

Rajalakshmi Engineering College

Name: Kanishka S
Email: 240701227@rajalakshmi.edu.in
Roll no: 2116240701227
Phone: 8825651385
Branch: REC
Department: CSE - Section 4
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

```
// You are using Java
// You are using Java
import java.util.*;

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);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String email = sc.nextLine();
        sc.close();

        try {
            // Check @ usage
            int atCount = email.length() - email.replace("@", "").length();
            if (atCount != 1 || email.startsWith "@" || email.endsWith "@") {
```

```
        throw new AtTheRateException("Invalid @ usage");
    }

    // Check . usage
    if (email.endsWith(".") || email.startsWith(".") || email.contains("..")) {
        throw new DotException("Invalid Dot usage");
    }

    // Split by @ to check for dot in domain part
    String[] parts = email.split("@");
    if (parts.length != 2 || !parts[1].contains(".")) {
        throw new DotException("Invalid Dot usage");
    }

    // Check domain
    String domainPart = parts[1];
    String[] domainSplit = domainPart.split("\\.");
    String domainExtension = domainSplit[domainSplit.length - 1];

    if (!(domainExtension.equals("in") || domainExtension.equals("com") ||
          domainExtension.equals("net") || domainExtension.equals("biz")))) {
        throw new DomainException("Invalid Domain");
    }

    System.out.print("Valid email address");
}

catch (DotException e) {
    System.out.print("DotException: " + e.getMessage() + " Invalid email
address");
}

catch (AtTheRateException e) {
    System.out.print("AtTheRateException: " + e.getMessage() + " Invalid
email address");
}

catch (DomainException e) {
    System.out.print("DomainException: " + e.getMessage() + " Invalid email
address");
}

catch (Exception e) {
    System.out.print("Invalid email address");
}
```

}

Status : Correct

Marks : 10/10