

Aryam Srivastava

AD 73 / 70612400019

①

JAVA ASSIGNMENT - 2.

1) public class MyGenericClass < T > {
 private T obj;

public void add (T obj) {
 this.obj = obj;

public T get () {
 return obj;

public static void main (String [] args) {

MyGenericClass < String > stringObj =
 new MyGenericClass < > ();
 stringObj.add ("Hello");

MyGenericClass < Integer > intObj =
 new MyGenericClass < > ();
 intObj.add (20);

System.out.println (stringObj.get ());

System.out.println (intObj.get ());

2)

3)

OUTPUT:

Hello

World

2) public class MyArrayList<T,U> {

private T obj1;

private U obj2;

public MyArrayList(T obj1, U obj2) {

this.obj1 = obj1;

this.obj2 = obj2;

public void printObj() {

System.out.println("object1 : " + obj1);

System.out.println("object2 : " + obj2);

public static void main(String[] args) {

MyArrayList<String, Integer> obj1 =

new MyArrayList<T>("Text", 122);

obj1.printObj();

y

Output,

object 1: Test

object 2: 122

3) import java.util.*;

public class SetOperations {

public static void main (String [] args) {

Set < Integer > mySet = new TreeSet < >();

mySet.add(40);

mySet.add(5);

mySet.add(15);

System.out.println ("Set contains :" + mySet);

mySet.remove(10);

System.out.println ("Size of set :" + mySet.size());

int elementToCheck = 5;

if (mySet.contains(elementToCheck)) {

System.out.println ("1");

else

System.out.println ("-1");

OUTPUT;

Set contains : [5, 10, 15]

Size of set: 3

4) import java.util.*;

```
public class ArrayListOperations {  
    public static void main(String[] args) {  
        List<Integer> arrayList = new ArrayList<>();
```

```
        arrayList.add(10);  
        arrayList.add(20);  
        arrayList.add(5);
```

Collections.~~sort~~^{sort}(arrayList);

System.out.println("Sorted in array
 ascending Order: " + arrayList);

Collections.reverse(arrayList);

System.out.println("Reversed ArrayList: " +
arrayList);

System.out.println("Size of ArrayList: " +
arrayList.size());

```
for (int num : arraylist) {  
    System.out.println(num + " ");  
}
```

```
System.out.println();
```

```
arraylist.sort(Comparator.naturalOrder());  
System.out.println("Sorted in Descending  
order" + arraylist);
```

Output,

Sorted in Ascending Order: [5, 10, 20].

Revised ArrayList: [20, 10, 5]

Size of ArrayList: 3

Sorted in Descending Order (20, 10, 5)

5.) Phases of event handling in Java FX

(A) Event capturing phase,

 - The event is passed down from the root node of the target node.

 - Nodes have a chance to capture the event but usually pass it down until it reaches the target.

(B) Event Target Phase

↳ The event reaches the target node.

(C) Event Bubbling Phase

↳ After the event is processed by the target, it "bubbles up" from the target node back to the root node.

↳ Parent node may handle or consume the event during this phase.

6) JDBC Architecture

(A) JDBC API: This API defines this set of interfaces and classes that allow Java programs to interact with databases. It includes methods for:-

- Connecting to a database → Retrieving results
- Executing SQL queries → Managing transactions.

(B) JDBC Driver Manager: The driver manager handles communications between the JDBC API and the appropriate driver for the specific DB. It loads the correct DB driver dynamically.

↳ establishes a connection.

