Programación

Montero Greshel

Portfolio

Clase 2 Variables en java

Explicacion :En Java, una variable es un contenedor que almacena un valor de un determinado tipo de datos. Las variables se utilizan para almacenar valores que pueden ser utilizados en operaciones o en la ejecución de un programa. En Java, una variable se declara especificando su tipo de datos, su nombre y su valor inicial

En Java, los tipos de variables se clasifican en tres categorías principales:

**Variables locales:** son declaradas dentro de un bloque de código, como una función o un método, y solo existen mientras se ejecuta el bloque. No pueden ser accedidas desde fuera del bloque. Las variables locales no tienen valores predeterminados y deben inicializarse antes de utilizarse.

**Variables de instancia**: son declaradas dentro de una clase pero fuera de cualquier método, constructor o bloque. Estas variables se inicializan cuando se crea un objeto de la clase y existen mientras el objeto exista. Pueden ser accedidas desde cualquier parte de la clase y tienen valores predeterminados.

**Variables estáticas:** son declaradas con la palabra clave static y existen durante toda la duración del programa. Se pueden acceder a ellas desde cualquier parte del programa sin necesidad de crear una instancia de la clase. Las variables estáticas también tienen valores predeterminados.

ejemplo 1

public void ejemplo() {

int x = 5; // variable local declarada e inicializada

System.out.println(x);

}

ejemplo 2

public class Persona {

String nombre; // variable de instancia declarada

int edad; // variable de instancia declarada

}

Clase N°3

Tema : Ventana en java

Explicación :

En la clase de hoy vamos a aprender cómo crear una ventana en el lenguaje de programación Java,un dato importante es que el nombre del archivo en este caso "Ventanita" con el nombre de la clase "public class ventanita"

Lo primero que necesitamos es importar las librerías

import java.awt.\*;

import javax.swing.\*;

Procedemos a crear la ventana y darle las dimensiones que se quiere, el título de la venta

JFrame ventana = new JFrame();

ventana.setTitle(" Primera Ventana");

ventana.setBounds(0, 0, 900, 700);

Estás librerías hacen que nuestra ventana pueda adquirir cualquier color

import java.awt.Color;

*import* java.awt.Color;

*import* java.awt.\*;

*import* javax.swing.\*;

public class ventanita{

public static void main(String []args){

JFrame ventana *=* *new* JFrame();

ventana.setTitle(" primera ventana");

ventana.setBounds(0, 0, 900, 700);

ventana.setResizable(false);

Ventana.setBackground(*new* Color(225,149,133));

panel.setLayout(null);

ventana.add();

ventana.setVisible(true);

}

}

Ejercicio 2: ventana con texto

*import* java.awt.Color;

*import* java.awt.\*;

*import* javax.swing.\*;

public class ventanita{

public static void main(String []args){

JFrame ventana *=* *new* JFrame();

ventana.setTitle(" holis profe");

ventana.setBounds(0, 0, 400, 400);

ventana.setResizable(false);

JPanel panel *=* *new* JPanel();

panel.setBounds(0, 0, 700, 700);

panel.setBackground(*new* Color(225,149,133));

panel.setLayout(null);

JLabel titulo1*=* *new* JLabel("hola profe ");

titulo1.setBounds(100, 40, 150, 150);

titulo1.setFont(*new* Font("Baskerville Old Face", Font.PLAIN, 30));

titulo1.setForeground(Color.BLACK);

ventana.add (titulo1);

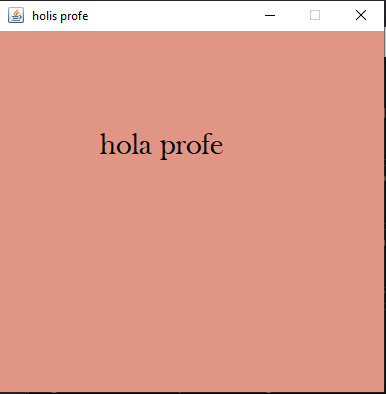
ventana.add(panel);

ventana.setVisible(true);

}

}

Teniendo como resultado la siguiente imagen :



Clase N°4:

Tema : Botones

Explicación:

En la clase de hoy vamos a aprender a hacer un botón en la materia de java y posteriormente colocarle un color , teniendo en cuenta lo aprendido en la clase anterior

Ejercicio 1:Crear un botón con color rosa pastel

JButton boton *=* *new* JButton("Mi boton");

boton.setSize(100, 50);

