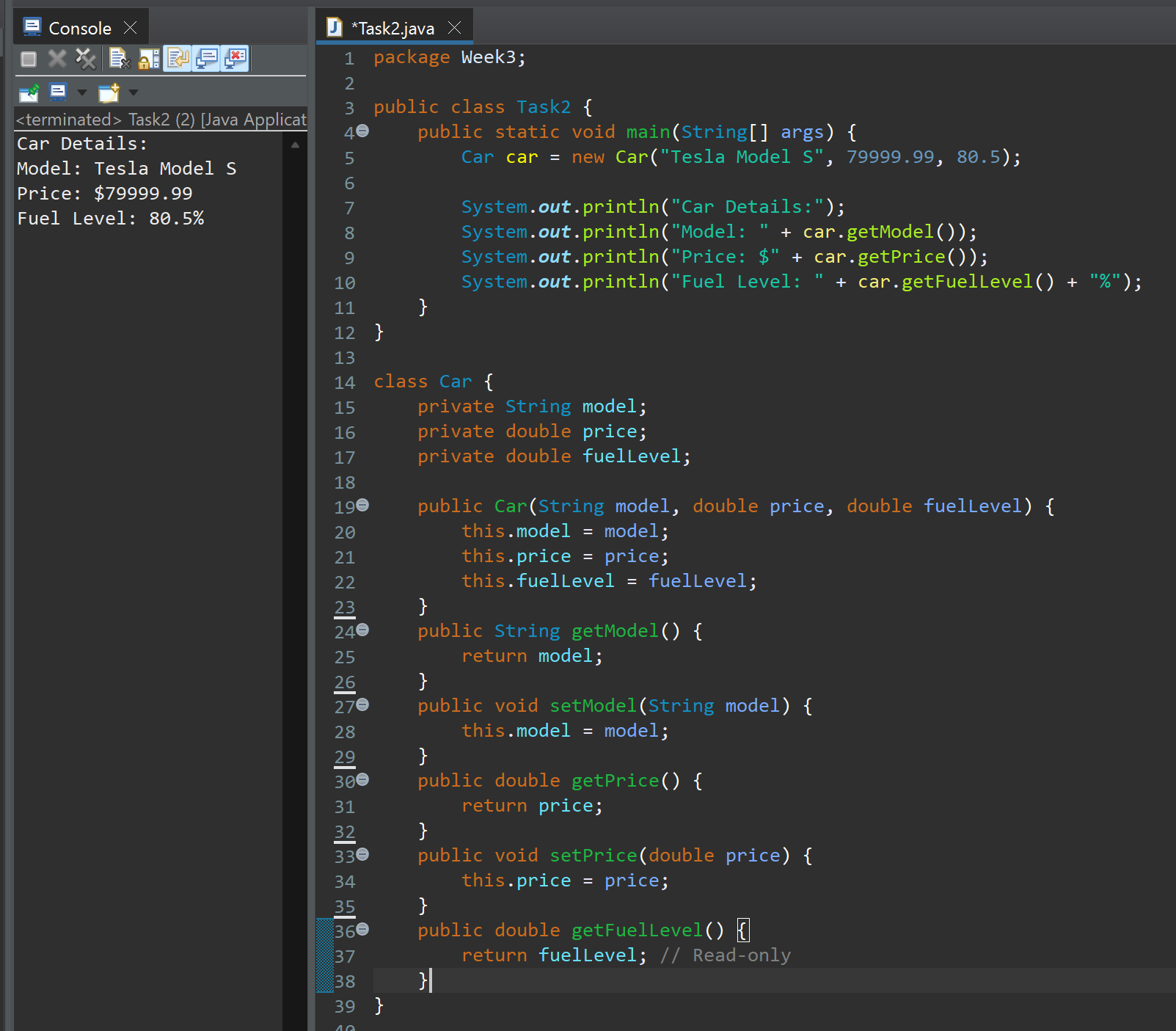
**Encapsulation**

1. Create a class Student with private fields id, name, age, and grade(A, B, C, D, E, F). Provide getter and setter methods to access and modify these fields.

