1.

import java.util.Scanner;

class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String message) {

super(message);

}

}

class NameNotValidException extends Exception {

public NameNotValidException(String message) {

super(message);

}

}

class Student {

int rollno;

String name;

int age;

String course;

Student(int rollno, String name, int age, String course) throws AgeNotWithinRangeException, NameNotValidException {

if (age < 15 || age > 21) {

throw new AgeNotWithinRangeException("Age should be between 15 and 21");

}

if (!name.matches("[a-zA-Z\\s]+")) {

throw new NameNotValidException("Name should not contain numbers or special characters");

}

this.rollno = rollno;

this.name = name;

this.age = age;

this.course = course;

}

public void display() {

System.out.println("Student Details:");

System.out.println("Roll No: " + rollno);

System.out.println("Name: " + name);

System.out.println("Age: " + age);

System.out.println("Course: " + course);

}

}

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

try {

System.out.print("Enter Roll No: ");

int roll = sc.nextInt();

sc.nextLine();

System.out.print("Enter Name: ");

String name = sc.nextLine();

System.out.print("Enter Age: ");

int age = sc.nextInt();

sc.nextLine();

System.out.print("Enter Course: ");

String course = sc.nextLine();

Student s = new Student(roll, name, age, course);

s.display();

} catch (AgeNotWithinRangeException | NameNotValidException e) {

System.out.println("Error: " + e.getMessage());

}

}

}

2.Voter Problem

import java.util.Scanner;

// Custom Checked Exception

class InvalidVoterAgeException extends Exception {

public InvalidVoterAgeException(String message) {

super(message);

}

}

// Voter class with checked exception

class Voter {

int age;

public Voter( int age) throws InvalidVoterAgeException {

if (age < 18) {

throw new InvalidVoterAgeException("invalid age for voter");

}

this.age = age;

}

public void display() {

System.out.println("Age: " + age);

}

}

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

try {

System.out.print("Enter Age: ");

int age = sc.nextInt();

Voter v = new Voter( age);

v.display();

} catch (InvalidVoterAgeException e) {

System.out.println("Exception: " + e.getMessage());

}

}

}

3.Week Day Problem

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String[] days = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

System.out.print("Enter the position of the day (0-6): ");

int index = sc.nextInt();

try {

System.out.println("Day is: " + days[index]);

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Invalid input! Please enter a value between 0 and 6.");

}

}

}

4.HashMap

import java.util.\*;

class StudentGrades {

private HashMap<String, Integer> studentGrades;

public StudentGrades() {

studentGrades = new HashMap<>();

}

public void addStudent(String name, int grade) {

studentGrades.put(name, grade);

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

}

public void removeStudent(String name) {

if (studentGrades.containsKey(name)) {

studentGrades.remove(name);

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

} else {

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

}

}

public void displayGrade(String name) {

if (studentGrades.containsKey(name)) {

System.out.println("Grade of " + name + ": " + studentGrades.get(name));

} else {

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

}

}

}

public class Main {

public static void main(String[] args) {

StudentGrades studentGrades = new StudentGrades();

studentGrades.addStudent("Alice", 90);

studentGrades.addStudent("Bob", 85);

studentGrades.addStudent("Charlie", 88);

studentGrades.displayGrade("Alice");

studentGrades.displayGrade("Bob");

studentGrades.removeStudent("Bob");

}

}

5.Stack

import java.util.\*;

class StackOperations {

private Stack<Integer> stack;

public StackOperations() {

stack = new Stack<>();

}

public void pushElement(int element) {

stack.push(element);

System.out.println("Pushed element: " + element);

}

public void popElement() {

if (!stack.isEmpty()) {

int poppedElement = stack.pop();

System.out.println("Popped element: " + poppedElement);

} else {

System.out.println("Stack is empty. Cannot pop.");

}

}

public void checkIfEmpty() {

if (stack.isEmpty()) {

System.out.println("The stack is empty.");

} else {

System.out.println("The stack is not empty.");

}

}

}

public class Main {

public static void main(String[] args) {

StackOperations stackOps = new StackOperations();

stackOps.pushElement(10);

stackOps.pushElement(20);

stackOps.pushElement(30);

stackOps.checkIfEmpty();

stackOps.popElement();

stackOps.popElement();

stackOps.checkIfEmpty();

stackOps.popElement();

stackOps.checkIfEmpty();

}

}