class Student

{

int rollno;

string name;

static String college="ACE";

Student(int r, String n)

{

rollno=r;

name=n;

}

static void hello()

{

System.out.println("Welcome to Adhiyamaan college");

}

void display()

{

System.out.println(rollno+"."+name+"."+college);

}

public static void main(String args[])

{

Student s1=new Student(10,"Arun");

Student s2=new Student(20,"Kumar");

s1.display();

s2.display();

hello();

}

}

10.Arun.ACE

20.Kumar.ACE

Welcome to Adhiyamaan college

class Employee {

int id; // Remove 'static' so each Employee has its own id

String name;

String address;

String phno;

void emp() { // 'emp' method should not be static if it's instance-specific

System.out.println("Employee ID: " + id + "\n");

}

public static void main(String args[]) {

Employee emp1 = new Employee();

emp1.id = 1689;

emp1.phno = "9999988888";

emp1.address = "Hosur";

emp1.name = "John";

emp1.emp();

System.out.println("Employee Name: " + emp1.name + "\n");

System.out.println("Employee Phone Number: " + emp1.phno + "\n");

System.out.println("Employee Address: " + emp1.address + "\n");

}

}

Employee ID: 1689

Employee Name: John

Employee Phone Number: 9999988888

Employee Address: Hosur

import java.util.Scanner;

class Area {

int length;

int breadth;

int height;

int base;

int radius;

void rectangle() {

System.out.println("Area of Rectangle: " + (length \* breadth));

}

void triangle() {

System.out.println("Area of Triangle: " + (0.5 \* base \* height));

}

void circle() {

System.out.println("Area of Circle: " + (3.142 \* radius \* radius));

}

static class Output extends Area {

void start() {

Scanner input = new Scanner(System.in);

System.out.println("Find area:\n1. Rectangle\n2. Triangle\n3. Circle");

int shape = input.nextInt();

switch (shape) {

case 1:

System.out.println("Enter length:");

int l = input.nextInt();

this.length = l;

System.out.println("Enter breadth:");

int b = input.nextInt();

this.breadth = b;

this.rectangle();

break;

case 2:

System.out.println("Enter base:");

int ba = input.nextInt();

this.base = ba;

System.out.println("Enter height:");

int h = input.nextInt();

this.height = h;

this.triangle();

break;

case 3:

System.out.println("Enter radius:");

int r = input.nextInt();

this.radius = r;

this.circle();

break;

default:

System.out.println("Invalid input");

break;

}

input.close();

}

}

public static void main(String args[]) {

Output op = new Output();

op.start();

}

}

Find area:

1. Rectangle

2. Triangle

3. Circle

3

Enter radius:

7

Area of Circle: 153.958

package com.example;

public class MyPackageExample {

public void display() {

System.out.println("This is a class inside the com.example package.");

}

}

package com.example;

public interface MyInterface {

void myMethod();

}

package com.example;

public class MyImplementation implements MyInterface {

@Override

public void myMethod() {

System.out.println("Implementation of the myMethod() from the interface.");

}

}

package com.example;

public class Main {

public static void main(String[] args) {

MyPackageExample packageExample = new MyPackageExample();

packageExample.display();

MyImplementation implementation = new MyImplementation();

implementation.myMethod();

}

}

This is a class inside the com.example package.

Implementation of the myMethod() from the interface.

import java.util.Scanner;

class Student {

int roll\_no;

String student\_name;

String branch;

String department = "Information technology";

static String year = "2022-2026";

String phno;

String college;

String city;

// Constructor overloading

Student() {

Scanner scan = new Scanner(System.in);

System.out.print("Enter the roll\_no: ");

roll\_no = scan.nextInt();

System.out.println("THE INFORMATION OF THE STUDENT ......");

System.out.println("The roll no of the student: " + roll\_no);

scan.close();

}

Student(String name, String branch\_name) {

student\_name = name;

branch = branch\_name;

}

// Method overloading

public void display(String phno) {

System.out.println("Mobile no: " + phno);

}

public void display(String college, String city) {

System.out.println("College name: " + college);

System.out.println("Name of the city: " + city);

}

public static void main(String[] args) {

Student s1 = new Student();

Student s = new Student("Sona", "B-Tech");

System.out.println("The name of the student is: " + s.student\_name);

s.display("7010588669");

System.out.println("The branch of the student is: " + s.branch);

System.out.println("The name of the department: " + s1.department);

System.out.println("Year: " + year);

s.display("Adhiyamaan", "Hosur");

}

}

Enter the roll\_no: 101

THE INFORMATION OF THE STUDENT ......

