

# **Tax Calculation Application**

## **Table of Contents**

1. Introduction
2. System Overview
3. Class Structure
4. Class Descriptions
  - 4.1. Property Class
  - 4.2. Vehicle Class
  - 4.3. TaxEntry Class
  - 4.4. PropertyOperations Class
  - 4.5. VehicleOperations Class
  - 4.6. TaxMain Class
5. Program Flow
6. How to Use
7. Sample Usage
8. Conclusion

## **1. Introduction**

This document provides documentation for the Tax Calculation Application, a Java program that calculates property and vehicle taxes based on user input. The program is designed to handle property and vehicle details, calculate taxes, and provide a summary of tax-related information.

## **2. System Overview**

The Tax Calculation Application consists of several classes that manage property and vehicle operations. Users can log in, add property and vehicle details, calculate tax amounts, and view tax summaries. The application is organized into classes to ensure modularity and ease of maintenance.

## **3. Class Structure**

The program is structured into the following classes:

### **3.1. Property Class**

- Manages property details such as base value of land, city status, age of property, property tax, ID, built-up area, and tax.
- Calculates property tax based on input parameters.

### **3.2. Vehicle Class**

- Manages vehicle details such as registration number, brand, maximum velocity, number of seats, vehicle type, cost, and tax.
- Calculates vehicle tax based on the type of vehicle.

### **3.3. TaxEntry Class**

- Represents tax summaries with fields for particular, quantity, and tax amount.

### **3.4. PropertyOperations Class**

- Handles property-related operations including adding property details, calculating property tax, viewing property details, and generating property tax summaries.

### **3.5. VehicleOperations Class**

- Handles vehicle-related operations including adding vehicle details, calculating vehicle tax, viewing vehicle details, and generating vehicle tax summaries.

### **3.6. TaxMain Class**

- Manages the main program flow.  
- Calculates and displays the total tax based on property and vehicle taxes.

## **4. Class Descriptions**

In this section, each class is described in detail, including its properties and methods.

### **4.1. Property Class**

Properties:

- baseValueOfLand  
- isInCity  
- ageOfProperty  
- propertyTax  
- id  
- builduparea  
- tax

**Methods:**

- calculatePropertyTax()

**4.2. Vehicle Class**

## Properties:

- RegNum
- brand
- maxvelocity
- noOfSeats
- vehicleType
- cost
- tax

**Methods:**

- calculateVehicleTax()

**4.3. TaxEntry Class**

## Properties:

- particular
- quantity
- tax

#### **4.4. PropertyOperations Class**

##### **Methods:**

- addPropertyDetails1()
- viewPropertyDetails()
- calculatePropertyTax()
- propertytaxmenu()
- getPropertyTaxSummary()
- calculateTotalPropertyTax()

#### **4.5. VehicleOperations Class**

##### **Methods:**

- addVehicleDetails()
- DisplayVehicle()
- calculateVehicleTaxByRegNum1()
- vehicletaxmenu()
- getVehicleTaxSummary()
- calculateTotalVehicleTax()

#### **4.6. TaxMain Class**

##### **Methods:**

- calculateTotalTax()

## **5. Program Flow**

The Program starts by initializing lists to store property and vehicle objects. Instances of the PropertyOperations, vehicleOperations, and TaxMain classes are created.

- **Login**

The Program prompts the user for a username and password to login.

If the login is successful the user proceeds to the main menu. Otherwise the program displays a login failed message and exits.

- **Main Menu**

The user is presented with a main menu that offers several options:

- Property Tax Operations
- Vehicle Tax Operations
- Calculate Total Tax
- Exit

- **Property Tax Operations**

If the user selects this option they can add properties, calculate property taxes for a specific property and they can view the what are properties are present and return to main menu

- **Vehicle Tax Operations**

If the user selects this option they can add, calculate and view the vehicle details and return back to main menu

- **Calculate Total Tax**

If the user selects this option the user can retrieve property and vehicle tax summaries and calculates the grand total tax by summing up property and vehicle taxes and display the summaries.

- **Exit**

If the user chooses this the program displays a thank you message and terminates.

## 6. Sample Usage

Examples of how to use the program with screenshots output.

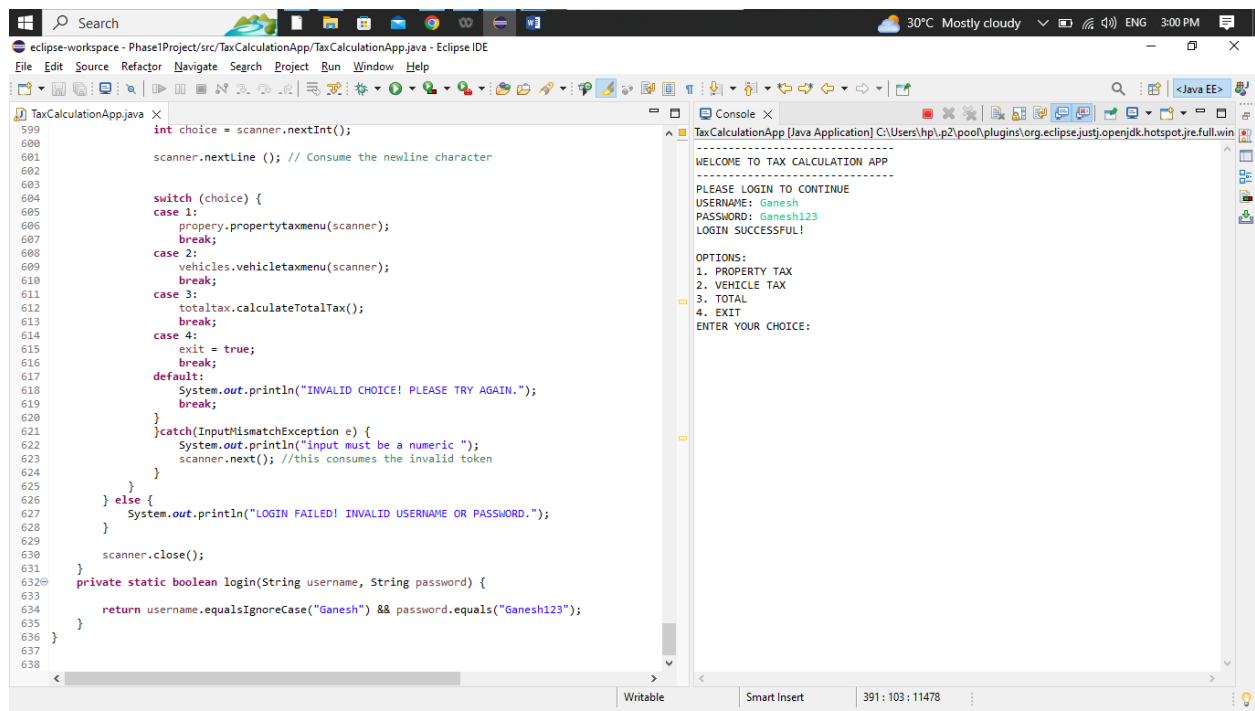
### Step 1

- Login

The Program prompts the user for a username and password to login.

If the login is successful the user proceeds to the main menu. Otherwise the program displays a login failed message and exits.

Enter the Correct username and password to Login.



The screenshot displays the Eclipse IDE interface. The main editor window shows the source code for `TaxCalculationApp.java`. The code includes a `switch` statement for handling user choices (1: Property Tax, 2: Vehicle Tax, 3: Total Tax, 4: Exit) and a `login` method that checks the username and password. The console window on the right shows the program's output, which includes a welcome message, a login prompt, and the successful login of the user 'Ganesh' with password 'Ganesh123'. The console also displays the main menu options.

```
599 int choice = scanner.nextInt();
600 scanner.nextLine (); // Consume the newline character
601
602
603
604 switch (choice) {
605 case 1:
606     property.propertytaxmenu(scanner);
607     break;
608 case 2:
609     vehicles.vehicletaxmenu(scanner);
610     break;
611 case 3:
612     totaltax.calculateTotalTax();
613     break;
614 case 4:
615     exit = true;
616     break;
617 default:
618     System.out.println("INVALID CHOICE! PLEASE TRY AGAIN.");
619     break;
620 }
621 catch(InputMismatchException e) {
622     System.out.println("input must be a numeric ");
623     scanner.next(); //this consumes the invalid token
624 }
625 }
626 } else {
627     System.out.println("LOGIN FAILED! INVALID USERNAME OR PASSWORD.");
628 }
629 }
630 scanner.close();
631 }
632 private static boolean login(String username, String password) {
633
634     return username.equalsIgnoreCase("Ganesh") && password.equals("Ganesh123");
635 }
636 }
637
638
```

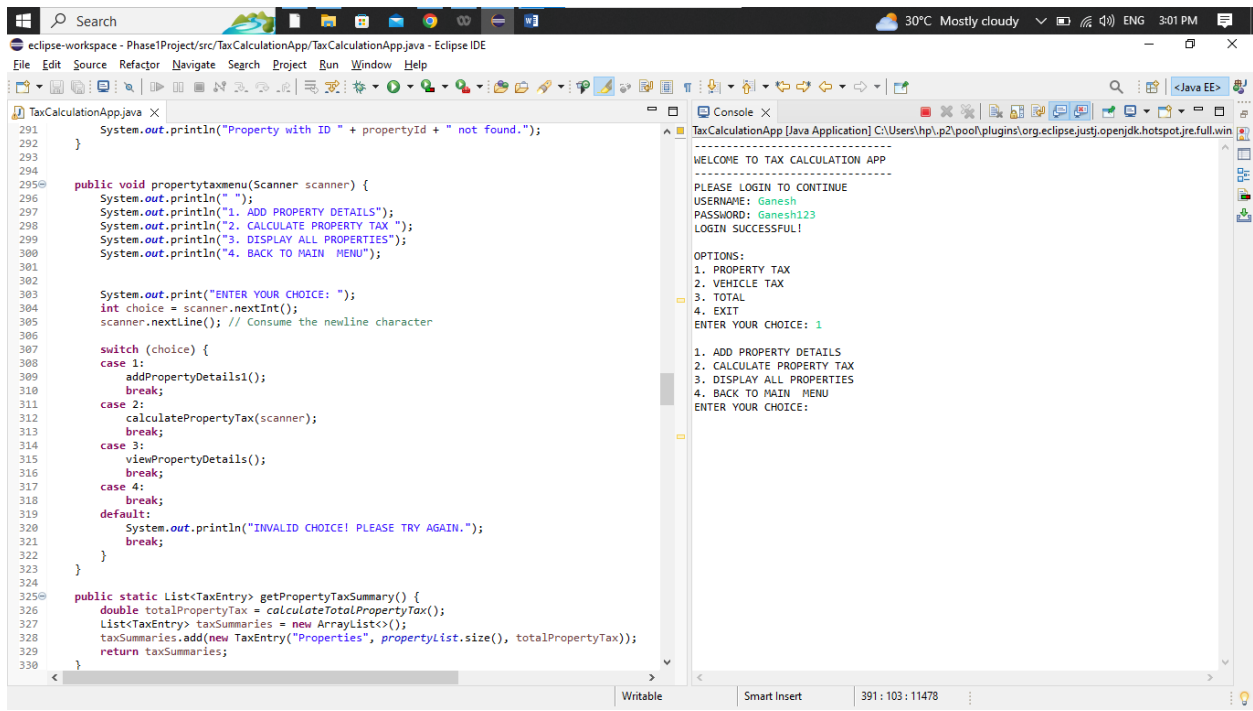
Console Output:

```
WELCOME TO TAX CALCULATION APP
=====
PLEASE LOGIN TO CONTINUE
USERNAME: Ganesh
PASSWORD: Ganesh123
LOGIN SUCCESSFUL!

OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE:
```

## Step 2:

To perform Property Tax Operations select Property Tax i.e. Enter 1.