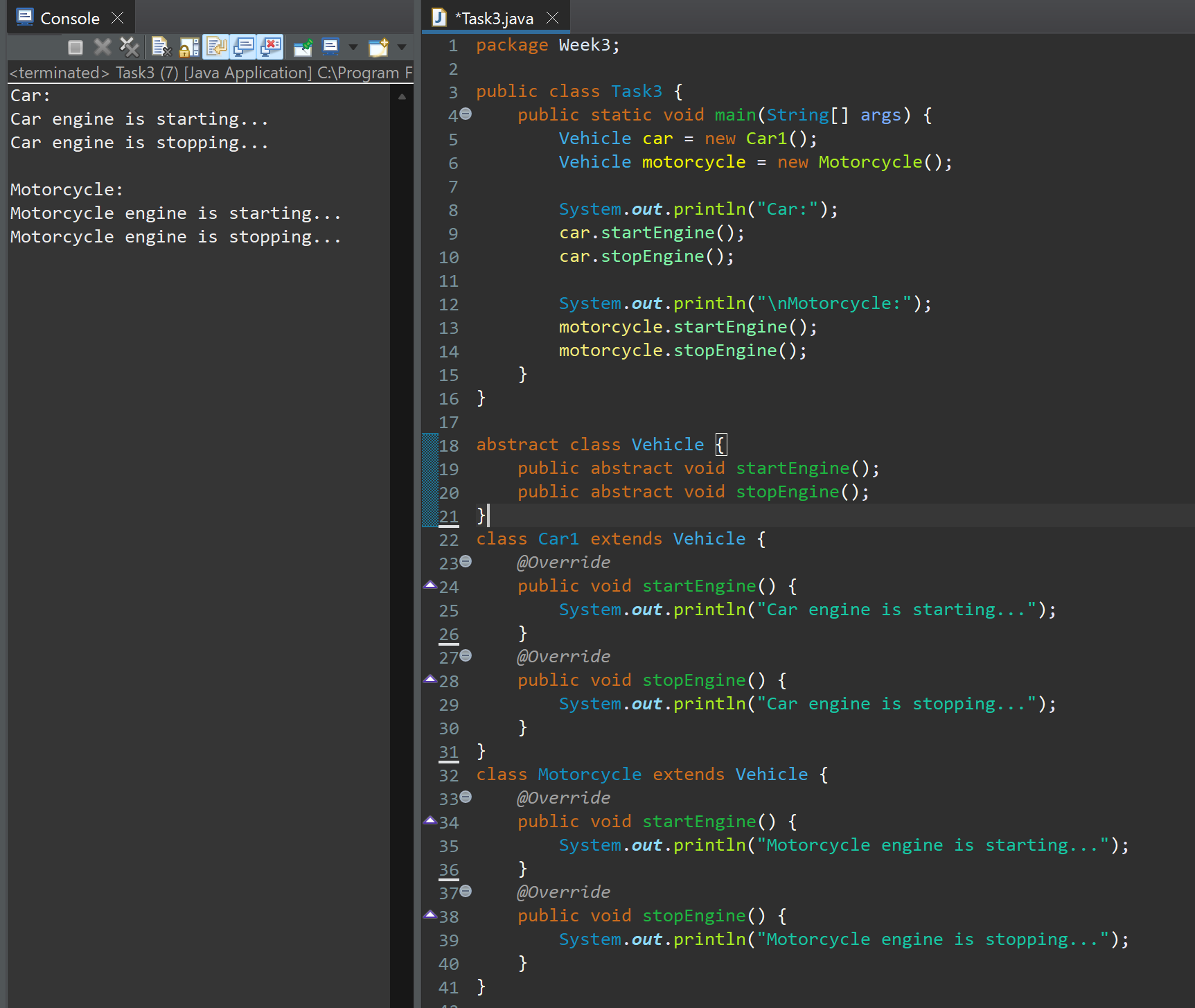


1. Create a class Car with private fields model, price and fuelLevel. Provide getter and setter methods for model and price, but ensure that the fuelLevel field is read-only.

