Tax Calculation Application

Git Hub url:-https://github.com/AbdulrasheedShaik4127/Projects.git

Step 1:-Introduction

This Application deals with Calculation of Tax's for Property tax and Vehicle tax

Step 2:-Process

1.When we Run the application a welcome screen will display and asks the user to enter the username and password and then we process the username and password if it matches with the data then it will allow user to enter into application

2.After entering into Application again a list of options will be displayed such as

1.Property tax
2.vehicle tax
3.total
4.Exit

3.The user has to Selected the above four options only else an exception will be raised

4.If the user selects the option 1 then the again Property related tasks will be displayed such as

1.Add Property
2.Calculate Tax
3.Display Property
4.Back to main menu

5.Based on user choice respective details will be asked to user and then it will perform task which user has chosen

6.Same process will be continued for the remaining options like

1.Vehicle Tax

2.total

7.If the user chose option 4 i.e;Exit then the application will stoped by displaying a message “THANKYOU VISIT AGAIN”.

Step 3:-Methods and Classes

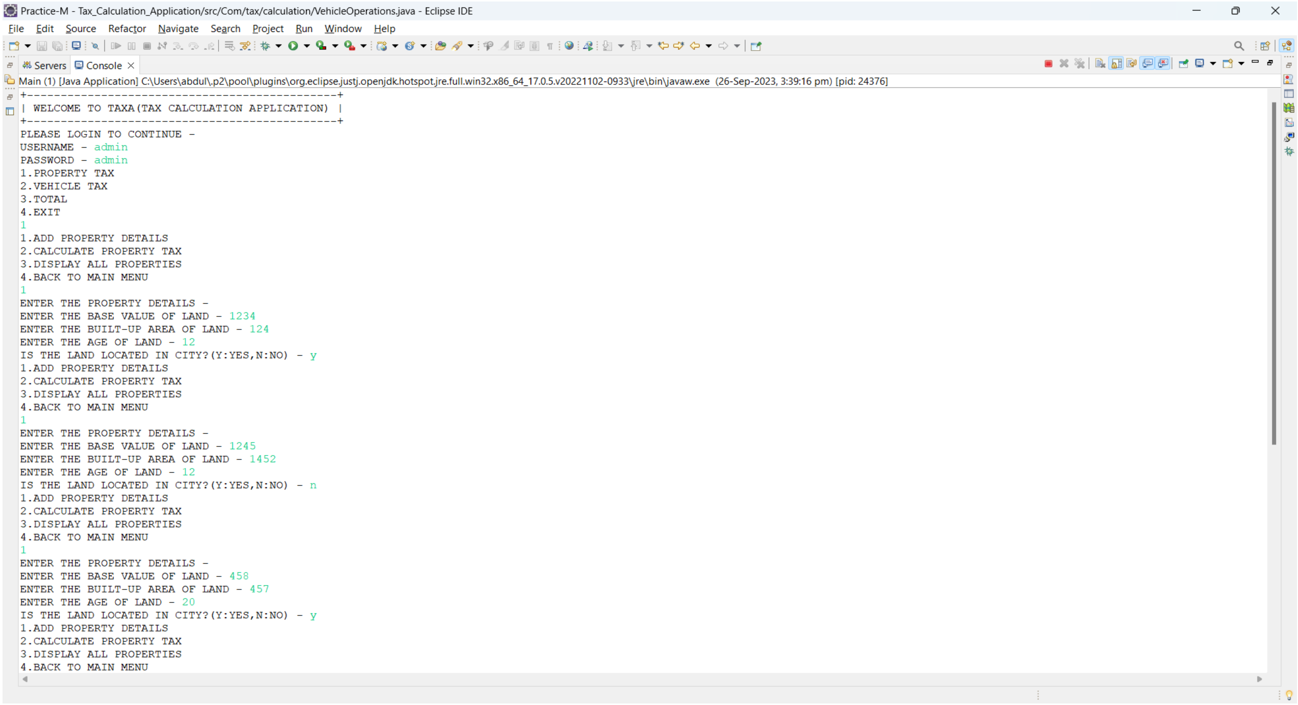
1.Classes:-

PropertClass:-contains attributes of Property
VehicleClass:-contains attributes of Vehicle
PropertyOperationsClass:-contains methods for PropertyClass
VehicleOperationsClass:-contains methods for VehicleClass