The screenshot shows the Eclipse IDE with the `TaxCalculationApp.java` file open. The code is a Java application for property tax calculation. The console output shows the following sequence of events:

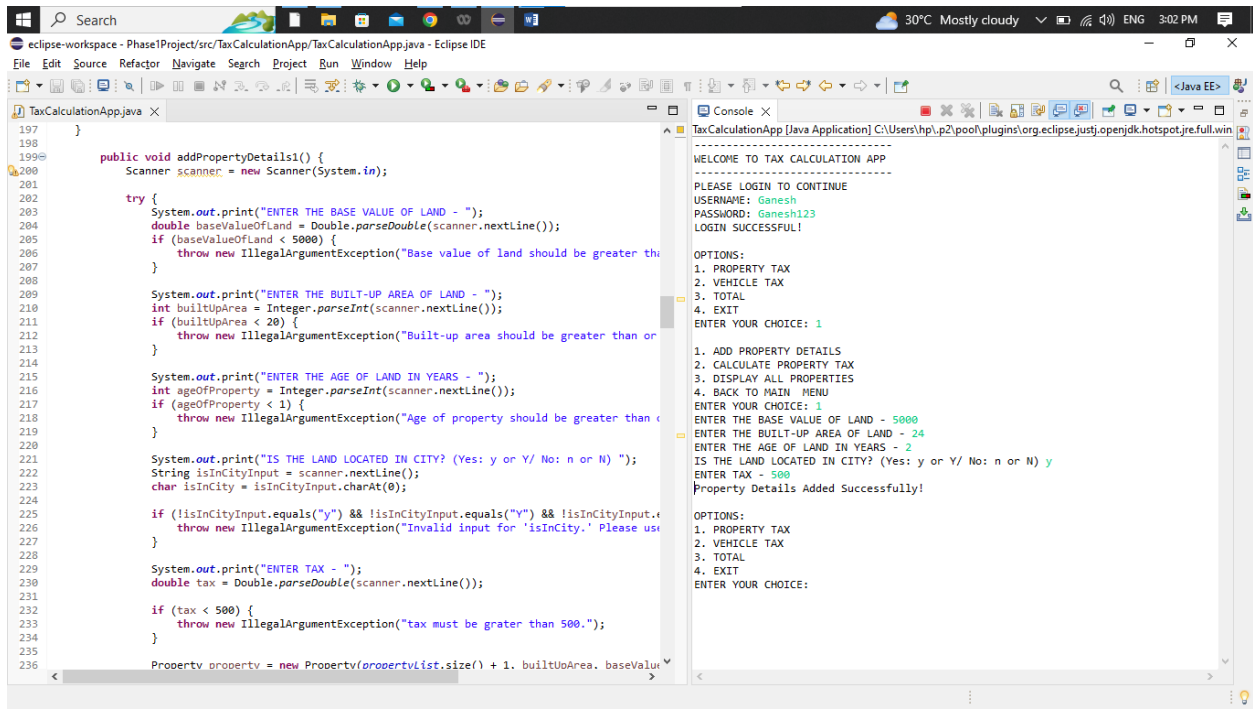
```
WELCOME TO TAX CALCULATION APP
PLEASE LOGIN TO CONTINUE
USERNAME: Ganesh
PASSWORD: Ganesh123
LOGIN SUCCESSFUL!

OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 1

1. ADD PROPERTY DETAILS
2. CALCULATE PROPERTY TAX
3. DISPLAY ALL PROPERTIES
4. BACK TO MAIN MENU
ENTER YOUR CHOICE:
```

## Step 3:

After choosing Property Tax, Enter 1 to add property details



The screenshot shows the Eclipse IDE with the `TaxCalculationApp.java` file open. The code is a Java application for property tax calculation. The console output shows the following sequence of events:

```
WELCOME TO TAX CALCULATION APP
PLEASE LOGIN TO CONTINUE
USERNAME: Ganesh
PASSWORD: Ganesh123
LOGIN SUCCESSFUL!

OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 1

1. ADD PROPERTY DETAILS
2. CALCULATE PROPERTY TAX
3. DISPLAY ALL PROPERTIES
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 1

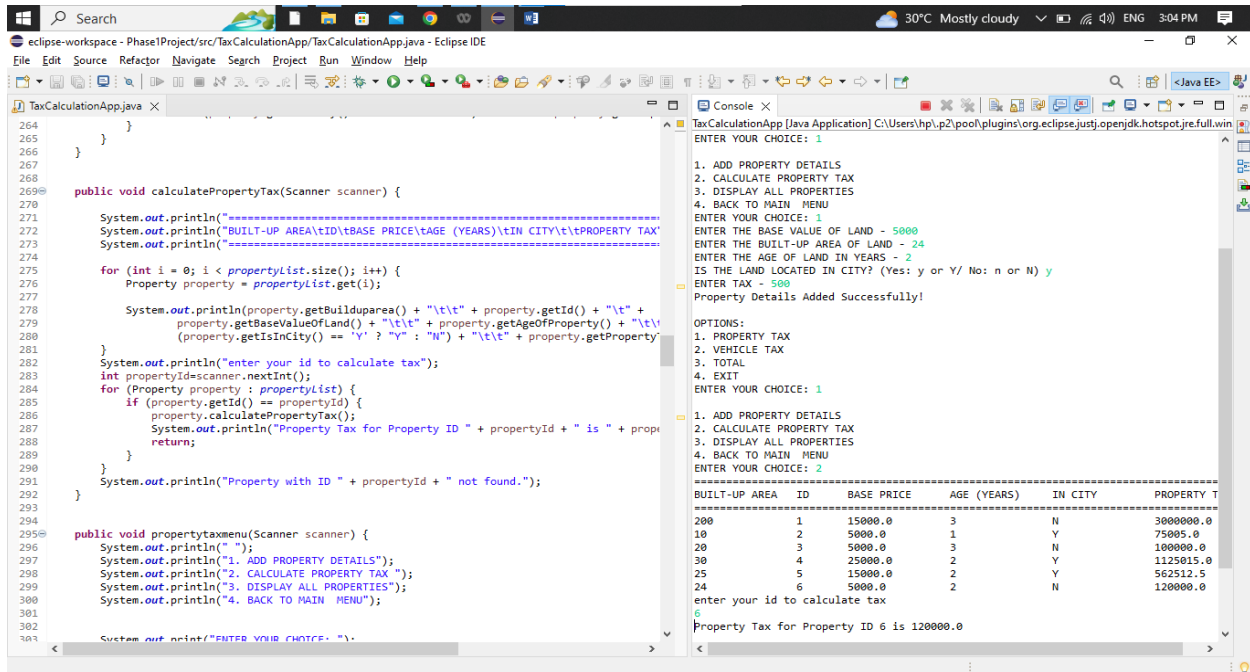
ENTER THE BASE VALUE OF LAND - 5000
ENTER THE BUILT-UP AREA OF LAND - 24
ENTER THE AGE OF LAND IN YEARS - 2
IS THE LAND LOCATED IN CITY? (Yes: y or Y/ No: n or N) y
ENTER TAX - 500
Property Details Added Successfully!

OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE:
```



