

Mini Project 3

Package.java

```
package com.model;
```

```
public class Package {  
    private String packageld;  
    private String sourcePlace;  
    private String destinationPlace;  
    private int noOfDays;  
    private double packageCost;  
    private double basicFare;  
    public String getPackageld() {  
        return packageld;  
    }  
    public void setPackageld(String packageld) {  
        this.packageld = packageld;  
    }  
    public String getSourcePlace() {  
        return sourcePlace;  
    }  
    public void setSourcePlace(String sourcePlace) {  
        this.sourcePlace = sourcePlace;  
    }  
    public String getDestinationPlace() {  
        return destinationPlace;  
    }  
    public void setDestinationPlace(String destinationPlace) {  
        this.destinationPlace = destinationPlace;  
    }  
    public int getNoOfDays() {  
        return noOfDays;  
    }  
    public void setNoOfDays(int noOfDays) {  
        this.noOfDays = noOfDays;  
    }  
    public double getPackageCost() {  
        return packageCost;  
    }  
    public void setPackageCost(double packageCost) {  
        this.packageCost = packageCost;  
    }  
    public double getBasicFare() {  
        return basicFare;  
    }  
}
```

```

    public void setBasicFare(double basicFare) {
        this.basicFare = basicFare;
    }
    @Override
    public String toString() {
        return " packageId=" + packageId + ",\n sourcePlace=" + sourcePlace + ",\n
destinationPlace=" + destinationPlace
        + ",\n noOfDays=" + noOfDays + ",\n basicFare=" + basicFare + ",\n packageCost=" +
packageCost ;
    }
}

```

2.PackageDao.java

```

package com.model;
public class Package {
    private String packageId;
    private String sourcePlace;
    private String destinationPlace;
    private int noOfDays;
    private double packageCost;
    private double basicFare;
    public String getPackageId() {
        return packageId;
    }
    public void setPackageId(String packageId) {
        this.packageId = packageId;
    }
    public String getSourcePlace() {
        return sourcePlace;
    }
    public void setSourcePlace(String sourcePlace) {
        this.sourcePlace = sourcePlace;
    }
    public String getDestinationPlace() {
        return destinationPlace;
    }
    public void setDestinationPlace(String destinationPlace) {
        this.destinationPlace = destinationPlace;
    }
    public int getNoOfDays() {
        return noOfDays;
    }
    public void setNoOfDays(int noOfDays) {
        this.noOfDays = noOfDays;
    }
}

```

```

public double getPackageCost() {
    return packageCost;
}
public void setPackageCost(double packageCost) {
    this.packageCost = packageCost;
}
public double getBasicFare() {
    return basicFare;
}
public void setBasicFare(double basicFare) {
    this.basicFare = basicFare;
}
@Override
public String toString() {
    return " packageId=" + packageId + ",\n sourcePlace=" + sourcePlace + ",\n
destinationPlace=" + destinationPlace
        + ",\n noOfDays=" + noOfDays + ",\n basicFare=" + basicFare + ",\n packageCost=" +
packageCost ;
}
}

```

3.packageDaoImpl

```

package com.dao;
import java.util.ArrayList;
import java.util.List;
import com.exception.InvalidPackageIdException;
import com.model.Package;
public class PackageDaoImpl implements PackageDao {
    private List<Package> packageList = new ArrayList<>();
    @Override
    public void addPackage(Package p) {
        packageList.add(p);
    }
    @Override
    public List<Package> fetchAllPackages() {
        return new ArrayList<>(packageList); // return a copy to avoid modification outside
    }
    @Override
    public Package findPackageById(String packageId) {
        for (Package p : packageList) {
            if (p.getPackageId().equalsIgnoreCase(packageId)) {
                return p;
            }
        }
    }
}

```

```

        return null;
    }
    @Override
    public double calculatePackageCost(String packageld) throws InvalidPackageldException {
        Package p = findPackageByld(packageld);
        if (p == null) {
            throw new InvalidPackageldException("Invalid Package ID: " + packageld);
        }
        double baseCost = p.getBasicFare() * p.getNoOfDays();
        double discount = 0.0;
        int days = p.getNoOfDays();
        if (days > 5 && days <= 8) {
            discount = 0.03 * baseCost;
        } else if (days > 8 && days <= 10) {
            discount = 0.05 * baseCost;
        } else if (days > 10) {
            discount = 0.07 * baseCost;
        }
        double costAfterDiscount = baseCost - discount;
        double gst = 0.12 * costAfterDiscount;
        double finalCost = costAfterDiscount + gst;
        p.setPackageCost(finalCost);
        return finalCost;
    }
}

```

4.PackageService.java

```

package com.service;
import java.util.List;
import com.model.Package;
import com.exception.InvalidPackageldException;
public interface PackageService {
    void addPackage(Package p) throws InvalidPackageldException;
    List<Package> fetchAllPackages();
    Package findPackageByld(String packageld);
    double calculatePackageCost(String packageld) throws InvalidPackageldException;
}