2.Methods:-

addProperty():-used to add details of property
setTax():-used to calculate Property Tax
displayDetails():-used to display details of property
addVehicleDetails():-used to add vehicle details
setTax():-used to calculate vehicle Tax
displayVehicle1():-used to display details of vehicle
totalTax():-used to find the total tax i.e;property tax+vehicle tax

Step 4:-Outputs


Practice-M - Tax_Calculation_Application/src/Com/tax/calculation/VehicleOperations.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Servers
Console

Main (1) [Java Application] C:\Users\abdul\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (26-Sep-2023, 3:39:16 pm) [pid: 24376]

```
+-----------------------------------------------+
| WELCOME TO TAXA(TAX CALCULATION APPLICATION) |
+-----------------------------------------------+
PLEASE LOGIN TO CONTINUE -
USERNAME - admin
PASSWORD - admin
1.PROPERTY TAX
2.VEHICLE TAX
3.TOTAL
4.EXIT
1
1.ADD PROPERTY DETAILS
2.CALCULATE PROPERTY TAX
3.DISPLAY ALL PROPERTIES
4.BACK TO MAIN MENU
1
ENTER THE PROPERTY DETAILS -
ENTER THE BASE VALUE OF LAND - 1234
ENTER THE BUILT-UP AREA OF LAND - 124
ENTER THE AGE OF LAND - 12
IS THE LAND LOCATED IN CITY?(Y:YES,N:NO) - y
1.ADD PROPERTY DETAILS
2.CALCULATE PROPERTY TAX
3.DISPLAY ALL PROPERTIES
4.BACK TO MAIN MENU
1
ENTER THE PROPERTY DETAILS -
ENTER THE BASE VALUE OF LAND - 1245
ENTER THE BUILT-UP AREA OF LAND - 1452
ENTER THE AGE OF LAND - 12
IS THE LAND LOCATED IN CITY?(Y:YES,N:NO) - n
1.ADD PROPERTY DETAILS
2.CALCULATE PROPERTY TAX
3.DISPLAY ALL PROPERTIES
4.BACK TO MAIN MENU
1
ENTER THE PROPERTY DETAILS -
ENTER THE BASE VALUE OF LAND - 458
ENTER THE BUILT-UP AREA OF LAND - 457
ENTER THE AGE OF LAND - 20
IS THE LAND LOCATED IN CITY?(Y:YES,N:NO) - y
1.ADD PROPERTY DETAILS
2.CALCULATE PROPERTY TAX
3.DISPLAY ALL PROPERTIES
4.BACK TO MAIN MENU
```

PropertyClass.java

```
package Com.tax.calculation;

public class PropertyClass {

	double value;

	int built_up_area;

	int age_of_land;

	String located;

	double tax;

	public double getTax() {
        return tax;

    }

    public void setTax(double tax) {

        this.tax = tax;

    }

    public PropertyClass(double value, int
built_up_area, int age_of_land, String located) {

        super();

        this.value = value;

        this.built_up_area = built_up_area;

        this.age_of_land = age_of_land;

        this.located = located;

    }

    public double getValue() {

        return value;

    }

    public void setValue(double value) {

        this.value = value;
```

```
}
public int getBuilt_up_area() {
    return built_up_area;
}
public void setBuilt_up_area(int built_up_area) {
    this.built_up_area = built_up_area;
}
public int getAge_of_land() {
    return age_of_land;
}
public void setAge_of_land(int age_of_land) {
    this.age_of_land = age_of_land;
}
public String getLocated() {
    return located;
}
public void setLocated(String located) {
        this.located = located;
    }
    public PropertyClass() {
        super();
    }
}
```

