Problem:- Creating ok & cancel button using event handling..

Corresponding code:-

Package Ok\_Cancel;

import javafx.application.Application;

import static javafx.application.Application.launch;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.HBox;

import javafx.stage.Stage;

public class Ok\_Cancel extends Application { @Override

public void start(Stage primaryStage) { HBox pane = new HBox(10);

pane.setAlignment(Pos.CENTER);

Button btOK = new Button("OK");

Button btCancel = new Button("Cancel");

OK handler1 = new OK();

btOK.setOnAction(handler1);

Cancel handler2 = new Cancel();

btCancel.setOnAction(handler2);

pane.getChildren().addAll(btOK, btCancel);

Scene scene = new Scene(pane);

primaryStage.setTitle("HandleEvent");

primaryStage.setScene(scene);

primaryStage.setHeight(500);

primaryStage.setWidth(1000);

primaryStage.show(); }

public static void main(String[] args) { launch(args); }

}

class OK implements EventHandler<ActionEvent> { @Override

public void handle(ActionEvent e) {

System.out.println("OK is clicked"); }

}

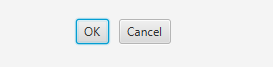
class Cancel implements EventHandler<ActionEvent> { @Override

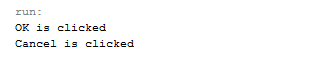
public void handle(ActionEvent e) {

System.out.println("Cancel is clicked"); }

}

Output:





Problem: Create a program by JavaFX..

Corresponding Code:-

package sample;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.FlowPane;

import javafx.scene.layout.Pane;

import javafx.stage.FileChooser;

import javafx.stage.Stage;

import java.awt.\*;

import java.io.\*;

import java.util.EventListenerProxy;

public class MyTestFX1 extends Application implements EventHandler<ActionEvent> {

private TextField textFieldFirstName = new TextField();

private Label labelFirstName = new Label("First Name:");

private Button button = new Button("Submit");

private Button fileChooserButton = new Button("Browse Files");

FileChooser fileChooser; @Override

public void start(Stage stage) throws Exception {

FlowPane flowPane = new FlowPane();

flowPane.setPadding(new Insets(11,12,13,14));

flowPane.setVgap(5);

flowPane.setHgap(5);

flowPane.getChildren().addAll(labelFirstName, textFieldFirstName, button , fileChooserButton);

button.setOnAction(this);

fileChooserButton.setOnAction(this);

Scene scene = new Scene(flowPane,500,250);

stage.setTitle("This is a Test");

stage.setScene(scene);

stage.show(); }

public static void main(String[] args)

{ Application.launch(); }

@Override

public void handle(ActionEvent actionEvent) {

if (actionEvent.getSource()==button) { System.out.println("Button Clicked!!\n The of Text Field:"+textFieldFirstName.getText());

ServiceClass serviceClass = new ServiceClass();

if(serviceClass.InsetDB(textFieldFirstName.getText()))

{ System.out.print("Database Successfully Updated!!"); }

else if (false)

{ System.out.print("Database not updated!"); }

try { FileWriter fileWriter = new FileWriter("files/topup.txt", true);

fileWriter.write(textFieldFirstName.getText()+"\n");

fileWriter.close();

} catch (IOException e) {

e.printStackTrace(); }

}

if (actionEvent.getSource().equals(fileChooserButton)) {

System.out.println("File Chooser Button!");

fileChooser = new FileChooser();

fileChooser.setTitle("Open Resource File");

String currentDir = System.getProperty("user.dir") + File.separator;

File file = new File(currentDir);

fileChooser.setInitialDirectory(file);

File selectedFile = fileChooser.showOpenDialog(new Stage());

if (selectedFile != null) { System.out.println(selectedFile); }

try {

Desktop.getDesktop().edit(new File(String.valueOf(selectedFile)));

} catch (IOException e) {

e.printStackTrace(); } } }

}

Output:



Problem: Create a program using arraylist.

Corresponding code: import java.util.\*;

class TestCollection1{

public static void main(String args[]){

ArrayList<String> list=new ArrayList<String>(); //Creating arraylist

list.add("Ravi"); //Adding object in arraylist

list.add("Vijay");

list.add("Ravi");

list.add("Ajay");

//Traversing list through Iterator

Iterator itr=list.iterator();

while(itr.hasNext()){

System.out.println(itr.next()); } } }

Output:



Problem:Create a program using linkedlist.