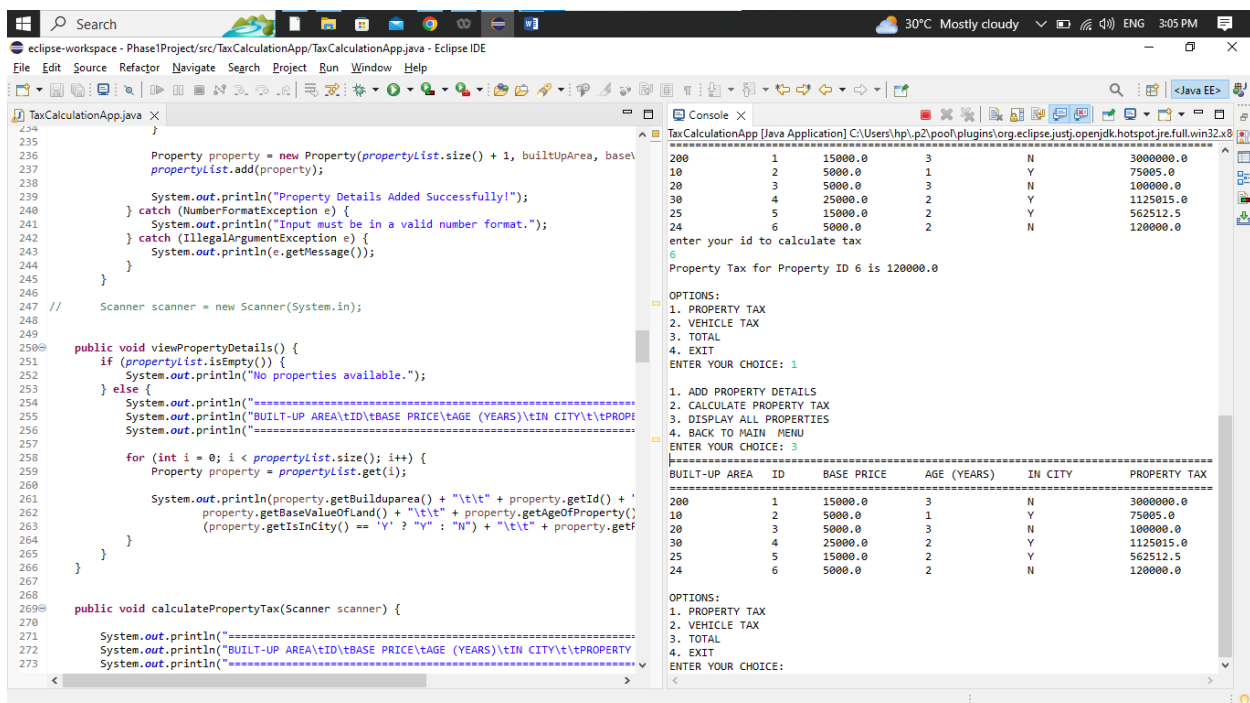
### Step 4:

After Adding the property details to calculate the added property details Enter 2 and select the property ID to perform the tax calculation in Property Tax Menu



### Step 5:

To display all the Property Details Enter 3 in Property Tax Menu.



## Step 6:

To perform Vehicle Operations, select the Vehicle Tax i.e. Enter 2.in Main Menu

```
484  
485     System.out.println("Vehicle with Registration Number " + regNumber + " not found.");  
486 }  
487  
488  
489  
490 public void vehicleTaxmenu(Scanner scanner) {  
491     System.out.println("");  
492     System.out.println("1. ADD VEHICLE DETAILS");  
493     System.out.println("2. CALCULATE VEHICLE TAX");  
494     System.out.println("3. DISPLAY ALL VEHICLE");  
495     System.out.println("4. BACK TO MAIN MENU");  
496  
497     System.out.print("ENTER YOUR CHOICE: ");  
498     int choice = scanner.nextInt();  
499     scanner.nextLine(); // Consume the newline character  
500  
501     switch (choice) {  
502     case 1:  
503         addVehicleDetails();  
504         break;  
505     case 2:  
506         calculateVehicleTaxByRegNum1();  
507         break;  
508     case 3:  
509         DisplayVehicle();  
510         break;  
511     case 4:  
512         break;  
513     default:  
514         System.out.println("INVALID CHOICE! PLEASE TRY AGAIN.");  
515         break;  
516     }  
517 }  
518  
519  
520  
521 public static List<TaxEntry> getVehicleTaxSummary() {  
522     double totalVehicleTax = calculateTotalVehicleTax();  
523     List<TaxEntry> taxSummary = new ArrayList<>();  
524     for (TaxEntry taxEntry : taxSummary) {  
525         totalVehicleTax += taxEntry.getTax();  
526     }  
527     return taxSummary;  
528 }  
529  
530 }
```