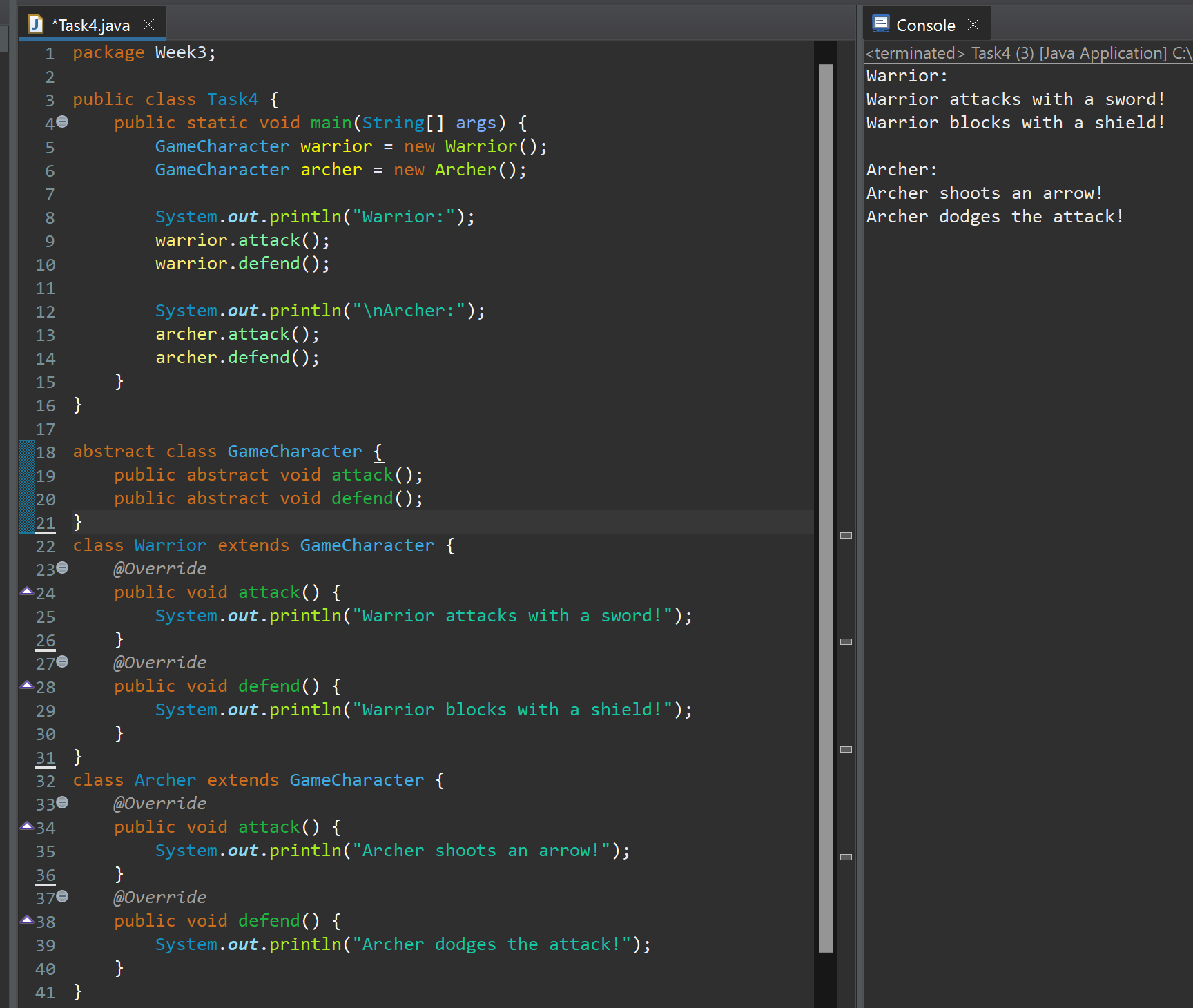


**Abstraction**

1. Create an abstract class Vehicle with abstract methods startEngine() and stopEngine(). Then create two classes Car and Motorcycle that extend Vehicle and implement these methods differently.

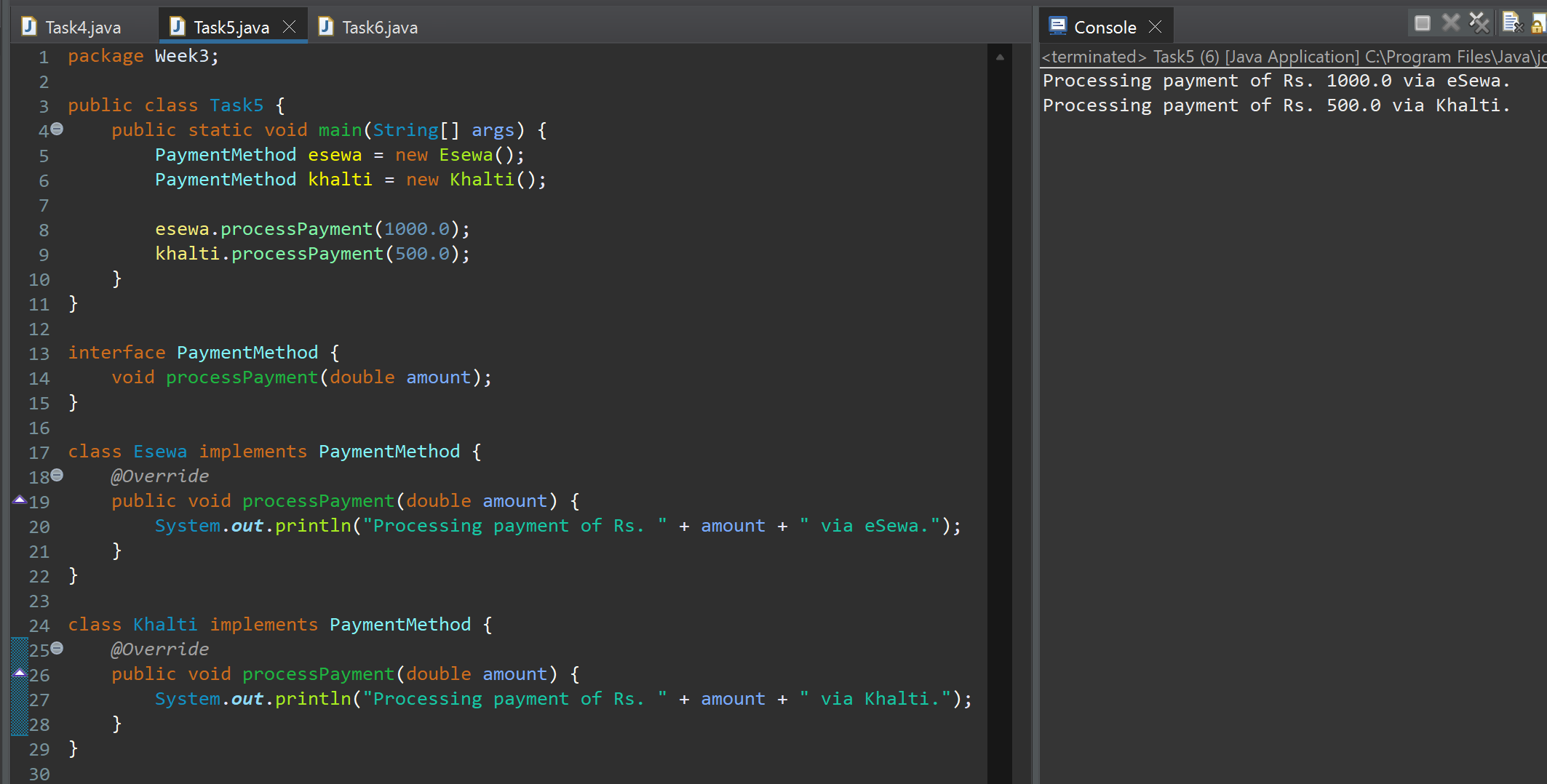


1. Create an abstract class GameCharacter with abstract methods like attack() and defend(). Then, create subclasses Warrior and Archer with different attack and defense behaviors.