```

5.PackageServiceImpl

```

package com.service;
import java.util.List;
import com.dao.PackageDao;
import com.dao.PackageDaoImpl;
import com.model.Package;
import com.exception.InvalidPackageldException;

```

```

public class PackageServiceImpl implements PackageService {
    private PackageDao packageDao = new PackageDaoImpl();
    @Override
    public double calculatePackageCost(String packageld) throws InvalidPackageldException {
        return packageDao.calculatePackageCost(packageld);
    }
    @Override
    public void addPackage(Package p) throws InvalidPackageldException {
        if (p.getPackageld() == null || p.getPackageld().length() != 7) {
            throw new InvalidPackageldException("Invalid Package Id");
        }
        packageDao.addPackage(p);
    }
    @Override
    public List<Package> fetchAllPackages() {
        return packageDao.fetchAllPackages();
    }
    @Override
    public Package findPackageById(String packageld) {
        return packageDao.findPackageById(packageld);
    }
}

```

6.Main.java

```

package com.client;
import java.util.List;
import java.util.Scanner;
import com.model.Package;
import com.service.PackageService;
import com.service.PackageServiceImpl;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        PackageService service = new PackageServiceImpl();
        int choice;
        do {
            System.out.println("\n1. Add Package");
            System.out.println("2. Display All Packages");
            System.out.println("3. Search Package by ID");
            System.out.println("4. Calculate Package Cost");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            sc.nextLine();

```

```

switch (choice) {
    case 1:
        try {
            Package p = new Package();
            System.out.print("Enter Package ID: ");
            p.setPackageId(sc.nextLine());
            System.out.print("Enter Source Place: ");
            p.setSourcePlace(sc.nextLine());
            System.out.print("Enter Destination Place: ");
            p.setDestinationPlace(sc.nextLine());
            System.out.print("Enter No. of Days: ");
            p.setNoOfDays(sc.nextInt());
            System.out.print("Enter Basic Fare: ");
            p.setBasicFare(sc.nextDouble());
            sc.nextLine();
            service.addPackage(p);
            System.out.println("Package added successfully.");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        break;
    case 2:
        List<Package> packages = service.fetchAllPackages();
        for (Package pkg : packages) {
            System.out.println(pkg);
        }
        break;
    case 3:
        System.out.print("Enter Package ID to search: ");
        String searchId = sc.nextLine();
        Package result = service.findPackageById(searchId);
        if (result != null)
            System.out.println(result);
        else
            System.out.println("Package not found.");
        break;
    case 4:
        System.out.print("Enter Package ID to calculate cost: ");
        String pkgId = sc.nextLine();
        try {
            double cost = service.calculatePackageCost(pkgId);
            System.out.println("Total Package Cost (with GST): ₹" + cost);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
}

```

```

        }
        break;
    case 5:
        System.out.println("Exiting program.");
        break;
    default:
        System.out.println("Invalid choice!");
    }
} while (choice != 5);
sc.close();
}
}

```

7.InvalidPackageldException

```

package com.exception;
public class InvalidPackageldException extends Exception {
    public InvalidPackageldException(String message) {
        super(message);
    }
}

```

Output:

```

1. Add Package
2. Display All Packages
3. Search Package by ID
4. Calculate Package Cost
5. Exit
Enter your choice: 1
Enter Package ID: 1234567
Enter Source Place: erode
Enter Destination Place: cbe
Enter No. of Days: 60
Enter Basic Fare: 500
Package added successfully.
1. Add Package
2. Display All Packages
3. Search Package by ID
4. Calculate Package Cost
5. Exit
Enter your choice: 2
packageId=1234567,
sourcePlace=erode,
destinationPlace=cbe,
noOfDays=60,
basicFare=500.0,

```

packageCost=0.0

1. Add Package
2. Display All Packages
3. Search Package by ID
4. Calculate Package Cost
5. Exit

Enter your choice: 1234567

Invalid choice!

1. Add Package
2. Display All Packages
3. Search Package by ID
4. Calculate Package Cost
5. Exit

Enter your choice: 3

Enter Package ID to search: 1234567

packageId=1234567,
sourcePlace=erode,
destinationPlace=cbe,
noOfDays=60,
basicFare=500.0,
packageCost=0.0

1. Add Package
2. Display All Packages
3. Search Package by ID
4. Calculate Package Cost
5. Exit

Enter your choice: 4

Enter Package ID to calculate cost: 1234567

Total Package Cost (with GST): ₹31248.0

1. Add Package
2. Display All Packages
3. Search Package by ID
4. Calculate Package Cost
5. Exit

Enter your choice: 5

Exiting program.