Console Output:

```
TaxCalculationApp [Java Application] C:\Users\hp\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe  
enter your id to calculate tax  
Property Tax for Property ID 6 is 120000.0  
OPTIONS:  
1. PROPERTY TAX  
2. VEHICLE TAX  
3. TOTAL  
4. EXIT  
ENTER YOUR CHOICE: 1  
1. ADD PROPERTY DETAILS  
2. CALCULATE PROPERTY TAX  
3. DISPLAY ALL PROPERTIES  
4. BACK TO MAIN MENU  
ENTER YOUR CHOICE: 3  
BUILT-UP AREA ID BASE PRICE AGE (YEARS) IN CITY PROPERTY TAX  
-----  
200 1 15000.0 3 N 3000000.0  
10 2 5000.0 1 Y 75005.0  
20 3 5000.0 3 N 100000.0  
30 4 25000.0 2 Y 1125015.0  
25 5 15000.0 2 Y 562512.5  
24 6 5000.0 2 N 120000.0  
OPTIONS:  
1. PROPERTY TAX  
2. VEHICLE TAX  
3. TOTAL  
4. EXIT  
ENTER YOUR CHOICE: 2  
1. ADD VEHICLE DETAILS  
2. CALCULATE VEHICLE TAX  
3. DISPLAY ALL VEHICLE  
4. BACK TO MAIN MENU  
ENTER YOUR CHOICE:
```

## Step 7:

After Selecting the Vehicle Tax enter 1 to add Vehicle Details.

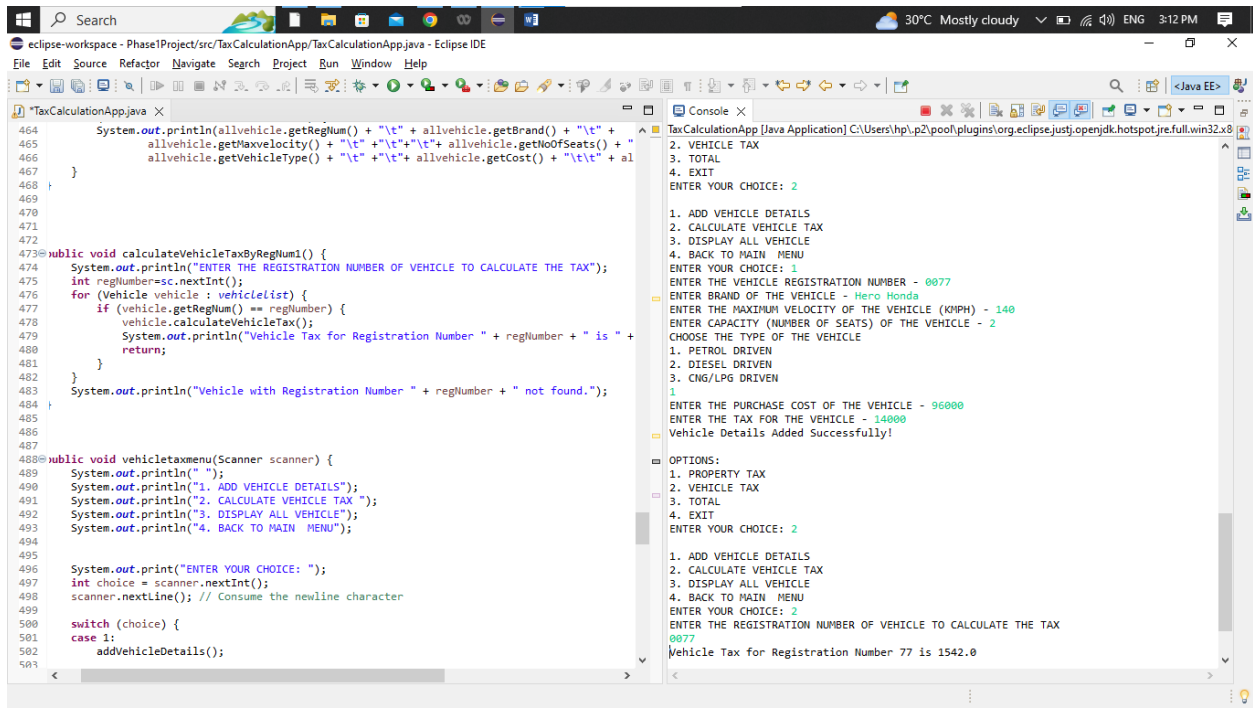
```
377  
378 }  
379  
380 System.out.print("ENTER BRAND OF THE VEHICLE - ");  
381 String brand = scanner.nextLine();  
382 if (brand == "") {  
383     throw new IllegalArgumentException("Brand name should not be empty.");  
384 }  
385  
386 System.out.print("ENTER THE MAXIMUM VELOCITY OF THE VEHICLE (KMPH) - ");  
387 int maxVelocity = Integer.parseInt(scanner.nextLine());  
388 if (maxVelocity < 120) {  
389     throw new IllegalArgumentException("maxVelocity of the vehicle should be more than 120");  
390 }else if (maxVelocity > 300) {  
391     throw new IllegalArgumentException("maxVelocity of the vehicle should be less than 300");  
392 }  
393  
394 System.out.print("ENTER CAPACITY (NUMBER OF SEATS) OF THE VEHICLE - ");  
395 int noOfSeats = Integer.parseInt(scanner.nextLine());  
396 if (noOfSeats < 2) {  
397     throw new IllegalArgumentException("noOfSeats of the vehicle should be more than 2");  
398 }else if (noOfSeats > 1000000) {  
399     throw new IllegalArgumentException("noOfSeats of the vehicle should be less than 1000000");  
400 }  
401  
402 try {  
403  
404     System.out.println("CHOOSE THE TYPE OF THE VEHICLE");  
405     System.out.println("1. PETROL DRIVEN");  
406     System.out.println("2. DIESEL DRIVEN");  
407     System.out.println("3. CNG/LPG DRIVEN");  
408  
409  
410 } catch (InputMismatchException e) {  
411     throw new IllegalArgumentException("Vehicle type must be numeric.");  
412 }  
413  
414 int vehicleType = Integer.parseInt(scanner.nextLine());  
415 String type;  
416 switch (vehicleType) {  
417     case 1:  
418         type = "PETROL DRIVEN";  
419     case 2:  
420         type = "DIESEL DRIVEN";  
421     case 3:  
422         type = "CNG/LPG DRIVEN";  
423     default:  
424         type = "INVALID CHOICE! PLEASE TRY AGAIN.";  
425 }  
426  
427 }
```