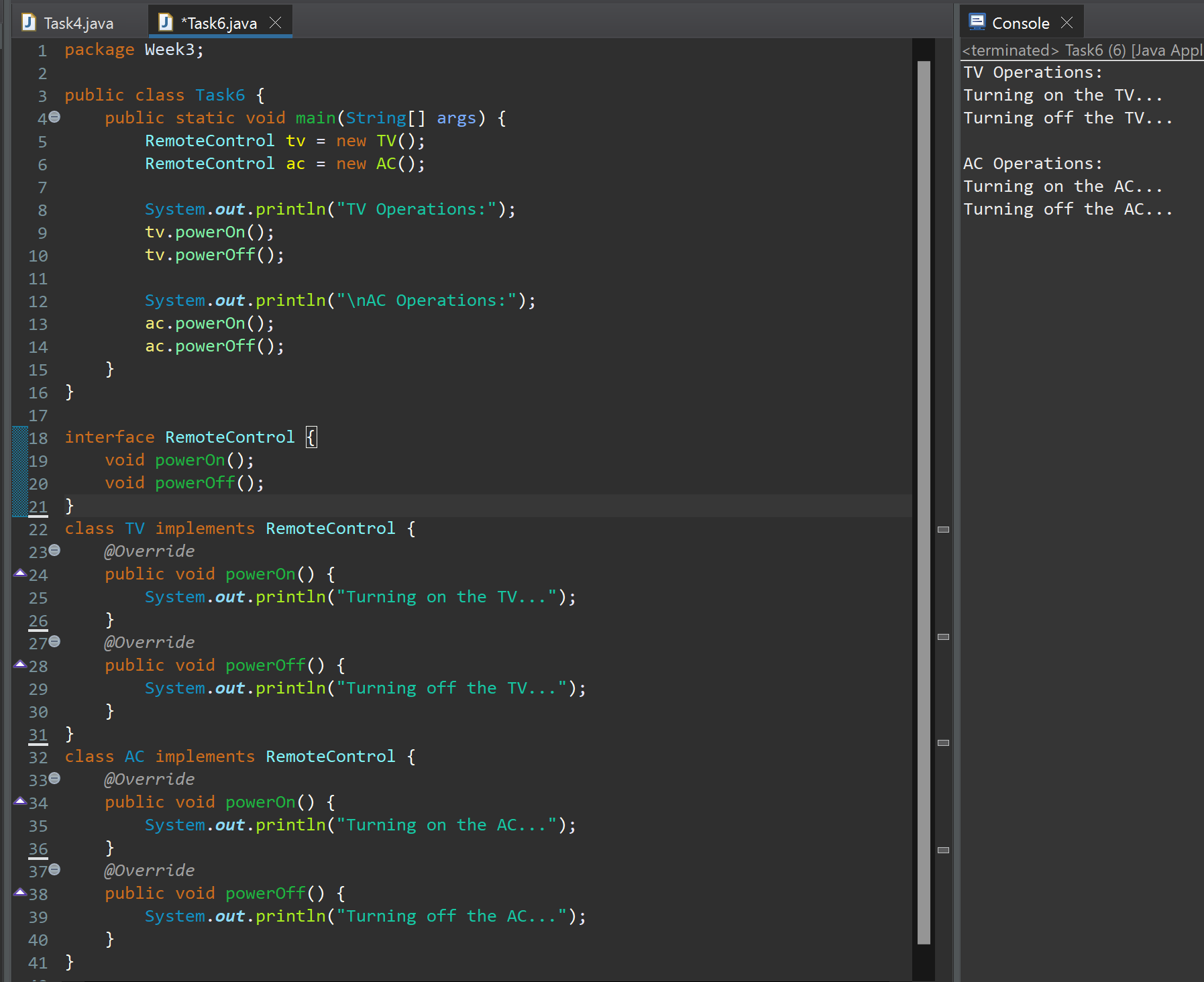


**Interface**

1. Create an interface PaymentMethod with a method processPayment(double amount). Implement it in classes Esewa and Khalti.