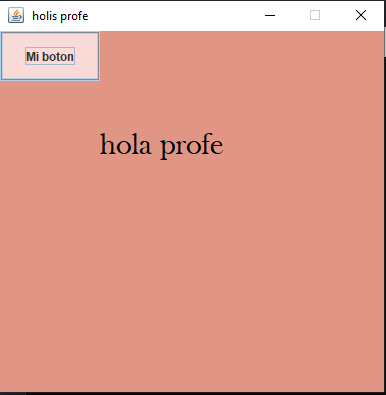
boton.setLocation(50, 50);

ventana.setSize(300, 300);

con el .setBackground vamos a indicar con tabla RGB

boton.setBackground(*new* Color(250, 219, 216 ));

**Resolucion practica ejercicio 1**



ejercicio 2:

crear un boton con color azul pastel

JButton boton *=* *new* JButton("Mi boton");

boton.setSize(100, 50);

boton.setLocation(0, 0);

boton.setBackground(*new* Color(174, 214, 241));

ventana.add(boton);

ventana.add (titulo1);

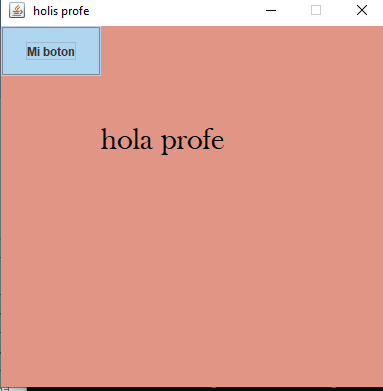
ventana.add(panel);

ventana.setVisible(true);

}

}

resultado



Clase N°4 :Radio Button

Curso:4to 1era

Fecha de clase: 05/03/2023

Tema: Radio Button

Clase N°5

Explicación:

En la clase de hoy vamos a aprender a colocar botones en forma de círculo, para crear un radiobutton en java , se puede utilizar la clase de JRadioButton de la biblioteca Swing de java

Ejemplo 1:

import javax.swing.\*;

public class Radio {

public static *void* main(*String*[] *args*) {

*JFrame* frame = new JFrame("Ejemplo de Radio Buttons");

*JPanel* panel = new JPanel();

*JRadioButton* radio1 = new JRadioButton("Opción 1");

*JRadioButton* radio2 = new JRadioButton("Opción 2");

panel.add(radio1);

panel.add(radio2);

*ButtonGroup* grupo = new ButtonGroup();

grupo.add(radio1);

grupo.add(radio2);

frame.add(panel);

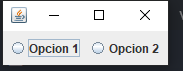
frame.setSize();

frame.setVisible(true);

}

}

Resolución práctica 1:



Ejemplo 2:

import javax.swing.\*;

public class radio {

public static *void* main(*String*[] *args*) {

*JFrame* frame = new JFrame("");

*JPanel* panel = new JPanel();

*JRadioButton* radio1 = new JRadioButton("Seleccionar");

panel.add(radio1);

frame.add(panel);

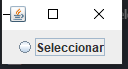
frame.setSize();

frame.setVisible(true);

}

}

Resolución práctica 2:



Clase N° 6

Tema: Ciclo for

Explicacion: Para poder hacer un ciclo “For” se utiliza para repetir un bloque de codiglo de un numero o de una conducción para ser repetida determinadas veces como se le indique ,

for (inicialización; condición; actualización)

Ejemplo 1:

for (int i = 1; i <= 5; i++) {

System.out.println(i);

}

Resolucion practica ejercico 1

1

2

3

4

5

Ejemplo 2 :

int suma *=* 0;

*for* (int i *=* 1; i *<=* 10; i*++*) {

suma *+=* i;

}

System.out.println("La suma de los primeros 10 números naturales es: " *+* suma);

Resolucion practica del ejercicio 2

(La suma de los primeros 10 números naturales es*:* 55)

Clase N° 8

Fecha : 19/ 04/2023

Tema: Condicionales

Explicación : Los condicionales en Java son estructuras de control que permiten ejecutar ciertas líneas de código solo si se cumple una determinada condición. Los condicionales más comunes en Java son el "if", el "if-else".El "if" es el más simple y se utiliza para ejecutar un bloque de código si una determinada condición es verdadera.

if (edad >= 18) {

System.out.println("Eres mayor de edad");

}

El "if-else" se utiliza cuando se quiere ejecutar un bloque de código si la condición es verdadera, y otro bloque si la condición es falsa

if (nota >= 5) {

System.out.println("Has aprobado");

} *else* {

System.out.println("Has suspendido");

Ejemplo 1

public class ParImpar {

public static void main(String[] args) {

int num *=* 5;

*if* (num *%* 2 *==* 0) {

System.out.println(num *+* " es un numero par");

} *else* {

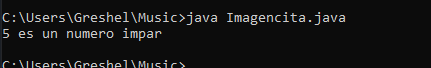
System.out.println(num *+* " es un numero impar");

