Rajalakshmi Engineering College

Name: SANJAY V
Email: 241801247@rajalakshmi.edu.in
Roll no: 241801247
Phone: 7397492247
Branch: REC
Department: AI & DS - Section 3
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Write a program to validate the email address and display suitable exceptions if there is any mistake.

Create 3 custom exception classes as below

DotExceptionAtTheRateExceptionDomainException

A typical email address should have a " . " character, and a "@" character, and also the domain name should be valid. Valid domain names for practice be 'in', 'com', 'net', or 'biz'.

Display Invalid Dot usage, Invalid @ usage, or Invalid Domain message based on email id.

Get the email address from the user, validate the email by checking the above-mentioned criteria, and print the validity status of the input email address.

Input Format

The first line of input contains the email to be validated.

Output Format

The output prints a Valid email address or an Invalid email address along with the suitable exception

If email ends with . or contains not exactly one . after @, it throws:

DotException: Invalid Dot usage

Invalid email address

If @ appears not exactly once, it throws:

AtTheRateException: Invalid @ usage

Invalid email address

If the part after the last dot is not among accepted domains:

DomainException: Invalid Domain

Invalid email address

If all conditions satisfied then print:

Valid email address

Refer to the sample input and output for format specifications.

Sample Test Case

Input: sample@gmail.com
Output: Valid email address

Answer

```
import java.util.*;

class DotException extends Exception {
    public DotException(String msg) { super(msg); }
}

class AtTheRateException extends Exception {
    public AtTheRateException(String msg) { super(msg); }
}

class DomainException extends Exception {
    public DomainException(String msg) { super(msg); }
}

class EmailValidator {
    public void validate(String email) throws DotException, AtTheRateException,
DomainException {
        if (email.startsWith("@") || email.endsWith("@") || email.startsWith(".") ||
email.endsWith(".")) {
            throw new DotException("Invalid Dot usage");
        }
        int atCount = 0;
        for (char c : email.toCharArray()) if (c == '@') atCount++;
        if (atCount != 1) throw new AtTheRateException("Invalid @ usage");
        int atIndex = email.indexOf('@');
        int lastDot = email.lastIndexOf('.');
        if (lastDot < atIndex + 2 || lastDot == email.length() - 1) throw new
DotException("Invalid Dot usage");
        String domain = email.substring(lastDot + 1);
        if (!domain.equals("in") && !domain.equals("com") && !domain.equals("net")
&& !domain.equals("biz"))
            throw new DomainException("Invalid Domain");
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String email = sc.nextLine();
        EmailValidator validator = new EmailValidator();
        try {
            validator.validate(email);
            System.out.println("Valid email address");
        } catch (DotException e) {
            System.out.println("DotException: " + e.getMessage());
            System.out.println("Invalid email address");
        } catch (AtTheRateException e) {
            System.out.println("AtTheRateException: " + e.getMessage());
            System.out.println("Invalid email address");
        } catch (DomainException e) {
            System.out.println("DomainException: " + e.getMessage());
            System.out.println("Invalid email address");
        }
        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***