Corresponding Code:-

import java.util.\*;

public class LinkedList{

public static void main(String args[]) {

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

al.add("Ravi");

al.add("Vijay");

al.add("Ravi");

al.add("Ajay");

Iterator<String> itr=al.iterator();

while(itr.hasNext()){

System.out.println(itr.next()); } }

}

Output:



Problem: Create a program showing Exception handling..

Corresponding Code:

import java.util.Scanner;

public class Bank {

public static void main(String [] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Hi Welcome to our Bank!\nPlease Enter Your Account Number:");

String accountNumber = sc.next();

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

String password = sc.next();

CheckingAccount c = new CheckingAccount(accountNumber, password);

System.out.print("How much you want to Deposit:");

double amountDep = sc.nextDouble();

c.deposit(amountDep);

try { System.out.print("How much you want to withdraw:");

double amountW = sc.nextDouble();

c.withdraw(amountW);

System.out.print("How much you want to withdraw:");

amountW = sc.nextDouble();

c.withdraw(amountW);

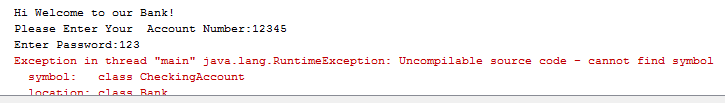
}catch(InsufficientFundsException e) {

System.out.println("Sorry, but you are short of " + e.getAmount()+" Taka");

//e.printStackTrace(); } }

}

Output:



Problem: Create a program for a web crawler using FileIO

Corresponding Code:

import java.util.Scanner;

import java.util.ArrayList;

public class WebCrawler {

public static void main(String[] args) {

java.util.Scanner input = new java.util.Scanner(System.in);

System.out.print("Enter a URL: ");

String url = input.nextLine();

crawler(url); // Traverse the Web from the a starting url

}

public static void crawler(String startingURL) {

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

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

listOfPendingURLs.add(startingURL);

while (!listOfPendingURLs.isEmpty() &&

listOfTraversedURLs.size() <= 100) {

String urlString = listOfPendingURLs.remove(0);

if (!listOfTraversedURLs.contains(urlString)) {

listOfTraversedURLs.add(urlString);

System.out.println("Craw " + urlString);

for (String s: getSubURLs(urlString)) {

if (!listOfTraversedURLs.contains(s))

listOfPendingURLs.add(s); } }

}

}

public static ArrayList<String> getSubURLs(String urlString) {

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

try {

java.net.URL url = new java.net.URL(urlString);

Scanner input = new Scanner(url.openStream());

int current = 0;

while (input.hasNext()) {

String line = input.nextLine();

current = line.indexOf("http:", current);

while (current > 0) {

int endIndex = line.indexOf("\"", current);

if (endIndex > 0) { // Ensure that a correct URL is found

list.add(line.substring(current, endIndex));

current = line.indexOf("http:", endIndex);

}

else

current = -1; }

}

}

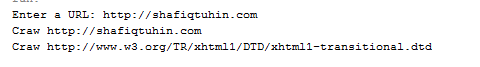
catch (Exception ex) {

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

} return list; }

}

Output:



Problem: Create a program for a digital clock using java swing…

Corresponding Code:-

import javax.swing.\*;

import java.awt.\*;

import java.text.\*;

import java.util.\*;

public class DigitalWatch implements Runnable{

JFrame f;

Thread t=null;

int hours=0, minutes=0, seconds=0;

String timeString = "";

JButton b;

DigitalWatch(){

f=new JFrame();

t = new Thread(this);

t.start();

b=new JButton();

b.setBounds(100,100,100,50);

f.add(b);

f.setSize(300,400);

f.setLayout(null);

f.setVisible(true);

}

public void run() {

try {

while (true) {

Calendar cal = Calendar.getInstance();

hours = cal.get( Calendar.HOUR\_OF\_DAY );

if ( hours > 12 ) hours -= 12;

minutes = cal.get( Calendar.MINUTE );

seconds = cal.get( Calendar.SECOND );

SimpleDateFormat formatter = new SimpleDateFormat("hh:mm:ss");

Date date = cal.getTime();

timeString = formatter.format( date );

printTime();

t.sleep( 1000 ); // interval given in milliseconds

} }

catch (Exception e) { }

}

public void printTime(){

b.setText(timeString); }

public static void main(String[] args) {

new DigitalWatch(); }

}

Output:

