Assignments

Hands-on Object-Oriented programming with Java 11

1. Inheritance and Polymorphism: -

Create a base class "Vehicle" with attributes like "make," "model," and a method "startEngine()." Derive two classes, "Car" and "Motorcycle," from the base class. Implement the "startEngine()" method differently in each derived class. Create instances of both classes and demonstrate polymorphism by invoking the "startEngine()" method on each.

2. Abstraction and Interfaces: -

Define an interface named "Shape" with methods like "calculateArea()" and "displayShapeInfo()." Implement this interface in classes like "Circle" and "Rectangle." Use abstraction to calculate the area differently for each shape. Demonstrate the usage of the interface by creating instances of both shapes and invoking their methods.

3. Abstract Classes and Polymorphism: -

Design an abstract class "Animal" with attributes like "name" and methods like "makeSound()." Create two concrete classes, "Dog" and "Cat," that extend the "Animal" class. Implement the "makeSound()" method differently for each class. Demonstrate polymorphism by creating an array of "Animal" objects and calling their "makeSound()" methods in a loop.

4. Encapsulation and Interfaces: -

Create an interface "Logger" with methods like "logInfo()" and "logError()." Implement this interface in a class named "FileLogger" and another class named "DatabaseLogger." Now, create a class "Application" that encapsulates the logging functionality. It should have a method like "performApplicationTask()" that internally uses the "Logger" interface to log information and errors. Demonstrate how different loggers can be plugged into the application without changing its core logic.

5 Exception Handling –

Design a class named "DataProcessor" that has a method "processData()" which takes an array of integers as input. Within this method, implement logic to calculate the average of the integers in the array. However, introduce a custom exception called "InvalidDataException" to handle the scenario when the array is empty or null.

Solution: https://github.com/TharunPatel20/UST-techAcademy/tree/2670b2b13d91696bb37ede78bc20b649ff7feeb3/USTJavaCourse/HandsOnObjectOrientedProgrammingWithJava11/src