

Rajalakshmi Engineering College

Name: shyam sundar

Email: 241801268@rajalakshmi.edu.in

Roll no: 241801268

Phone: 8667394275

Branch: REC

Department: AI & DS - Section 4

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

DotException AtTheRateException DomainException

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 {
    DotException(String message) {
        super(message);
    }
}
class AtTheRateException extends Exception {
    AtTheRateException(String message) {
        super(message);
    }
}
class DomainException extends Exception {
    DomainException(String message) {
        super(message);
    }
}
class EmailValidator {
    public static void validateEmail(String email) throws DotException,
    AtTheRateException, DomainException {
        String[] validDomains = {"in", "com", "net", "biz"};
        int atCount = email.length() - email.replace("@", "").length();
        if (atCount != 1) {
            throw new AtTheRateException("Invalid @ usage");
        }
        if (email.startsWith(".") || email.endsWith(".") || email.startsWith("@") ||
        email.endsWith("@")
        || email.contains("..") || email.contains("@.") || email.contains(".@")) {
            throw new DotException("Invalid Dot usage");
        }
        String[] parts = email.split("@");
        String domainPart = parts[1];
```

```

if (!domainPart.contains(".")) {
    throw new DotException("Invalid Dot usage");
}
int lastDot = domainPart.lastIndexOf('.');
String domainExt = domainPart.substring(lastDot + 1);
boolean valid = false;
for (String d : validDomains) {
    if (d.equals(domainExt)) {
        valid = true;
        break;
    }
}
if (!valid) {
    throw new DomainException("Invalid Domain");
}
}
}

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

```

Status : Correct

Marks : 10/10