1.Bill using interface

// Interface with calculate method

interface Billable {

    double calculate(); // to calculate total amount

}

// Product class implementing the interface

class Product implements Billable {

    private int productId;

    private String name;

    private int quantity;

    private double unitPrice;

    public Product(int id, String productName, int qty, double price) {

        productId = id;

        name = productName;

        quantity = qty;

        unitPrice = price;

    }

    // Implementation of calculate method from interface

    public double calculate() {

        return quantity \* unitPrice;

    }

    // Method to display product details

    public void displayProduct() {

        System.out.printf("%-10d%-10s%-10d%-15.2f%-10.2f%n",

                          productId, name, quantity, unitPrice, calculate());

    }

}

// Main class

public class BillWithInterface {

    public static void main(String[] args) {

        Product product1 = new Product(101, "A", 2, 25.0);

        Product product2 = new Product(102, "B", 1, 100.0);

        double netAmount = product1.calculate() + product2.calculate();

        // Display the bill

        System.out.println("Order No.   Date :");

        System.out.println("Product Id  Name      Quantity   Unit Price     Total");

        product1.displayProduct();

        product2.displayProduct();

        System.out.printf("%-50s%-10.2f%n", "Net. Amount", netAmount);

    }

}

2.authentication

import java.util.Scanner;

// Custom exception for username error

class UsernameException extends Exception {

    public UsernameException(String msg) {

        super(msg);

    }

}

// Custom exception for password error

class PasswordException extends Exception {

    public PasswordException(String msg) {

        super(msg);

    }

}

public class DemoExcp {

    public static void main(String args[]) {

        String name = "nidhi";

        String pass = "nidhips";

        Scanner s = new Scanner(System.in);

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

        String user = s.nextLine();

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

        String pwd = s.nextLine();

        try {

            // Checking username and password validity

            if (!user.equals(name)) {

                throw new UsernameException("Username incorrect");

            } else if (!pwd.equals(pass)) {

                throw new PasswordException("Password incorrect");

            } else {

                System.out.println("Login Successful!!!");

            }

        }

        catch (UsernameException u) {

            u.printStackTrace();

        }

        catch (PasswordException p) {

            p.printStackTrace();

        }

        finally {

            s.close(); // Close the scanner resource

        }

    }

}

3.

import java.util.Scanner;

// Custom exception for negative input

class NegativeInputException extends Exception {

    public NegativeInputException() {

        super("Negative input is not allowed.");

    }

}

public class Average {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the value of N: ");

        int N = scanner.nextInt();  // Total number of inputs expected

        int sum = 0;

        int count = 0;  // Count of valid numbers entered

        // Loop to take N inputs

        for (int i = 1; i <= N; i++) {

            try {

                System.out.print("Enter an integer: ");

                int num = scanner.nextInt();

                // Check if the number is negative

                if (num < 0) {

                    throw new NegativeInputException();

                }

                sum += num;  // Add the valid number to the sum

                count++;  // Increment count of valid numbers

            } catch (NegativeInputException e) {

                System.out.println(e.getMessage());  // Show the error message

                i--;  // Do not count this iteration towards N

            }

        }

        // Calculate and display the average if count > 0 (to avoid division by zero)

        if (count > 0) {

            double average = (double) sum / count;  // Corrected average calculation

            System.out.println("The Average of the valid numbers is: " + average);

        } else {

            System.out.println("No valid numbers entered, cannot calculate average.");

        }

        scanner.close();  // Close the scanner resource

    }

}

4. import java.util.LinkedList;

public class RemoveAll {

public static void main(String[] args) {

// Create a LinkedList

LinkedList<String> linkedList = new LinkedList<>();

// Add elements to the LinkedList

linkedList.add("Apple");

linkedList.add("Banana");

linkedList.add("Cherry");

linkedList.add("Date");

// Display the initial LinkedList

System.out.println("Initial LinkedList: " + linkedList);

// Remove all elements from the LinkedList

linkedList.clear();

// Display the LinkedList after removal

System.out.println("LinkedList after removing all elements: " + linkedList);

}

}