VehicleClass.java

```
package Com.tax.calculation;
public class VehicleClass {
    int registration_number;
    String brand;
    int max_velocity;
    int no_of_seats;
    String type_of_vehicle;
    double purchase_cost;
    double vehicle_tax;
    public int getRegistration_number() {
        return registration_number;

    }

    public void setRegistration_number(int
registration_number) {

        this.registration_number = registration_number;

    }

    public String getBrand() {

        return brand;

    }

    public void setBrand(String brand) {

        this.brand = brand;

    }

    public int getMax_velocity() {

        return max_velocity;

    }

    public void setMax_velocity(int max_velocity) {

        this.max_velocity = max_velocity;

    }
    public int getNo_of_seats() {
        return no_of_seats;
    }
    public void setNo_of_seats(int no_of_seats) {
        this.no_of_seats = no_of_seats;
    }
    public String getType_of_vehicle() {
        return type_of_vehicle;
    }
    public void setType_of_vehicle(String
type_of_vehicle) {
        this.type_of_vehicle = type_of_vehicle;
    }
    public double getPurchase_cost() {
        return purchase_cost;
    }
    public void setPurchase_cost(double purchase_cost)
{
        this.purchase_cost = purchase_cost;
    }
    public double getVehicle_tax() {
        return vehicle_tax;
    }
    public void setVehicle_tax(double vehicle_tax) {
        this.vehicle_tax = vehicle_tax;
    }
    public VehicleClass(int registration_number, String
brand, int max_velocity, int no_of_seats,
            String type_of_vehicle, double
purchase_cost, double vehicle_tax) {
        super();
        this.registration_number = registration_number;
        this.brand = brand;
        this.max_velocity = max_velocity;
        this.no_of_seats = no_of_seats;
        this.type_of_vehicle = type_of_vehicle;
        this.purchase_cost = purchase_cost;

        this.vehicle_tax = vehicle_tax;

    }

    public VehicleClass() {

        super();

    }

}
```

PropertyOperations.java

```
package Com.tax.calculation;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;



public class PropertyOperations {

    PropertyClass pc = new PropertyClass();

    List<PropertyClass> pl = new
ArrayList<PropertyClass>();
```

```
    Scanner s = new Scanner(System.in);

    public void addProperty(ArrayList<PropertyClass>
pl) throws ExceptionClass

    {

        System.out.println("ENTER THE PROPERTY DETAILS
-");

        System.out.print("ENTER THE BASE VALUE OF LAND
- ");

        double basevalue=s.nextDouble();

        if(basevalue<=0)

        {

            throw new ExceptionClass("Base value
should be non zero and positive only");

        }else

        {

        pc.setValue(basevalue);

        }

        System.out.print("ENTER THE BUILT-UP AREA OF
LAND - ");

        pc.setBuilt_up_area(s.nextInt());
        System.out.print("ENTER THE AGE OF LAND - ");
        int age=s.nextInt();
        if(age<=0)
        {
            throw new ExceptionClass("Age of building
should be non-zero positive");
        }
        else
        {
        pc.setAge_of_land(age);
        }
        System.out.print("IS THE LAND LOCATED IN
CITY?(Y:YES,N:NO) - ");
        String located=s.next();
        if(located.equals("y") || located.equals("n")
|| located.equals("Y")||located.equals("N"))
        {
            pc.setLocated(located);
        }

        else

        {

            throw new ExceptionClass("Enter only y for
YES and n for NO");

        }

        PropertyClass pc1 = new
PropertyClass(pc.getValue(), pc.getBuilt_up_area(),
pc.getAge_of_land(),

                pc.getLocated());

        pl.add(pc1);

    }

      public void setTax(ArrayList<PropertyClass> pl)
throws ExceptionClass {

      System.out.println("ENTER THE PROPERTY ID TO
CALCULATE THE TAX - ");

      int id =s.nextInt();

      if(pl.size()==0)

      {

          throw new ExceptionClass("List is empty");
      }

      if(id<0 && id>pl.size())

      {

          throw new ExceptionClass("Id value must be
starting from 1");

      }

      else

      {

     if(pl.get(id-1).located.equalsIgnoreCase("y"))

         {


pl.get(id-1).tax=(pl.get(id-1).built_up_area*pl.get(id-
1).age_of_land*pl.get(id-1).value)+(0.5*pl.get(id-1).bu
ilt_up_area);

         }

         else

         {


pl.get(id-1).tax=(pl.get(id-1).built_up_area*pl.get(id-
1).age_of_land*pl.get(id-1).value);
```

```
          }

          System.out.println("PROPERTY TAX FOR PROPERTY
ID - "+id+" IS "+pl.get(id-1).tax);

      }

      }

    public void displayDetails(ArrayList<PropertyClass>
pl)

    {

System.out.println("====================================
=================================");

         System.out.println("ID\t\tBUILT-UP AREA\tBASE
PRICE\tAGE(YEARS)\tIN CITY\t\tPROPERTY TAX");


System.out.println("====================================
=================================");

         int i=1;

         for (PropertyClass pc : pl) {

             System.out.print(i+"\t\t");

             System.out.print(pc.built_up_area+
"\t\t");

             System.out.print(pc.value + "\t\t");
```

```
            System.out.print(pc.age_of_land +
"\t\t\t");

            System.out.print(pc.located+"\t\t");

            System.out.print(pc.tax);

            System.out.println();

            i++;

        }

    }}}
```