Console Output:

```
TaxCalculationApp [Java Application] C:\Users\hp\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe  
1. ADD PROPERTY DETAILS  
2. CALCULATE PROPERTY TAX  
3. DISPLAY ALL PROPERTIES  
4. BACK TO MAIN MENU  
ENTER YOUR CHOICE: 3  
BUILT-UP AREA ID BASE PRICE AGE (YEARS) IN CITY PROPERTY TAX  
-----  
200 1 15000.0 3 N 3000000.0  
10 2 5000.0 1 Y 75005.0  
20 3 5000.0 3 N 100000.0  
30 4 25000.0 2 Y 1125015.0  
25 5 15000.0 2 Y 562512.5  
24 6 5000.0 2 N 120000.0  
OPTIONS:  
1. PROPERTY TAX  
2. VEHICLE TAX  
3. TOTAL  
4. EXIT  
ENTER YOUR CHOICE: 2  
1. ADD VEHICLE DETAILS  
2. CALCULATE VEHICLE TAX  
3. DISPLAY ALL VEHICLE  
4. BACK TO MAIN MENU  
ENTER YOUR CHOICE: 1  
ENTER THE VEHICLE REGISTRATION NUMBER - 0077  
ENTER BRAND OF THE VEHICLE - Hero Honda  
ENTER THE MAXIMUM VELOCITY OF THE VEHICLE (KMPH) - 140  
ENTER CAPACITY (NUMBER OF SEATS) OF THE VEHICLE - 2  
CHOOSE THE TYPE OF THE VEHICLE  
1. PETROL DRIVEN  
2. DIESEL DRIVEN  
3. CNG/LPG DRIVEN  
1  
ENTER THE PURCHASE COST OF THE VEHICLE - 96000  
ENTER THE TAX FOR THE VEHICLE - 14000  
Vehicle Details Added Successfully!  
1
```

## Step 8 :

To Calculate the Vehicle Tax Details Enter 2. In Vehicle Tax Menu.



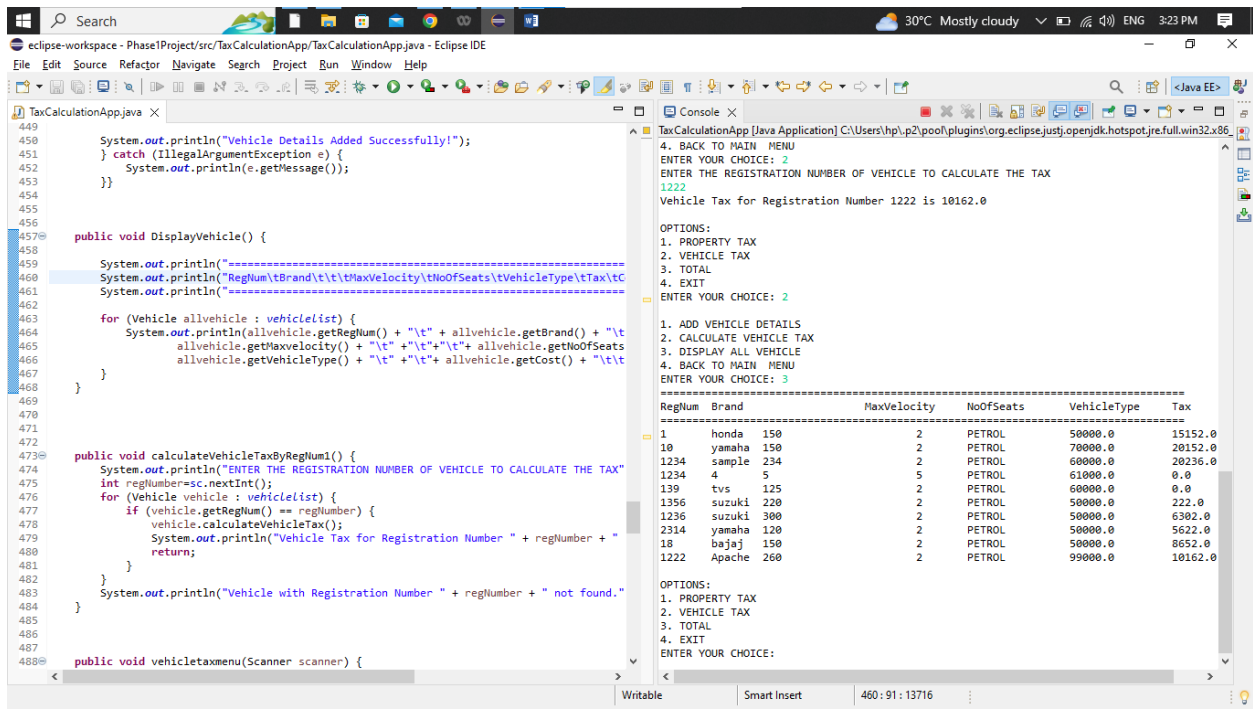
```
464      System.out.println(allvehicle.getRegNum() + "\t" + allvehicle.getBrand() + "\t" +
465      allvehicle.getMaxVelocity() + "\t" + "\t" + "\t" + allvehicle.getNoOfSeats() + "\t" +
466      allvehicle.getVehicleType() + "\t" + "\t" + allvehicle.getCost() + "\t\t" + al
467  }
468
469
470
471
472
473 public void calculateVehicleTaxByRegNum() {
474     System.out.println("ENTER THE REGISTRATION NUMBER OF VEHICLE TO CALCULATE THE TAX");
475     int regNumber = scanner.nextInt();
476     for (Vehicle vehicle : vehicleList) {
477         if (vehicle.getRegNum() == regNumber) {
478             vehicle.calculateVehicleTax();
479             System.out.println("Vehicle Tax for Registration Number " + regNumber + " is " +
480             return;
481         }
482     }
483     System.out.println("Vehicle with Registration Number " + regNumber + " not found.");
484
485
486
487
488 public void vehicleTaxMenu(Scanner scanner) {
489     System.out.println(" ");
490     System.out.println("1. ADD VEHICLE DETAILS");
491     System.out.println("2. CALCULATE VEHICLE TAX ");
492     System.out.println("3. DISPLAY ALL VEHICLE");
493     System.out.println("4. BACK TO MAIN MENU");
494
495     System.out.print("ENTER YOUR CHOICE: ");
496     int choice = scanner.nextInt();
497     scanner.nextLine(); // Consume the newline character
498
499     switch (choice) {
500     case 1:
501         addVehicleDetails();
502     }
```

