PROJECT OBJECTIVE:-

TO BUILD A TAX CALCULATION APPLICATION:-

1. *MainApplication*: This is a public class serves as the entry point for the application. It initializes storage management and tax calculation components and provides a menu-driven interface for users to interact with property and vehicle tax functionalities.

2. *UserInterface*: An interface defining methods for displaying welcome messages, main menu options, error messages, and getting user input.

3. *StorageManager*: Manages storage of property and vehicle data using lists. It provides methods to add properties and vehicles, as well as retrieve lists of stored properties and vehicles.

4. *TaxCalculator*: Calculates total property and vehicle taxes based on provided lists of properties and vehicles. It iterates through the lists and calculates taxes accordingly.

5. *Property*: Represents a property with attributes such as base value, built-up area, age of construction, and location (in-city or not). It calculates property tax based on these attributes.

6. *Vehicle*: Represents a vehicle with attributes like registration number, brand, purchase cost, maximum velocity, passenger count, and vehicle type. It calculates vehicle tax based on these attributes.

The application provides functionalities to add property and vehicle details, calculate property and vehicle taxes, display all properties and vehicles, and view the total tax details. It includes error handling for invalid inputs and exceptions. Additionally, it displays a welcome screen upon startup and utilizes a menu-driven interface for user interaction.

flow of the program:-

1. The program starts with the MainApplication class, which initializes necessary components like StorageManager, TaxCalculator, and Scanner for user input.

2. The displayWelcomeScreen method displays a welcome message and prompts the user to log in (although no actual login functionality is implemented).

3. The run method runs the main loop of the program, displaying the main menu and handling user choices until the user chooses to exit.

4. Depending on the user's choice, the program executes methods to handle property tax, vehicle tax, display tax details, or exit.

5. For property tax and vehicle tax, the program provides options to add details, calculate tax, display all entries, or go back to the main menu.

6. The StorageManager class manages lists of properties and vehicles.

7. The TaxCalculator class calculates total property tax and total vehicle tax.

8. The Property and Vehicle classes represent property and vehicle objects respectively, each with methods to calculate tax based on their attributes.

9. The UserInterface interface defines methods for displaying the welcome screen, main menu, and error messages.

10. Overall, the program provides a simple interface for users to manage property and vehicle tax calculations.

Conclusion:-

This project is a tax calculation application designed to calculate property and vehicle taxes. It provides users with an intuitive interface to add property and vehicle details, calculate tax amounts based on specific criteria, and view summaries of all properties and vehicles along with their respective tax amounts. The application helps users manage their tax obligations effectively and efficiently.