Rajalakshmi Engineering College

Name: Nishant Gurung
Email: 241501131@rajalakshmi.edu.in
Roll no: 241501131
Phone: 8670069246
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a user registration system, there is a requirement to implement a username validation module. Users attempting to register must adhere to specific criteria for their usernames to be considered valid.

Your task is to develop a program that takes user input for a desired username and validates it according to the following rules:

The username must not contain any spaces.The username must be at least 5 characters long.

Implement a custom exception, InvalidUsernameException, to handle cases where the entered username does not meet the specified criteria.

Input Format

The input consists of a string S, representing the desired username.

Output Format

If the username is valid, print "Username is valid: [S]".

If the username is invalid:

1. If the username is short, print "Invalid Username: Username must be at least 5 characters long"
2. If the username contains spaces, print "Invalid Username: Username cannot contain spaces"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: John
Output: Invalid Username: Username must be at least 5 characters long

Answer

```
import java.util.Scanner;

// Step 1: Create a custom exception class
class InvalidUsernameException extends Exception {
    // Constructor that takes a message and passes it to the base Exception class
    public InvalidUsernameException(String message) {
        super(message);
    }
}

class UsernameValidator {

    // Step 2: Method to validate the username
    public static void validateUsername(String username) throws
InvalidUsernameException {
        // Rule 1: Check if username contains spaces
        if (username.contains(" ")) {

            throw new InvalidUsernameException("Invalid Username: Username
cannot contain spaces");
```

```
        }

        // Rule 2: Check if username is at least 5 characters long
        if (username.length() < 5) {

            throw new InvalidUsernameException("Invalid Username: Username must
be at least 5 characters long");
        }

        // If both checks pass, print success message
        System.out.println("Username is valid: " + username);
    }

    // Step 3: Main method to take input and call validation
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // Create Scanner object to
read input
        String username = scanner.nextLine(); // Read the username from user

        try {
            // Try to validate the input
            validateUsername(username);
        } catch (InvalidUsernameException e) {

            System.out.println(e.getMessage());
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: Nishant Gurung
Email: 241501131@rajalakshmi.edu.in
Roll no: 241501131
Phone: 8670069246
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A local municipality is implementing an online voting system for a community event and wants to ensure that only eligible voters (those aged 18 or older) can participate.

Your task is to develop a program that validates the age of individuals attempting to vote online. If the user's age is below 18, the program should throw a custom exception, InvalidAgeException, preventing them from casting their vote. If the input is invalid, catch the appropriate InputMismatchException and print the in-built exception message.

Input Format

The input consists of an integer representing the age.

Output Format

If the age is 18 or older, print "Eligible to vote"

If the age is below 18, print "Exception occurred: InvalidAgeException: Age is not valid to vote"

If there is any other type of exception, print "An error occurred: " followed by the in-built exception message.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20
Output: Eligible to vote

Answer

```
import java.util.InputMismatchException;
import java.util.*;

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

class main {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            int age = sc.nextInt();

            if (age < 18) {
                throw new InvalidAgeException("Age is not valid to vote");
            }
            else {
                System.out.println("Eligible to vote");
            }
        }
        catch (InputMismatchException e) {
            System.out.println("An error occurred: " + e.getClass().getName());
```

```
        }
        catch (InvalidAgeException e) {
            System.out.println("Exception occurred: " + e.getClass().getName() + ": " +
e.getMessage());
        }
        catch (Exception e) {
            System.out.println("An error occurred: " + e.getMessage());
        }
        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: Nishant Gurung
Email: 241501131@rajalakshmi.edu.in
Roll no: 241501131
Phone: 8670069246
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a file management system, users are required to provide a valid file name when creating new files. The system enforces specific rules for file names to maintain consistency and avoid potential issues. Your task is to implement a Java program named FileNameValidator that takes user input for a file name and validates it according to the specified rules.

Rules for Valid File Name:

The file name must consist of alphanumeric characters (letters and digits) only.The file name must have a minimum length of 3 characters.

Implement a custom exception, FileNameValidator, to handle cases where the entered filename does not meet the specified criteria.

Input Format

The input consists of a string S, representing the desired filename.

Output Format

The output is displayed in the following format:

If the entered file name meets the specified criteria, the program outputs

"Valid file name"

If the entered file name does not meet the criteria and triggers the InvalidFileNameException, the program outputs

"Error: Invalid file name. It must be alphanumeric and have a minimum length of 3 characters."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: myfile123

Output: Valid file name

Answer

```
import java.util.Scanner;

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

class FileNameValidator {
    public static void fileValidator(String fileName) throws
InvalidFileNameException {
        if (fileName.length() < 3 || !fileName.matches("[A-Za-z0-9]+")) {
            throw new InvalidFileNameException(
                "Error: Invalid file name. It must be alphanumeric and have a minimum
length of 3 characters."
            );
        }
```

```
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String fileName = sc.nextLine();

        try {
            fileValidator(fileName);
            System.out.println("Valid file name");
        }

        catch (InvalidFileNameException e) {
            System.out.println(e.getMessage());
        }

        sc.close();
    }
}
```

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