}

}

}

Resolucion Practica

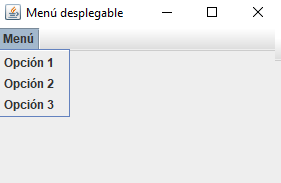


Clase N° 9

Tema:Menús desplegables

Los menús desplegables en Java utilizando el paquete Swing es la manera más fácil de poder hacer un menú que se despliquede colocando cada uno de los items que sea necesarios

Ejemplo 1: Menú desplegable simple Este ejemplo muestra un menú desplegable simple con tres elementos de menú: "Opción 1", "Opción 2" y "Opción 3". Cada vez que se hace clic en una opción del menú, se muestra un mensaje de diálogo que indica la opción seleccionada.



Clase N° 11

Tema: buferedReader

En Java, el "bufferedReader" es una clase utilizada para leer datos de una fuente de entrada, como un archivo o una secuencia de caracteres, de manera eficiente. Proporciona métodos para leer líneas completas de texto, así como caracteres individuales.

Aquí tienes dos ejemplos de cómo se puede utilizar la clase BufferedReader en Java:

Ejemplo 1: Lectura de un archivo de texto línea por línea

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class BufferedReader {

public static void main(String[] args) {

try {

BufferedReader reader = new BufferedReader(new FileReader("archivo.txt"));

String line;

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

reader.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

Clase N° 12

Tema: Temporizador

*import* javax.swing.JFrame;

*import* javax.swing.JLabel;

*import* javax.swing.SwingUtilities;

public class Temporizador extends JFrame {

private JLabel tiempoLabel;

private long tiempoInicial;

public Temporizador() {

tiempoLabel *=* *new* JLabel("Tiempo: 0 segundos");

add(tiempoLabel);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(200, 100);

setLocationRelativeTo(null);

setVisible(true);

iniciarTemporizador();

}

private void iniciarTemporizador() {

tiempoInicial *=* System.currentTimeMillis();

Thread thread *=* *new* Thread(() -> {

*while* (true) {

long tiempoTranscurrido *=* (System.currentTimeMillis() *-* tiempoInicial) */* 1000;

tiempoLabel.setText("Tiempo: " *+* tiempoTranscurrido *+* " segundos");

*try* {

Thread.sleep(1000); *// Espera 1 segundo*

} *catch* (InterruptedException e) {

e.printStackTrace();

}

}

});

thread.start();

}

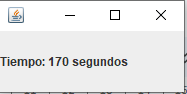
public static void main(String[] args) {

SwingUtilities.invokeLater(Temporizador*::new*);

}

}

teniendo como resultado :



y en el ejemplo 2 podremos pausar el temporizador y reiniciarlo

*import* javax.swing.JButton;

*import* javax.swing.JFrame;

*import* javax.swing.JLabel;

*import* javax.swing.JPanel;

*import* javax.swing.SwingUtilities;

public class Temporizador extends JFrame {

private JLabel tiempoLabel;

private long tiempoInicial;

private long tiempoPausado;

public Temporizador() {

tiempoLabel *=* *new* JLabel("Tiempo: 0 segundos");

JButton pausarButton *=* *new* JButton("Pausar");

pausarButton.addActionListener(e -> pausarTemporizador());

JButton reiniciarButton *=* *new* JButton("Reiniciar");

reiniciarButton.addActionListener(e -> reiniciarTemporizador());

JPanel panel *=* *new* JPanel();

panel.add(tiempoLabel);

panel.add(pausarButton);

panel.add(reiniciarButton);

add(panel);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(250, 100);

setLocationRelativeTo(null);

setVisible(true);

iniciarTemporizador();

}

private void iniciarTemporizador() {

tiempoInicial *=* System.currentTimeMillis();

Thread thread *=* *new* Thread(() -> {

*while* (true) {

*if* (*!*Thread.interrupted()) {

long tiempoTranscurrido *=* ((System.currentTimeMillis() *-* tiempoInicial) *-* tiempoPausado) */* 1000;

tiempoLabel.setText("Tiempo: " *+* tiempoTranscurrido *+* " segundos");

}

*try* {

Thread.sleep(1000); *// Espera 1 segundo*

} *catch* (InterruptedException e) {

e.printStackTrace();

*break*;

}

}

});

thread.start();

}

private void pausarTemporizador() {

tiempoPausado *=* System.currentTimeMillis() *-* tiempoInicial *-* tiempoPausado;

tiempoLabel.setText("Temporizador pausado");

}

private void reiniciarTemporizador() {

tiempoInicial *=* System.currentTimeMillis();

tiempoPausado *=* 0;

tiempoLabel.setText("Tiempo: 0 segundos");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(Temporizador*::new*);

}

}

teniendo como resultado