The roll no of the student: 101

The name of the student is: Sona

Mobile no: 7010588669

The branch of the student is: B-Tech

The name of the department: Information technology

Year: 2022-2026

College name: Adhiyamaan

Name of the city: Hosur

// Custom exception class

class CustomException extends Exception {

public CustomException(String message) {

super(message);

}

}

// Sample class with a method that throws a custom exception

class Example {

public void performOperation(int value) throws CustomException {

if (value < 0) {

throw new CustomException("Value cannot be negative");

} else {

System.out.println("Operation performed successfully with value: " + value);

}

}

}

public class ExceptionHandlingDemo {

public static void main(String[] args) {

try {

Example example = new Example();

int inputValue = -3; // Test value that triggers the exception

example.performOperation(inputValue);

} catch (CustomException ce) {

System.err.println("Custom Exception Caught: " + ce.getMessage());

} finally {

System.out.println("Exception handling example completed.");

}

}

}

Custom Exception Caught: Value cannot be negative

Exception handling example completed.

// Java code for thread creation by extending the Thread class

class MultithreadingDemo extends Thread {

public void run() {

try {

// Displaying the thread that is running

System.out.println("Thread " + Thread.currentThread().getId() + " is running");

} catch (Exception e) {

// Catching any exception that occurs

System.out.println("Exception is caught");

}

}

}

// Main Class

public class Multithread {

public static void main(String[] args) {

int n = 8; // Number of threads

for (int i = 0; i < n; i++) {

MultithreadingDemo object = new MultithreadingDemo();

object.start();

}

}

}

Thread 12 is running

Thread 13 is running

Thread 14 is running

Thread 15 is running

Thread 16 is running

Thread 17 is running

Thread 18 is running

Thread 19 is running

import java.util.ArrayList;

import java.util.HashMap;

import java.util.HashSet;

import java.util.Map;

import java.util.Set;

public class CollectionExample {

public static void main(String[] args) {

// Array List Example

ArrayList<String> arrayList = new ArrayList<>();

arrayList.add("Apple");

arrayList.add("Banana");

arrayList.add("Orange");

// Map Example

Map<String, Integer> map = new HashMap<>();

map.put("One", 1);

map.put("Two", 2);

map.put("Three", 3);

// Set Example

Set<String> set = new HashSet<>();

set.add("Red");

set.add("Green");

set.add("Blue");

// Output Array List elements

System.out.println("Array List Elements:");

for (String fruit : arrayList) {

System.out.println(fruit);

}

// Output Map elements

System.out.println("\nMap Elements:");

for (Map.Entry<String, Integer> entry : map.entrySet()) {

System.out.println(entry.getKey() + ": " + entry.getValue());

}

// Output Set elements

System.out.println("\nSet Elements:");

for (String color : set) {

System.out.println(color);

}

}

}

Array List Elements:

Apple

Banana

Orange

Map Elements:

One: 1

Two: 2

Three: 3

Set Elements:

Red

Green

Blue

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

class FileReaderExample {

public static void main(String[] args) {

try {

File f = new File("filename.txt"); // Make sure this file exists in the correct path

Scanner read = new Scanner(f);

// Reading file line by line

while (read.hasNextLine()) {

System.out.println(read.nextLine());

}

read.close(); // Closing the scanner to release resources

} catch (FileNotFoundException exception) {

// Handle exception if file is not found

System.out.println("ERROR: File not found.");

}

}

}

Hello, this is the content of the file.

This is the second line of the file.

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

class FileWriterExample {

public static void main(String[] args) {

try {

// Creating a new file object

File f = new File("filename.txt");

// Creating a FileWriter object to write to the file

FileWriter fw = new FileWriter(f);

// Writing to the file

fw.write("HELLO JAVA CODE");

// Closing the FileWriter object to save changes

fw.close();

System.out.println("Data has been written to the file successfully.");

} catch (IOException e) {

// Handling IOException if file cannot be written

System.out.println("ERROR: An error occurred while writing to the file.");

}

}

}

Data has been written to the file successfully.

public class StringHandlingExample {

public static void main(String[] args) {

// Sample input string

String text = "Hello, World! This is a Java String Handling Example.";

// 1. Length of the String

int length = text.length();

System.out.println("1. Length of the String: " + length);

// 2. Character at a specific index

char characterAtIndex = text.charAt(7); // Index 7 is 'W'

System.out.println("2. Character at index 7: " + characterAtIndex);

// 3. Substring from index 13 to 22

String substring = text.substring(13, 23); // "This is a"

System.out.println("3. Substring from index 13 to 22: " + substring);

// 4. Concatenation

String string1 = "Hello, ";

String string2 = "World!";

String concatenatedString = string1 + string2;

System.out.println("4. Concatenation: " + concatenatedString);

// 5. String comparison

String str1 = "apple";

String str2 = "banana";

int comparisonResult = str1.compareTo(str2);

System.out.println("5. String Comparison (Lexicographically): " + comparisonResult);

// 6. String to Upper Case

String upperCaseText = text.toUpperCase();

System.out.println("6. Uppercase: " + upperCaseText);

// 7. String to Lower Case

String lowerCaseText = text.toLowerCase();

System.out.println("7. Lowercase: " + lowerCaseText);

// 8. Checking if a string contains a substring

boolean containsSubstring = text.contains("Java");

System.out.println("8. Contains 'Java': " + containsSubstring);

// 9. Replacing a substring

String replacedText = text.replace("World", "Universe");

System.out.println("9. Replace 'World' with 'Universe': " + replacedText);

// 10. Splitting a string into an array

String[] words = text.split(" ");

System.out.println("10. Splitting into words (space-separated):");

for (String word : words) {

System.out.println(word);

}

}

}

1. Length of the String: 54

2. Character at index 7: W

3. Substring from index 13 to 22: This is a

4. Concatenation: Hello, World!

5. String Comparison (Lexicographically): -1

6. Uppercase: HELLO, WORLD! THIS IS A JAVA STRING HANDLING EXAMPLE.

7. Lowercase: hello, world! this is a java string handling example.

8. Contains 'Java': true

9. Replace 'World' with 'Universe': Hello, Universe! This is a Java String Handling Example.

10. Splitting into words (space-separated):

Hello,

World!

This

is

a

Java

String

Handling

Example.

package jdbcswing;

import java.awt.Container;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.BoxLayout;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class JDBCSwing implements ActionListener {

// Declare the components

JLabel lblFName, lblLName, lblAddress, lblSalary, lblFVal, lblLVal, lblAVal, lblSVal;

JTextField txtFName, txtLName, txtAddress, txtSalary;

JButton btnAdd, btnUpdate, btnDelete, btnPrev, btnNext;

ResultSet rs;

public static void main(String[] args) {

JDBCSwing obj = new JDBCSwing();

obj.createUI();

}

private void createUI() {

JFrame frame = new JFrame("JDBC All in One");

JPanel pnlInput = new JPanel(new GridLayout(4, 2));

lblFName = new JLabel("First Name: ");

txtFName = new JTextField(15);

lblLName = new JLabel("Last Name: ");

txtLName = new JTextField();

lblAddress = new JLabel("Address: ");

txtAddress = new JTextField();

lblSalary = new JLabel("Salary: ");

txtSalary = new JTextField();

pnlInput.add(lblFName);

pnlInput.add(txtFName);

pnlInput.add(lblLName);

pnlInput.add(txtLName);

pnlInput.add(lblAddress);

pnlInput.add(txtAddress);

pnlInput.add(lblSalary);

pnlInput.add(txtSalary);

JPanel pnlButton = new JPanel(new GridLayout(1, 3));

btnAdd = new JButton("Add");

btnAdd.addActionListener(this);

btnUpdate = new JButton("Update");

btnUpdate.addActionListener(this);

btnDelete = new JButton("Delete");

btnDelete.addActionListener(this);

pnlButton.add(btnAdd);

pnlButton.add(btnUpdate);

pnlButton.add(btnDelete);

JPanel pnlNavigate = new JPanel(new GridLayout(1, 2));

btnPrev = new JButton(" << ");

btnPrev.setActionCommand("Prev");

btnPrev.addActionListener(this);

btnNext = new JButton(" >> ");

btnNext.setActionCommand("Next");

btnNext.addActionListener(this);

pnlNavigate.add(btnPrev);

pnlNavigate.add(btnNext);

JPanel pnlNavAns = new JPanel(new GridLayout(4, 2));

lblFVal = new JLabel("First Name: ");

lblLVal = new JLabel("Last Name: ");

lblAVal = new JLabel("Address: ");

lblSVal = new JLabel("Salary: ");

pnlNavAns.add(lblFName);

pnlNavAns.add(lblFVal);

pnlNavAns.add(lblLName);

pnlNavAns.add(lblLVal);

pnlNavAns.add(lblAddress);

pnlNavAns.add(lblAVal);

pnlNavAns.add(lblSalary);

pnlNavAns.add(lblSVal);

Container cn = frame.getContentPane();

cn.setLayout(new BoxLayout(cn, BoxLayout.Y\_AXIS));

frame.add(pnlInput);

frame.add(pnlButton);

frame.add(pnlNavAns);

frame.add(pnlNavigate);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.pack();

frame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent evt) {

String action = evt.getActionCommand();

if (action.equals("Add")) {

addOperation();

} else if (action.equals("Update")) {

updateOperation();

} else if (action.equals("Delete")) {

deleteOperation();

} else if (action.equals("Prev")) {

preNavigation();

} else if (action.equals("Next")) {

nextNavigation();

}

}

private void addOperation() {

try {

Class.forName("oracle.jdbc.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "SYSTEM", "SYSTEM");

String sql = "INSERT INTO Employee1 (FName, LName, Address, Salary) " +

"VALUES ('" + txtFName.getText() + "','" + txtLName.getText() + "','" +

txtAddress.getText() + "','" + txtSalary.getText() + "')";

Statement st = con.createStatement();

st.executeUpdate(sql);

JOptionPane.showMessageDialog(null, "Record Added Successfully.", "Record Added", JOptionPane.INFORMATION\_MESSAGE);

clearControls();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void updateOperation() {

try {

Class.forName("oracle.jdbc.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "SYSTEM", "SYSTEM");

String sql = "UPDATE Employee1 SET LName = '" + txtLName.getText() + "', Address = '" +

txtAddress.getText() + "', Salary = '" + txtSalary.getText() + "' WHERE FName = '" + txtFName.getText() + "'";

Statement st = con.createStatement();

st.executeUpdate(sql);

JOptionPane.showMessageDialog(null, "Record Updated Successfully.", "Record Updated", JOptionPane.INFORMATION\_MESSAGE);

clearControls();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void deleteOperation() {

int ans = JOptionPane.showConfirmDialog(null, "Are you sure to delete the Record?", "Delete Record", JOptionPane.YES\_NO\_OPTION);

if (ans == JOptionPane.YES\_OPTION) {

try {

Class.forName("oracle.jdbc.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "SYSTEM", "SYSTEM");

String sql = "DELETE FROM Employee1 WHERE FName = '" + txtFName.getText() + "'";

Statement st = con.createStatement();

st.executeUpdate(sql);

JOptionPane.showMessageDialog(null, "Record Deleted Successfully", "Success", JOptionPane.INFORMATION\_MESSAGE);

clearControls();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null, "Operation Canceled", "Cancel", JOptionPane.INFORMATION\_MESSAGE);

}

}

private void preNavigation() {

try {

if (rs == null) {

Class.forName("oracle.jdbc.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "SYSTEM", "SYSTEM");

String sql = "SELECT \* FROM Employee1";

Statement st = con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_UPDATABLE);

rs = st.executeQuery(sql);

}

if (rs.previous()) {

populateValue();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void nextNavigation() {

try {

if (rs == null) {

Class.forName("oracle.jdbc.OracleDriver");

Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE", "SYSTEM", "SYSTEM");

String sql = "SELECT \* FROM Employee1";

Statement st = con.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_UPDATABLE);

rs = st.executeQuery(sql);

}

if (rs.next()) {

populateValue();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void populateValue() throws Exception {

String fName = rs.getString("FName");

String lName = rs.getString("LName");

String address = rs.getString("Address");

String salary = rs.getString("Salary");

lblFVal.setText(fName);

lblLVal.setText(lName);

lblAVal.setText(address);

lblSVal.setText(salary);

txtFName.setText(fName);

txtLName.setText(lName);

txtAddress.setText(address);

txtSalary.setText(salary);

}

private void clearControls() {

txtFName.setText("");

txtLName.setText("");

txtAddress.setText("");

txtSalary.setText("");

}

}