Console Output:

```
TaxCalculationApp [Java Application] C:\Users\hp\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 2
1. ADD VEHICLE DETAILS
2. CALCULATE VEHICLE TAX
3. DISPLAY ALL VEHICLE
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 1
ENTER THE VEHICLE REGISTRATION NUMBER - 0077
ENTER BRAND OF THE VEHICLE - Hero Honda
ENTER THE MAXIMUM VELOCITY OF THE VEHICLE (KMPH) - 140
ENTER CAPACITY (NUMBER OF SEATS) OF THE VEHICLE - 2
CHOOSE THE TYPE OF THE VEHICLE
1. PETROL DRIVEN
2. DIESEL DRIVEN
3. CNG/LPG DRIVEN
1
ENTER THE PURCHASE COST OF THE VEHICLE - 96000
ENTER THE TAX FOR THE VEHICLE - 14000
Vehicle Details Added Successfully!
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 2
1. ADD VEHICLE DETAILS
2. CALCULATE VEHICLE TAX
3. DISPLAY ALL VEHICLE
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 2
ENTER THE REGISTRATION NUMBER OF VEHICLE TO CALCULATE THE TAX
0077
Vehicle Tax for Registration Number 77 is 1542.0
```

## Step 9:

To Display All the Vehicle Details Enter 3 in Vehicle Tax Menu.



```
449      System.out.println("Vehicle Details Added Successfully!");
450  } catch (IllegalArgumentException e) {
451      System.out.println(e.getMessage());
452  }
453  }
454
455
456
457 public void DisplayVehicle() {
458
459     System.out.println("=====");
460     System.out.println("RegNum\tBrand\t\t\t\tMaxVelocity\tNoOfSeats\tVehicleType\tTax\t\t");
461     System.out.println("=====");
462
463     for (Vehicle allvehicle : vehicleList) {
464         System.out.println(allvehicle.getRegNum() + "\t" + allvehicle.getBrand() + "\t\t\t\t" +
465         allvehicle.getMaxVelocity() + "\t" + "\t" + "\t" + allvehicle.getNoOfSeats() + "\t\t" +
466         allvehicle.getVehicleType() + "\t" + "\t" + allvehicle.getCost() + "\t\t" +
467     }
468
469
470
471
472
473 public void calculateVehicleTaxByRegNum() {
474     System.out.println("ENTER THE REGISTRATION NUMBER OF VEHICLE TO CALCULATE THE TAX");
475     int regNumber = scanner.nextInt();
476     for (Vehicle vehicle : vehicleList) {
477         if (vehicle.getRegNum() == regNumber) {
478             vehicle.calculateVehicleTax();
479             System.out.println("Vehicle Tax for Registration Number " + regNumber + "
480             return;
481         }
482     }
483     System.out.println("Vehicle with Registration Number " + regNumber + " not found.");
484
485
486
487
488 public void vehicleTaxMenu(Scanner scanner) {
```

Console Output:

```
TaxCalculationApp [Java Application] C:\Users\hp\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 2
ENTER THE REGISTRATION NUMBER OF VEHICLE TO CALCULATE THE TAX
1222
Vehicle Tax for Registration Number 1222 is 10162.0
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 2
1. ADD VEHICLE DETAILS
2. CALCULATE VEHICLE TAX
3. DISPLAY ALL VEHICLE
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 3
=====
RegNum Brand MaxVelocity NoOfSeats VehicleType Tax
=====
10 honda 150 2 PETROL 50000.0 15152.0
1234 yamaha 150 2 PETROL 70000.0 20152.0
1234 sample 234 2 PETROL 60000.0 20236.0
1234 4 5 PETROL 61000.0 0.0
139 tvs 125 2 PETROL 60000.0 0.0
1356 suzuki 220 2 PETROL 50000.0 222.0
1236 suzuki 300 2 PETROL 50000.0 6302.0
2314 yamaha 120 2 PETROL 50000.0 5622.0
18 bajaj 150 2 PETROL 50000.0 8652.0
1222 Apache 260 2 PETROL 99000.0 10162.0
=====
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE:
```

