

Rajalakshmi Engineering College

Name: Harish Raj
Email: 240701172@rajalakshmi.edu.in
Roll no: 2116240701172
Phone: 9345116365
Branch: REC
Department: CSE - Section 9
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 DomainException extends Exception {
    String expDescription;
    DomainException(String expDescription) {
        super(expDescription);
    }
}

class DotException extends Exception {
    String expDescription;
    DotException(String expDescription) {
        super(expDescription);
    }
}

class AtTheRateException extends Exception {
    String expDescription;
    AtTheRateException(String expDescription) {
        super(expDescription);
    }
}

class EmailValidationMain {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String email = myObj.next();
        boolean checkEndDot = false;
        checkEndDot = email.endsWith(".");
        int indexOfAt = email.indexOf('@');
        int lastIndexIndexOfAt = email.lastIndexOf('.');
    }
}
```

```
int countOfAt = 0;
for (int i = 0; i < email.length(); i++) {
    if(email.charAt(i)=='@')
        countOfAt++;
}
String buffering = email.substring(email.indexOf('@')+1, email.length());
int len = buffering.length();
int countOfDotAfterAt = 0;
for (int i=0; i < len; i++) {
    if(buffering.charAt(i)=='.')
        countOfDotAfterAt++;
}
String userName = email.substring(0, email.indexOf('@'));
String domainName = email.substring(email.indexOf('.')+1, email.length());
int domainCheck=0;
if((domainName.equals("in")) || (domainName.equals("com")) || (domainName.equals("net")) || (domainName.equals("biz")))
    domainCheck=1;

try {
    if((checkEndDot) || (countOfDotAfterAt!=1)) {
        throw new DotException("Invalid Dot usage");
    }

    if(countOfAt!=1) {
        throw new AtTheRateException("Invalid @ usage");
    }

    if(domainCheck!=1){
        throw new DomainException("Invalid Domain");
    }

}catch(DotException e) {
    System.out.println(e);
}catch(AtTheRateException e) {
    System.out.println(e);
}catch(DomainException e) {
    System.out.println(e);
}

if ((countOfAt==1) && (userName.endsWith("."))==false) && (domainCheck==1) && (countOfDotAfterAt ==1) &&((indexOfAt+3) <=
```

```
(lastIndexOfAt) && !checkEndDot)) {  
    System.out.println("Valid email address");  
}  
  
else {  
    System.out.println("Invalid email address");  
}  
myObj.close();  
}  
}
```

Status : Correct

Marks : 10/10