↳ four types of drivers;

→ JDBC-ODBC Bridge Driver

→ Native API driver

→ Network Protocol Driver

→ Thin Driver (pure java driver)

- (E) Steps for connecting database using JDBC,
- (A) Load the JDBC driver.
- use `Class.forName()` to load the database driver class.
- (B) Establish a connection.
- use `DriverManager.getConnection()` to establish a connection to the database.
- (C) Create a statement.
- use `Statement` or `PreparedStatement` to send SQL queries to the database.
- (D) Execute the query.
- use `executeQuery()` for select queries and `executeUpdate()` for insert, update & delete.
- (E) Process the result.
- iterate over the `ResultSet` to retrieve the query results.
- (F) Close the connection.
- Always close the `ResultSet`, `Statement`, & `Connection` objects to free up resources.
- 8)
- ```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
```

public class RegistrationForm extends Application

public static void main(String[] args) {

primaryStage = new Stage primaryStage.set

primaryStage.setTitle("Registration Form");

GridPane grid = new GridPane();

grid.setPadding(new Insets(10, 10, 10, 10));

grid.setVgap(8);

grid.setHgap(10);

Label nameLabel = new Label("Name:");

GridPane.setConstraints(nameLabel, 0, 0);

TextField nameInput = new TextField();

GridPane.setConstraints(nameInput, 1, 0);

Label emailLabel = new Label("Email:");

GridPane.setConstraints(emailLabel, 0, 1);

TextField emailInput = new TextField();

GridPane.setConstraints(emailInput, 1, 1);

Label phoneLabel = new Label("Phone:");

GridPane.setConstraints(phoneLabel, 0, 2);

TextField phoneInput = new TextField();

GridPane.setConstraints(phoneInput, 1, 2);

Label password = new Label ("Password:");  
andPan.setConstraints (passwordLabel, 0, 3);  
PasswordField pwinput = new PasswordField();  
andPan.setConstraints (pwinput, 1, 3);

Label confirm = new Label ("Confirm:");  
andPan.setConstraints (confirm, 0, 4);  
PasswordField confirmInput = new  
PasswordField ();  
andPan.setConstraints (confirmInput, 1, 4);

Label countryLabel = new Label ("Country:");  
andPan.setConstraints (countryLabel, 0, 5);  
WindowBox < String > countryInput = new  
WindowBox (>());

countryInput.getItems () .addAll ( "India",  
"USA", "UK" );

countryInput.setValue ("Select");

andPan.setConstraints (countryInput, 1, 5);

Label genderLabel = new Label ("Gender");  
andPan.setConstraints (genderLabel, 0, 6);  
ToggleGroup genderGroup = new ToggleGroup();  
RadioButtom maleRadio = new RadioButtom  
("Male");  
maleRadio.setToggleGroup (genderGroup);

RadioButton funleradio = new RadioButton ("Funeral");

funleradio .setToggle(true (under true));

UIBox genderbox = new UIBox (10, maleradio,

femaleradio;

cmdDone .setOnCommand (genderbox , 1, 6);

Label languageLabel = new Label ("Language");

cmdDone .setOnCommand (languageLabel , 0, 7);

UIBox arabiclick = new CheckBox ("Arabic");

checkbox englishclick = new CheckBox ("English");

checkbox spanishclick = new CheckBox ("Spanish");

UIBox languageBox = new UIBox (10, arabicclick,

englishclick, spanishclick);

cmdDone .setOnCommand (languageBox , 1, 7);

Label aboutLabel = new Label ("About You");

cmdDone .setOnCommand (aboutLabel , 0, 8);

TextArea aboutInput = new TextArea ();

aboutInput .setPrompt ("Your Biography");

aboutInput .setRows (4);

cmdDone .setOnCommand (aboutInput , 1, 8);

Button registerButton = new Button ("Register");

cmdDone .setOnCommand (registerButton , 1, 9);

grid, getChildren, addAll (nameLabel, nameInput,  
emailLabel, emailInput, phoneLabel, phoneInput;  
passwordLabel, passwordInput, confirm, confirmInput,  
countryLabel, countryInput, genderLabel, genderBox,  
languageLabel, languageBox, aboutLabel, aboutInput,  
registerButton);

Scene scene = new Scene (grid, 600, 450);  
primaryStage.setScene (scene);  
primaryStage.show();

public static void main (String args) {

    launch (args);

}