SOLUTION\_1

import java.util.Scanner;  
  
// Define the custom exceptions  
  
class AgeNotWithinRangeException extends Exception {  
 public AgeNotWithinRangeException(String message) {  
 super(message);  
 }  
}  
  
class NameNotValidException extends Exception {  
 public NameNotValidException(String message) {  
 super(message);  
 }  
}  
  
// Define the Student class  
class Student {  
 private int rollNo;  
 private String name;  
 private int age;  
 private String course;  
  
 // Constructor to initialize student attributes  
 public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, NameNotValidException {  
 this.rollNo = rollNo;  
 this.name = name;  
 this.age = age;  
 this.course = course;  
  
 // Validate age and name  
 validateAge();  
 validateName();  
 }  
  
 // Method to validate age  
 private void validateAge() throws AgeNotWithinRangeException {  
 if (age < 15 || age > 21) {  
 throw new AgeNotWithinRangeException("AgeNot WithinRangeExceptio " + age + ". Age must be between 15 and 21.");  
 }  
 }  
  
 // Method to validate name  
 private void validateName() throws NameNotValidException {  
 if (!name.matches("[a-zA-Z]+")) {  
 throw new NameNotValidException("Invalid name '" + name + "'. NameNot ValidException.");  
 }  
 }  
  
 // Method to display student details  
 public String toString() {  
 return "Student [Roll No: " + rollNo + ", Name: " + name + ", Age: " + age + ", Course: " + course + "]";  
 }  
}

import java.util.Scanner;  
  
public class StudentManagementSystem {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Loop to allow multiple student entries  
 while (true) {  
 try {  
 // Get input from user  
 System.*out*.print("Enter roll number: ");  
 int rollNo = Integer.*parseInt*(scanner.nextLine());  
  
 System.*out*.print("Enter student name: ");  
 String name = scanner.nextLine();  
  
 System.*out*.print("Enter age: ");  
 int age = Integer.*parseInt*(scanner.nextLine());  
  
 System.*out*.print("Enter course: ");  
 String course = scanner.nextLine();  
  
 // Create a student object and validate  
 Student student = new Student(rollNo, name, age, course);  
  
 // Print student details if valid  
 System.*out*.println(student);  
  
 } catch (NumberFormatException e) {  
 System.*out*.println("NameNot ValidException");  
 } catch (AgeNotWithinRangeException e) {  
 System.*out*.println(e.getMessage());  
 } catch (NameNotValidException e) {  
 System.*out*.println(e.getMessage());  
 }  
  
 // Ask if the user wants to add another student  
 System.*out*.print("Do you want to add another student? (yes/no): ");  
 String response = scanner.nextLine();  
 if (!response.equalsIgnoreCase("yes")) {  
 break;  
 }  
 }  
  
 scanner.close();  
 }  
}

OUTPUT

Enter roll number: 01

Enter student name: SEDHU

Enter age: 22

Enter course: IT

AgeNot WithinRangeExceptio 22. Age must be between 15 and 21.

Do you want to add another student? (yes/no): YES

Enter roll number: 02

Enter student name: $EDHU

Enter age: 18

Enter course: IT

Invalid name '$EDHU'. NameNot ValidException.

Do you want to add another student? (yes/no): YES

Enter roll number: 03

Enter student name: SEDHU

Enter age: 18

Enter course: IT

Student [Roll No: 3, Name: SEDHU, Age: 18, Course: IT]

Do you want to add another student? (yes/no):

SOLUTION\_2

import java.util.Scanner;  
  
// Custom checked exception class for invalid age  
class InvalidAgeException extends Exception {  
 public InvalidAgeException(String message) {  
 super(message);  
 }  
}  
  
public class Voter {  
 private String voterId;  
 private String name;  
 private int age;  
  
 // Parameterized constructor  
 public Voter(String voterId, String name, int age) throws InvalidAgeException {  
 if (age < 18) {  
 throw new InvalidAgeException("invalid age for voter");  
 }  
 this.voterId = voterId;  
 this.name = name;  
 this.age = age;  
 }  
  
 // Getter methods  
 public String getVoterId() {  
 return voterId;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 // Main method to test the Voter class  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Prompt user for input  
 System.*out*.println("Enter Voter ID: ");  
 String voterId = scanner.nextLine();  
  
 System.*out*.println("Enter Name: ");  
 String name = scanner.nextLine();  
  