VehicleOperations.java

```
package Com.tax.calculation;

import java.io.BufferedReader;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class VehicleOperations {

    VehicleClass vc = new VehicleClass();

    Scanner s = new Scanner(System.in);
    List<VehicleClass> vl = new
ArrayList<VehicleClass>();

    public void
addVehicelDetails(ArrayList<VehicleClass> vl) throws
ExceptionClass

    {

System.out.print("ENTER THE VEHICLE REGISTRATION
NUMBER");

        int reg_number=s.nextInt();

        int dummy=reg_number;

        int count=0;

        while(dummy!=0)

        {

            dummy/=10;

            count++;

        }

        if(count!=4||reg_number==0000)

        {

            throw new ExceptionClass("Please enter the
valid registration number");
```

```
         }else
         {
         vc.setRegistration_number(reg_number);
         }
         System.out.print("ENTER THE BRAND OF THE
VEHICLE");
         String brand=s.next();
         vc.setBrand(brand);
         System.out.print("ENTER THE MAXIMUM VELOCITY OF
THE VEHICLE(KMPH) - ");
         int velocity=s.nextInt();
         if(velocity<120 || velocity>300 )
         {
             throw new ExceptionClass("Velocity must be
in a range between 120kmph-300kmph");
         }else
         {
         vc.setMax_velocity(velocity);
         }
```

```
        System.out.print("ENTER CAPACITY(NUMBER OF
SEATS) OF THE VEICLE - ");

        int seats=s.nextInt();

        if(seats<2 || seats >50)

        {

            throw new ExceptionClass("Seats range
should be 2 to 50");

        }

        else

        {

        vc.setNo_of_seats(seats);

        }

        System.out.print("CHOOSE THE TYPE OF VEHICLE -
\n1.PETROL DRIVEN\n2.DIESEL DRIVEN\n3.CNG/LPG DRIVEN");

        int vchoice=s.nextInt();

        if(vchoice<0 || vchoice>3)

        {

            throw new ExceptionClass("Select with the
range only");
```

```
        }
        else
        {
        switch(vchoice)
        {
        case 1:vc.setType_of_vehicle("PETROL");
        break;
        case 2:vc.setType_of_vehicle("DIESEL");
        break;
        case 3:vc.setType_of_vehicle("CNG/LPG");
        break;
        }
        }
        System.out.print("ENTER THE PURCHASE COST OF
THE VEHICLE-");
        double cost=s.nextDouble();
        if(cost<50000 || cost >100000)
        {
            throw new ExceptionClass("cost must be
within range of 50000 - 100000");

        }

        else

        {

            vc.setPurchase_cost(cost);

        }



        vl.add(new
VehicleClass(vc.getRegistration_number(),vc.getBrand(),
vc.getMax_velocity(),vc.getNo_of_seats(),vc.getType_of_
vehicle(),vc.getPurchase_cost(),vc.getVehicle_tax()));

    }

    public void setTax(ArrayList<VehicleClass> vl)
throws ExceptionClass

    {   if(vl.size()==0)

    {

        throw new ExceptionClass("Cannot perform on
empty list");

    }
    else

    {

        System.out.print("ENTER THE REGISTRATION NO OF
VEHICLE TO CALCULATE TAX - ");