## Step 10:

To Calculate Both the Property Tax and the Vehicle Tax Enter 3 in Main Menu

```
527 double totalTax = 0.0;
528 for (Vehicle vehicle : vehicleList) {
529     vehicle.calculateVehicleTax();
530     totalTax += vehicle.getTax();
531 }
532 return totalTax;
533 }
534 }
535
536 class TaxMain {
537     public void calculateTotalTax() {
538         List<TaxEntry> propertyTaxSummary = PropertyOperations.getPropertyTaxSummary();
539         List<TaxEntry> vehicleTaxSummary = VehicleOperations.getVehicleTaxSummary();
540
541         double grandTotal = 0.0;
542         System.out.println("=====");
543         System.out.println("SR. NO. \t PARTICULAR \t QUANTITY \t TAX");
544         System.out.println("=====");
545
546         int srNo = 1;
547         for (TaxEntry summary : propertyTaxSummary) {
548             grandTotal += summary.getTax();
549             System.out.println(srNo++ + "\t" + summary.getParticular() + "\t\t" + summary.getTax());
550         }
551
552         for (TaxEntry summary : vehicleTaxSummary) {
553             grandTotal += summary.getTax();
554             System.out.println(srNo++ + "\t" + summary.getParticular() + "\t\t" + summary.getTax());
555         }
556         System.out.println("=====");
557         System.out.println("TOTAL \t\t\t\t\t" + grandTotal);
558         System.out.println("=====");
559     }
560 }
561
562 public class TaxCalculationApp {
563     public static void main(String[] args) {
564         PropertyOperations property = new PropertyOperations();
565         VehicleOperations vehicles = new VehicleOperations();
566     }
567 }
```

Console Output:

```
2. CALCULATE VEHICLE TAX
3. DISPLAY ALL VEHICLE
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 3
=====
RegNum Brand MaxVelocity NoOfSeats VehicleType Tax
=====
1 honda 150 2 PETROL 50000.0 15152.0
10 yamaha 150 2 PETROL 70000.0 20152.0
1234 sample 234 2 PETROL 60000.0 20236.0
1234 4 5 PETROL 61000.0 0.0
139 tvs 125 2 PETROL 60000.0 0.0
1356 suzuki 220 2 PETROL 50000.0 222.0
1236 suzuki 300 2 PETROL 50000.0 6302.0
2314 yamaha 120 2 PETROL 50000.0 5622.0
18 bajaj 150 2 PETROL 50000.0 8652.0
1222 Apache 260 2 PETROL 99000.0 10162.0
=====
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 3
=====
SR. NO. PARTICULAR QUANTITY TAX
=====
1 Properties 5 4862532.5
2 Vehicles 10 61737.0
=====
TOTAL 4924269.5
=====
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE:
```

## Step 11:

To Exit the Application Enter 4.

```
585 boolean exit = false;
586
587 while (!exit) {
588     System.out.println("\nOPTIONS:");
589     System.out.println("1. PROPERTY TAX");
590     System.out.println("2. VEHICLE TAX");
591     System.out.println("3. TOTAL");
592     System.out.println("4. EXIT");
593
594     System.out.print("ENTER YOUR CHOICE: ");
595     try {
596         int choice = scanner.nextInt();
597         scanner.nextLine(); // Consume the newline character
598
599         switch (choice) {
600             case 1:
601                 property.propertytaxmenu(scanner);
602                 break;
603             case 2:
604                 vehicles.vehicletaxmenu(scanner);
605                 break;
606             case 3:
607                 totaltax.calculateTotalTax();
608                 break;
609             case 4:
610                 exit = true;
611                 System.out.println("Thanks Visit Again.");
612                 break;
613             default:
614                 System.out.println("INVALID CHOICE! PLEASE TRY AGAIN.");
615                 break;
616         }
617     } catch (InputMismatchException e) {
618         System.out.println("input must be a numeric ");
619         scanner.next(); //this consumes the invalid token
620     }
621 }
622 }
```

Console Output:

```
4. BACK TO MAIN MENU
ENTER YOUR CHOICE: 3
=====
RegNum Brand MaxVelocity NoOfSeats VehicleType Tax
=====
1 honda 150 2 PETROL 50000.0 15152.0
10 yamaha 150 2 PETROL 70000.0 20152.0
1234 sample 234 2 PETROL 60000.0 20236.0
1234 4 5 PETROL 61000.0 0.0
139 tvs 125 2 PETROL 60000.0 0.0
1356 suzuki 220 2 PETROL 50000.0 222.0
1236 suzuki 300 2 PETROL 50000.0 6302.0
2314 yamaha 120 2 PETROL 50000.0 5622.0
18 bajaj 150 2 PETROL 50000.0 8652.0
=====
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 3
=====
SR. NO. PARTICULAR QUANTITY TAX
=====
1 Properties 5 4862532.5
2 Vehicles 9 51575.0
=====
TOTAL 4914107.5
=====
OPTIONS:
1. PROPERTY TAX
2. VEHICLE TAX
3. TOTAL
4. EXIT
ENTER YOUR CHOICE: 4
Thanks Visit Again.
```

## **7. Conclusion**

This Summarize the key features and functionality of the Tax Calculation Application.

This documentation provides an overview of the Tax Calculation Application, its class structure, and usage guidelines. For detailed information about each class and method, refer to the respective sections above.