Experiment 10 Date: 14/11/2023

Roll number: 22z215

Exception Handling

Consider a program that reads an integer from the user and prints its reciprocal. If the user enters zero, the program will throw an exception. Write the code to handle this exception using a try-catch block.

Aim:
To write a java program to handle exceptions using try-catch block.

Code:

```java
package Exp10;

import java.util.Scanner;

public class Q1_Reciprocal {
    public static void main(String[] args) {
        try {
            Scanner scanner = new Scanner(System.in);

            System.out.print("Enter an integer: ");
            int number = scanner.nextInt();
            scanner.close();
            if (number == 0) {
                throw new ArithmeticException("Cannot calculate reciprocal for
zero.");
            }

            double reciprocal = 1.0 / number;
            System.out.println("Reciprocal: " + reciprocal);
        } catch (ArithmeticException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Output:

```
Enter an integer: 0
Exception: Cannot calculate reciprocal for zero.
```

2. Define a class that has static methods main(), p(), q(), r() and s(). The main() method calls p(). Method p() calls q(). Method q() calls r(). Method r() calls s().Method s() declares a local array with six integer type elements and then attempts to access the element at position 10. Therefore, an exception of type ArrayIndexOutOfBoundsException is generated. Each method has a catch block for this type of exception. The catch blocks in q(), r() and s() contain a throw statement to generate this type of exception. Indicate the flow of control during the execution by displaying the suitable output messages.Also include a finally block with print statements.

Aim:

To write a java program to handle exceptions using try-catch block.

Code:

```java
package Exp10;

public class Q2 {
    public static void main(String[] args) {
        try {
            mainMethod();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception in mainMethod: " + e.getMessage());
        } finally {
            System.out.println("Finally block in main");
        }
    }

    static void mainMethod() {
        try {
            p();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception in mainMethod: " + e.getMessage());
        } finally {
            System.out.println("Finally block in mainMethod");
        }
    }

    static void p() {
        try {
            q();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception in p: " + e.getMessage());
        } finally {
            System.out.println("Finally block in p");
        }
    }

    static void q() {
        try {
            r();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception in q: " + e.getMessage());
            throw e; // Re-throwing the exception
        } finally {
            System.out.println("Finally block in q");
        }
    }

    static void r() {
        try {
            s();
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception in r: " + e.getMessage());
            throw e; // Re-throwing the exception
        } finally {
            System.out.println("Finally block in r");
        }
    }

    static void s() {
        try {
            int[] array = new int[6];
            array[10] = 42; // This will generate
ArrayIndexOutOfBoundsException
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception in s: " + e.getMessage());
            throw e; // Re-throwing the exception
        } finally {
            System.out.println("Finally block in s");
        }
    }
}
```

Output:

```
Exception in s: Index 10 out of bounds for length 6
Finally block in s
Exception in r: Index 10 out of bounds for length 6
Finally block in r
Exception in q: Index 10 out of bounds for length 6
Finally block in q
Exception in p: Index 10 out of bounds for length 6
Finally block in p
Finally block in mainMethod
Finally block in main
```

3. Create a class Student with the data members Name, Register Number, Mark1, Mark2, Mark3, Total and Average. Using necessary member functions get the input, calculate Total and Average and display the student information’s. If the register number exceeds 6 digits then raise one user defined exception called InvalidRegNoException. Similarly if marks value is greater than 100 throw an exception called MarkOutOfBoundsException

Aim:

To write a java program to handle exceptions using try-catch block.

Code:

```
package Exp10;

import java.util.Scanner;

public class Q3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.println("Enter student details:");
            System.out.print("Name: ");
            String name = scanner.nextLine();

            System.out.print("Register Number: ");
            int regNo = scanner.nextInt();

            System.out.print("Mark 1: ");
            int mark1 = scanner.nextInt();

            System.out.print("Mark 2: ");
            int mark2 = scanner.nextInt();

            System.out.print("Mark 3: ");
            int mark3 = scanner.nextInt();

            // Create a Student object
            Student student = new Student();

            // Use the getInput method to set student details, handling
exceptions
            student.getInput(name, regNo, mark1, mark2, mark3);

            // Calculate total and average
            student.calculateTotalAndAverage();

            // Display student information
            student.displayStudentInfo();

            scanner.close();
        } catch (InvalidRegNoException | MarkOutOfBoundsException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }

}

class InvalidRegNoException extends Exception {
    InvalidRegNoException(String message) {
        super(message);
    }
}

class MarkOutOfBoundsException extends Exception {
    MarkOutOfBoundsException(String message) {
        super(message);
    }
}

class Student {
    private String name;
    private int regNo;
    private int mark1;
    private int mark2;
    private int mark3;
    private int total;
    private double average;

    public void getInput(String name, int regNo, int mark1, int mark2, int mark3)
            throws InvalidRegNoException, MarkOutOfBoundsException {
        if (regNo > 999999) {
            throw new InvalidRegNoException("Invalid Register Number");
        }
        if (mark1 > 100 || mark2 > 100 || mark3 > 100) {
            throw new MarkOutOfBoundsException("Marks should be within 0 to 100");
        }

        this.name = name;
        this.regNo = regNo;
        this.mark1 = mark1;
        this.mark2 = mark2;
        this.mark3 = mark3;
    }

    public void calculateTotalAndAverage() {
        total = mark1 + mark2 + mark3;
        average = total / 3.0;
    }

    public void displayStudentInfo() {
        System.out.println("Student Information:");
        System.out.println("Name: " + name);
        System.out.println("Register Number: " + regNo);
        System.out.println("Mark 1: " + mark1);
        System.out.println("Mark 2: " + mark2);
        System.out.println("Mark 3: " + mark3);
        System.out.println("Total: " + total);
        System.out.println("Average: " + average);
    }
}
```

Output:

```
Enter student details:
Name: Me
Register Number: 1234567890
Mark 1: 99
Mark 2: 99
Mark 3: 101
Exception: Invalid Register Number
```

Result:

All the programs are executed and the output are verified.