        int reg_no=s.nextInt();

        for(VehicleClass vc1:vl)

        {

            if(vc1.registration_number == reg_no)

            {


if(vc1.type_of_vehicle.equals("PETROL"))

                {


vc1.vehicle_tax=Math.round(vc1.max_velocity+0.1*vc1.pur
chase_cost);

                }

                else
if(vc1.type_of_vehicle.equals("DIESEL"))

                {
```

```
vc1.vehicle_tax=Math.round(vc1.max_velocity+0.11*vc1.pu
rchase_cost);         }

                else
if(vc1.type_of_vehicle.equals("CNG/LPG"))

                {

vc1.vehicle_tax=Math.round(vc1.max_velocity+0.12*vc1.pu
rchase_cost);

                }

                System.out.println("VEHICLE TAX FOR
REGISTRATION NO - "+reg_no + " IS "+vc1.vehicle_tax);

            }

            else

            {

                System.out.println("Reg number not
Found");

            }

        }

    }

    }
    public void displayVehicel(ArrayList<VehicleClass>
vl)

    {


System.out.println("===================================
=====");

        System.out.println("|
REGISTRATION_NO\tBRAND\tMAX.VELOCITY\tNO.OF.SEATS\tVEHI
CLE TYPE\tPURCHASE COST\tVEHICLE TAX |");


System.out.println("===================================
=");

        for (VehicleClass vc : vl) {

            System.out.print(vc.registration_number+
"\t\t");

            System.out.print(vc.brand + "\t\t");

            System.out.print(vc.max_velocity +
"\t\t\t");

            System.out.print(vc.no_of_seats+"\t\t");


System.out.print(vc.type_of_vehicle+"\t\t");

            System.out.print(vc.purchase_cost+"\t\t");
            System.out.print(vc.vehicle_tax);
            System.out.println();
    }
    }
    public void totalTax(ArrayList<PropertyClass>
pl,ArrayList<VehicleClass> vl)
        {
            double propertytax=0;
            double vehicletax=0;
            for(PropertyClass pc:pl)
            {

                propertytax+=pc.tax;
            }
            for(VehicleClass vc:vl)
            {

                vehicletax+=vc.vehicle_tax;
```

```
             }


System.out.println("+----------------------------------
-----------------+");

             System.out.println("| SR. NO.
PARTICULAR\t\tQUANTITY\tTAX |");


System.out.println("+----------------------------------
-----------------+");

             System.out.print("|   1\t");

             System.out.print("PROPERTIES\t\t");

            System.out.print(pl.size()+"\t\t");

            System.out.print(propertytax+"  |\t\t");

            System.out.println();

            System.out.print("|    2\t");

            System.out.print("VEHICLES\t\t");

            System.out.print(vl.size()+"\t\t");

            System.out.println(vehicletax+"  |\t\t");
System.out.println("+----------------------------------
-----------------+");

            System.out.println("|
TOTAL-------------"+(pl.size()+vl.size())+"\t"+(propert
ytax+vehicletax)+" |");


System.out.println("+----------------------------------
-----------------+");

}

}
```