 System.*out*.println("Enter Age: ");  
 int age = scanner.nextInt();  
  
 try {  
 // Attempt to create a Voter object  
 Voter voter = new Voter(voterId, name, age);  
 // Print details of the valid voter  
 System.*out*.println("Voter created successfully!");  
 System.*out*.println("Voter ID: " + voter.getVoterId());  
 System.*out*.println("Name: " + voter.getName());  
 System.*out*.println("Age: " + voter.getAge());  
 } catch (InvalidAgeException e) {  
 // Catch the exception and print the error message  
 System.*out*.println(e.getMessage());  
 } finally {  
 // Close the scanner to prevent resource leak  
 scanner.close();  
 }  
 }  
}

OUTPUT

Enter the day position (0-6):

0

The day is: Sunday

SOLUTION\_3

import java.util.Scanner;  
  
public class Weekday {  
 public static void main(String[] args) {  
 // Array to store names of weekdays, starting from Sunday at index 0  
 String[] daysOfWeek = {  
 "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"  
 };  
  
 // Create a scanner object to get input from the user  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Ask the user to enter the day position (0-6)  
 System.*out*.println("Enter the day position (0-6): ");  
 int dayPosition = scanner.nextInt();  
  
 try {  
 // Print the corresponding day name from the array  
 System.*out*.println("The day is: " + daysOfWeek[dayPosition]);  
 } catch (ArrayIndexOutOfBoundsException e) {  
 // Handle invalid index by catching the exception  
 System.*out*.println("Invalid day index. Please enter a number between 0 and 6.");  
 } finally {  
 // Close the scanner to prevent resource leak  
 scanner.close();  
 }  
 }  
}

OUTPUT

Enter the day position (0-6):

5

The day is: Friday

SOLUTION\_4

import java.util.HashMap;

import java.util.Scanner;

public class StudentGrades {

// Create a HashMap to store student names and their grades

private HashMap<String, Integer> studentGrades = new HashMap<>();

// Method to add a new student with their grade

public void addStudent(String name, int grade) {

studentGrades.put(name, grade);

System.out.println(name + " has been added with grade: " + grade);

}

// Method to remove a student by name

public void removeStudent(String name) {

if (studentGrades.containsKey(name)) {

studentGrades.remove(name);

System.out.println(name + " has been removed.");

} else {

System.out.println("Student " + name + " not found.");

}

}

// Method to display a student's grade by name

public void displayGrade(String name) {

if (studentGrades.containsKey(name)) {

System.out.println(name + "'s grade is: " + studentGrades.get(name));

} else {

System.out.println("Student " + name + " not found.");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

StudentGrades studentGradesObj = new StudentGrades();

// Menu to perform operations

while (true) {

System.out.println("\nChoose an option:");

System.out.println("1. Add a new student");

System.out.println("2. Remove a student");

System.out.println("3. Display student's grade");

System.out.println("4. Exit");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character left by nextInt()

switch (choice) {

case 1:

System.out.print("Enter student's name: ");

String nameToAdd = scanner.nextLine();

System.out.print("Enter grade for " + nameToAdd + ": ");

int gradeToAdd = scanner.nextInt();

studentGradesObj.addStudent(nameToAdd, gradeToAdd);

break;

case 2:

System.out.print("Enter student's name to remove: ");

String nameToRemove = scanner.nextLine();

studentGradesObj.removeStudent(nameToRemove);

break;

case 3:

System.out.print("Enter student's name to display grade: ");

String nameToDisplay = scanner.nextLine();

studentGradesObj.displayGrade(nameToDisplay);

break;

case 4:

System.out.println("Exiting program.");

scanner.close();

return;

default:

System.out.println("Invalid option, please try again.");

}

}

}

}

OUTPUT

Choose an option:

1. Add a new student

2. Remove a student

3. Display student's grade

4. Exit

1

Enter student's name: SEDHU

Enter grade for SEDHU: 90

SEDHU has been added with grade: 90

Choose an option:

1. Add a new student

2. Remove a student

3. Display student's grade

4. Exit

3

Enter student's name to display grade: SEDHU

SEDHU's grade is: 90

Choose an option:

1. Add a new student

2. Remove a student

3. Display student's grade

4. Exit

2

SEDHU

Enter student's name to remove: SEDHU has been removed.

Choose an option:

1. Add a new student

2. Remove a student

3. Display student's grade

4. Exit