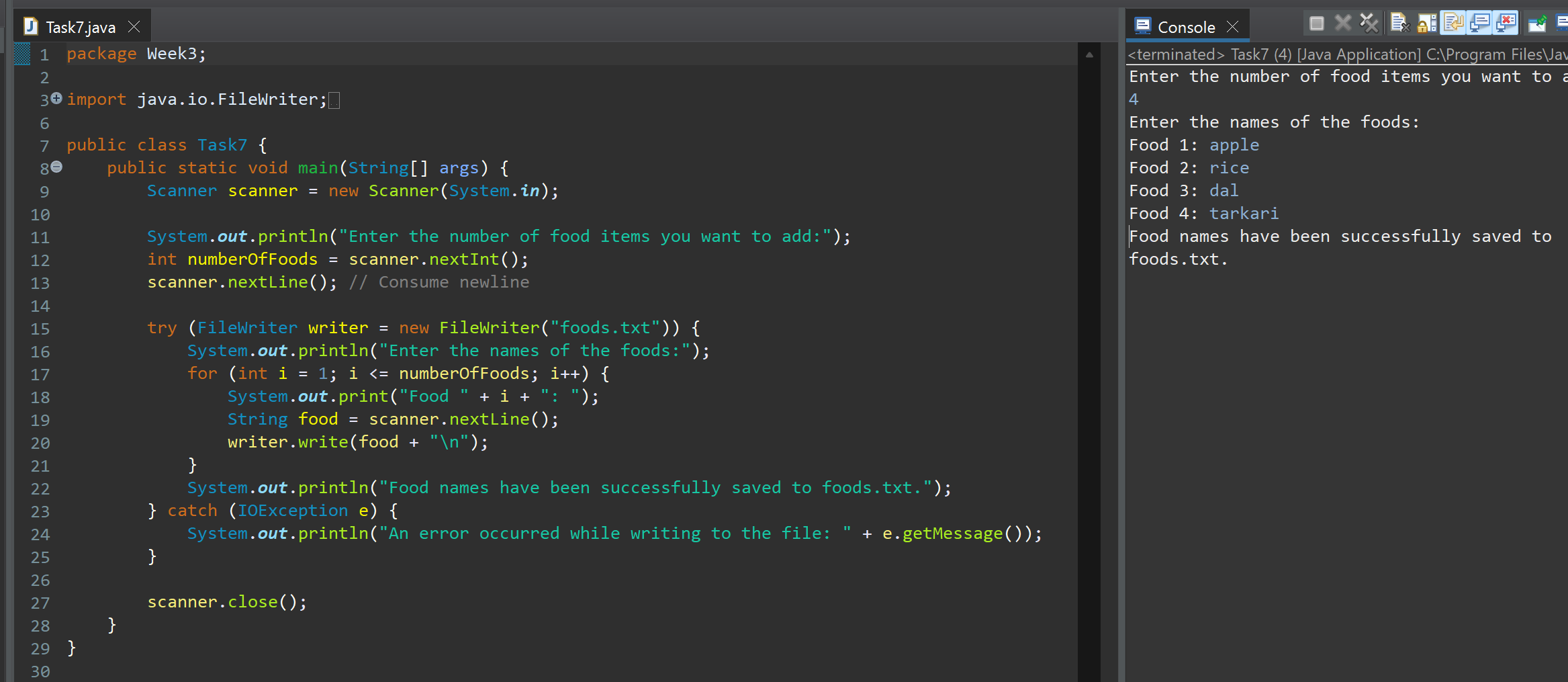


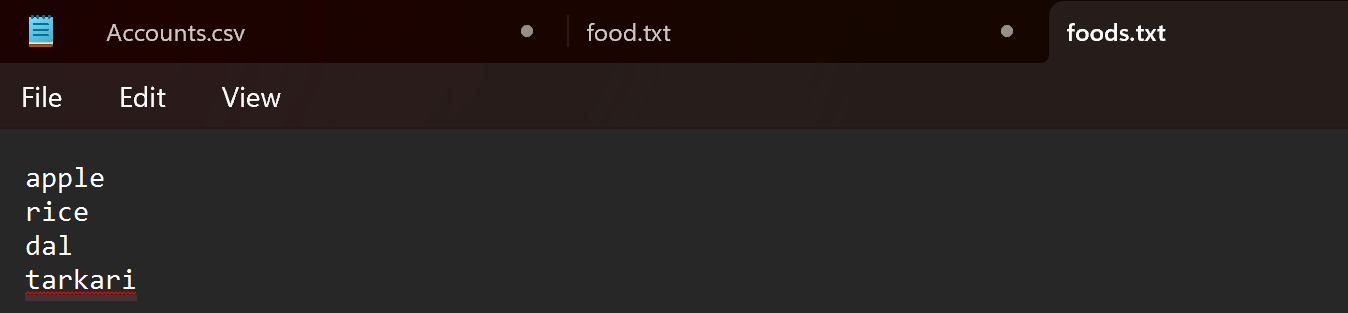
1. Create an interface RemoteControl with methods powerOn() and powerOff(). Implement this interface in classes TV and AC, which turn on and off their respective devices.