ExceptionClass.java

```
package Com.tax.calculation;

public class ExceptionClass extends Exception{

    String msg;

    public ExceptionClass(String msg) {

        super(msg);

    }

}
```

Main.java

```
package Com.tax.calculation;



import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;



public class Main {

    static final String username1="admin";

    static String password1 = "admin";

    public static void main(String[] args) throws
ExceptionClass

    {

        List<PropertyClass> pl= new
ArrayList<PropertyClass>();

        List<VehicleClass> vl= new
ArrayList<VehicleClass>();
```

```
System.out.println("+-----------------------------------
------------+");

        System.out.println("| WELCOME TO TAXA(TAX
CALCULATION APPLICATION) |");


System.out.println("+-----------------------------------
------------+");

        System.out.println("PLEASE LOGIN TO CONTINUE
-");

        Scanner s = new Scanner(System.in);

        System.out.print("USERNAME - ");

        String username = s.next();

        System.out.print("PASSWORD - ");

        String password = s.next();

        ArrayList<PropertyClass> properties = new
ArrayList<>();

        ArrayList<VehicleClass> vehicles = new
ArrayList<>();


if(username.equals(username1)&&password.equals(password
1))
```

```
        {
            boolean condition=true;
            while(condition)
            {
                System.out.println("1.PROPERTY TAX\n2.VEHICLE TAX\n3.TOTAL\n4.EXIT");
                int choice1=s.nextInt();
                if(choice1<0)
                {
                    throw new ExceptionClass("Enter only in range from 0");
                }

                boolean b=true;
                while(b)
                {
                switch(choice1)
                {
```

```
                case 1:

                    PropertyOperations po = new 
PropertyOperations();

                System.out.println("1.ADD PROPERTY 
DETAILS\n2.CALCULATE PROPERTY TAX\n3.DISPLAY ALL 
PROPERTIES\n4.BACK TO MAIN MENU");

                switch (s.nextInt())

                {

                case 1:

                    try {

                        po.addProperty(properties);

                    } catch (ExceptionClass e) {

                        // TODO Auto-generated catch 
block


System.out.println(e.getMessage());

                        po.addProperty(properties);

                    }

                    break;

                case 2:
                    try {

                        po.setTax(properties);

                    } catch (ExceptionClass e) {

                        // TODO Auto-generated catch
block

System.out.println(e.getMessage());

                        po.addProperty(properties);

                    }

                    break;

                case 3:

                    po.displayDetails(properties);

                    break;

                case 4:

                    b = false;

                    break;

                }

                break;
```

```
                case 2:
                    while(b)
                    {
                        VehicleOperations vo = new
VehicleOperations();
                        System.out.println("1.ADD
VEHICLE DETAILS\n2.CALCULATE VEHICLE TAX\n3.DISPLAY ALL
VEHICLES\n4.BACK TO MAIN MENU");
                        switch (s.nextInt())
                        {
                        case 1:
                            try {

vo.addVehicelDetails(vehicles);
                            } catch (ExceptionClass
e) {
                                // TODO
Auto-generated catch block

System.out.println(e.getMessage());
                            }
```

```
                              break;

                          case 2:

                              try {

                                  vo.setTax(vehicles);

                              } catch (ExceptionClass
e) {

                                  // TODO
Auto-generated catch block


System.out.println(e.getMessage());

                              }

                              break;

                          case 3:


vo.displayVehicel(vehicles);

                              break;

                          case 4:

                              b = false;

                              break;
```

```
                        }
                    }
                    break;
                case 3:
                    while(b)
                    {
                    VehicleOperations to = new
VehicleOperations();
                    to.totalTax(properties, vehicles);
                    b=false;
                    }
                    break;
                case 4:
                    System.out.println("THANK YOU
VISIT AGAIN");
                    System.exit(0);
                    break;
                    default:
                          System.out.println("Invalid
choise");
                  }
                      }
                  }
                  }
         else
         {
             System.out.println("Invalid
username/password");
         }
    }
}
```