Rajalakshmi Engineering College

Name: Kirithick R
Email: 240701627@rajalakshmi.edu.in
Roll no: 2116240701627
Phone: 9952595005
Branch: REC
Department: CSE - Section 6
Batch: 2028
Degree: B.E - CSE

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.Scanner;

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

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

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

class EmailValidator {
    private static final String[] VALID_DOMAINS = {"in", "com", "net", "biz"};

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        if (!sc.hasNextLine()) {
            sc.close();
            return;
        }
        String email = sc.nextLine().trim();
        sc.close();

        try {
            validateEmail(email);
            System.out.println("Valid email address");
        } catch (DotException de) {
            System.out.println("DotException: " + de.getMessage());
            System.out.println();
            System.out.println("Invalid email address");
        } catch (AtTheRateException ae) {
            System.out.println("AtTheRateException: " + ae.getMessage());
            System.out.println();
            System.out.println("Invalid email address");
        } catch (DomainException de) {
            System.out.println("DomainException: " + de.getMessage());
            System.out.println();
            System.out.println("Invalid email address");
        }
    }

    private static void validateEmail(String email) throws DotException,
AtTheRateException, DomainException {
        if (email == null) {
            throw new AtTheRateException("Invalid @ usage");
        }

        int atCount = 0;
        for (char c : email.toCharArray()) {
            if (c == '@') atCount++;
        }
        if (atCount != 1) {
            throw new AtTheRateException("Invalid @ usage");
        }

        int atIndex = email.indexOf('@');
        if (atIndex == 0 || atIndex == email.length() - 1) {
            throw new AtTheRateException("Invalid @ usage");
        }

        if (email.endsWith(".")) {
            throw new DotException("Invalid Dot usage");
        }

        String domainPart = email.substring(atIndex + 1);
        int dotCountInDomain = 0;
        for (char c : domainPart.toCharArray()) {
            if (c == '.') dotCountInDomain++;
        }
        if (dotCountInDomain != 1) {
            throw new DotException("Invalid Dot usage");
        }

        int lastDotIndex = email.lastIndexOf('.');
        String extension = email.substring(lastDotIndex + 1).toLowerCase();

        boolean valid = false;
        for (String d : VALID_DOMAINS) {
            if (d.equals(extension)) {
                valid = true;
                break;
            }
        }
        if (!valid) {
            throw new DomainException("Invalid Domain");
        }
    }
}
```

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