

Assignment no5 Exception Handling

Q1. Electricity Bill Calculation with Exception Handling

Design a Java program to calculate the electricity bill for a customer, including exception handling for invalid input values. Implement a class named ElectricityBill with the following specifications:

Class: ElectricityBill

Instance Variables

- customerName (String): Name of the customer
- unitsConsumed (double): Number of electricity units consumed
- billAmount (double): The calculated bill amount

Constructor

- A parameterized constructor to initialize the customerName and unitsConsumed fields.
- Throw an IllegalArgumentException if unitsConsumed is negative.

Method

- void calculateBillAmount(): This method calculates the electricity bill based on the following rules:
 - First 100 units: Rs. 5 per unit
 - Next 200 units (101–300): Rs. 7 per unit
 - Above 300 units: Rs. 10 per unit

Main Program

In the main() method:

1. Prompt the user to enter the customer's name and units consumed.
2. Use try-catch blocks to handle the following scenarios:
 - Catch InputMismatchException if the user enters non-numeric data for units.
 - Catch IllegalArgumentException if a negative value is entered for units.
3. If the input is valid, create an object of the ElectricityBill class, compute the bill using calculateBillAmount(), and print the customer's name, units consumed, and the total bill amount

```

import java.util.Scanner,

class ElectricityBill {

    private String customerName,
    private double unitsConsumed,
    private double billAmount,

    public ElectricityBill(String customerName, double unitsConsumed) {
        if (unitsConsumed < 0) {
            throw new IllegalArgumentException("Units consumed cannot be negative.");
        }
        this.customerName = customerName,
        this.unitsConsumed = unitsConsumed,
    }

    public void calculateBillAmount() {
        double units = unitsConsumed,
        if (units <= 100) {
            billAmount = units * 5,
        } else if (units <= 300) {
            billAmount = (100 * 5) + ((units - 100) * 7),
        } else {
            billAmount = (100 * 5) + (200 * 7) + ((units - 300) * 10),
        }
    }

    public void printBill() {
        System.out.println("Customer Name: " + customerName),
        System.out.println("Units Consumed: " + unitsConsumed),
        System.out.println("Total Bill Amount: Rs. " + billAmount),
    }
}

public class Bill {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in),

        try {

            System.out.print("Enter customer name: ");
            String name = scanner.nextLine(),

            System.out.print("Enter units consumed: ");
            double units = scanner.nextDouble(),

            ElectricityBill bill = new ElectricityBill(name, units),
            bill.calculateBillAmount(),
            bill.printBill(),

        } catch (InputMismatchException e) {
            System.out.println("Invalid input! Please enter a valid number for units consumed.");
        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage()),
        }
    }
}

```

Q2. Student Marks and Grade Calculation with Exception Handling

Design a Java program to calculate the total marks, average, and grade of a student, with proper exception handling for invalid inputs. Implement a class named Student with the following specifications:

Class: Student

Instance Variables

- name (String): Name of the student
- rollNo (int): Roll number of the student
- marks (double array of size 5): Marks obtained in 5 subjects
- average (double): Average marks
- grade (char): Grade based on average

Constructor

- A parameterized constructor to initialize the name, rollNo, and marks.
- Throw an IllegalArgumentException if any mark is negative or greater than 100.

Methods

- void calculateAverage(): Computes the average of marks.
- void calculateGrade(): Assigns grade based on the average as per the following

criteria:

- A: average ≥ 90
- B: $80 \leq \text{average} < 90$
- C: $70 \leq \text{average} < 80$
- D: $60 \leq \text{average} < 70$
- F: average < 60
- void displayStudentInfo(): Displays the student's name, roll number, marks, average, and grade.

Main Program

In the main() method:

1. Prompt the user to input student details and marks for 5 subjects.
2. Use a try-catch block to handle the following:
 - InputMismatchException for non-numeric input
 - IllegalArgumentException for invalid mark entries (e.g., < 0 or > 100)

3. Create a Student object, calculate average and grade, and display the full information

Answer

```
import java.util.InputMismatchException;
```

```
import java.util.Scanner;
```

```
class Student {
```

```
    private String name;
```

```
    private int rollNo;
```

```
    private double[] marks = new double[5];
```

```
    private double average;
```

```
    private char grade;
```

```
    public Student(String name, int rollNo, double[] marks) {
```

```
        this.name = name;
```

```
        this.rollNo = rollNo;
```

```
        for (double mark : marks) {
```

```
            if (mark < 0 || mark > 100) {
```

```
                throw new IllegalArgumentException("Marks should be between 0 and 100.");
```

```
            }
```

```
        }
```

```
        this.marks = marks;
```

```
    }
```

```
    public void calculateAverage() {
```

```
        double total = 0;
```

```
        for (double mark : marks) {
```

```
            total += mark;
```

```
        }
```

```
        this.average = total / marks.length;
```

```
}
```

```
public void calculateGrade() {
```

```
    if (average >= 90) {
```

```
        grade = 'A';
```

```
    } else if (average >= 80) {
```

```
        grade = 'B';
```

```
    } else if (average >= 70) {
```

```
        grade = 'C';
```

```
    } else if (average >= 60) {
```

```
        grade = 'D';
```

```
    } else {
```

```
        grade = 'F';
```

```
    }
```

```
}
```

```
public void displayStudentInfo() {
```

```
    System.out.println("Student Name: " + name);
```

```
    System.out.println("Roll Number: " + rollNo);
```

```
    System.out.print("Marks: ");
```

```
    for (double mark : marks) {
```

```
        System.out.print(mark + " ");
```

```
    }
```

```
    System.out.println("\nAverage Marks: " + average);
```

```
    System.out.println("Grade: " + grade);
```

```
}
```

```
}
```

```
public class StudentInfo {
```

```
    public static void main(String[] args) {
```

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

```
try {

    System.out.print("Enter student name: ");

    String name = scanner.nextLine();

    System.out.print("Enter roll number: ");

    int rollNo = scanner.nextInt();

    double[] marks = new double[5];

    System.out.println("Enter marks for 5 subjects:");

    for (int i = 0; i < 5; i++) {

        System.out.print("Subject " + (i + 1) + ": ");

        marks[i] = scanner.nextDouble();

    }

    Student student = new Student(name, rollNo, marks);

    student.calculateAverage();

    student.calculateGrade();

    student.displayStudentInfo();

} catch (InputMismatchException e) {

    System.out.println("Invalid input! Please enter numeric values for roll number and marks.");

} catch (IllegalArgumentException e) {

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

}

}
```