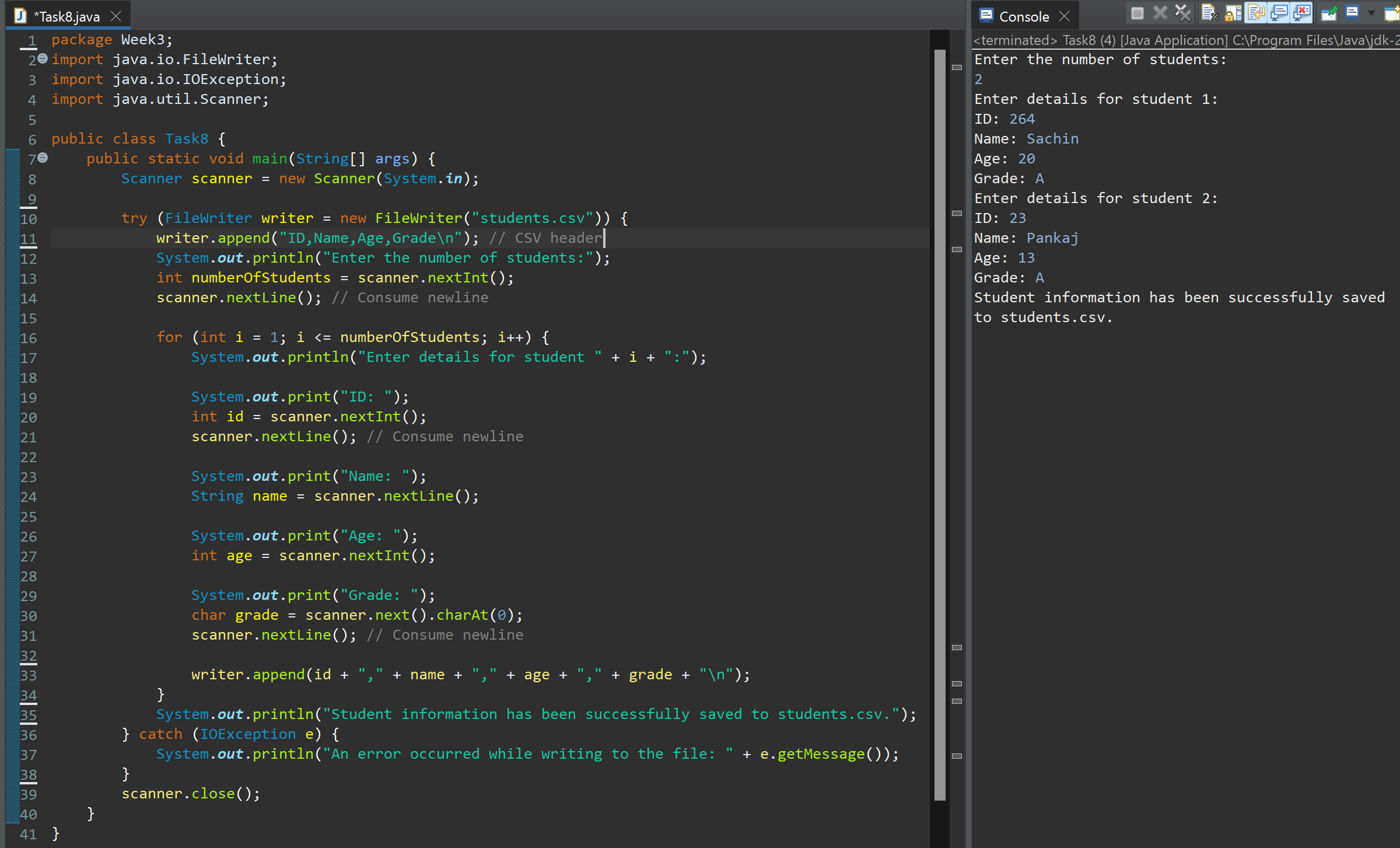
**File Handling**

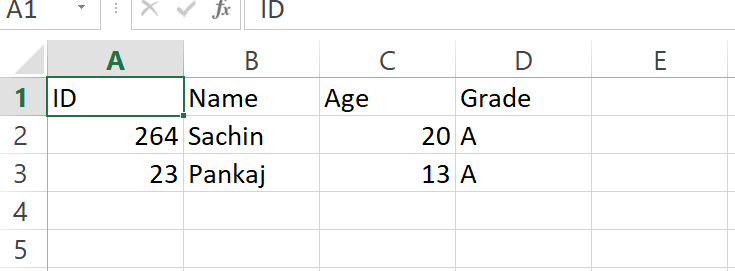
1. Write a program to take the name of foods as inputs from the user and store them in a .txt file.





1. Create a class Student with private fields name, age, grade(A, B, C, D, E, F). Then, write a program that stores student information(id, name, age, grade) into a .csv file.





1. Write a program that reads a list of students data from a csv file and stores them in a list. Then display the list of students according to their grade.

