Universidad Continental

Guía de Trabajo

Programación Orientada a Objetos

Carol Roxana Rojas Moreno

Guía de Trabajo
Programación Orientada a Objetos

Huancayo, 2024
De esta edición
© Universidad Continental, Oficina de Gestión Curricular Av. San Carlos 1795, Huancayo-Perú
Teléfono: (51 64) 481-430 anexo 7361
Correo electrónico: recursosucvirtual@continental.edu.pe
http://www.continental.edu.pe/

Contenido

Semana 9: Sesión 2

Programas con Gestión de errores y excepciones, Expresiones Lambda, Programación de tareas multiproceso (multihilo)

Semana 10: Sesión 2

Programas con interfaz gráfica Swing, GUI, AWT

Semana 11: Sesión 2

Programas con componente grafico Swing para preparación de interfaces de proyecto final: Menú y formularios

Semana 12: Sesión 2

Programas con la programación visual, manejo de eventos y expresiones

Cuarta Unidad

Administración de información usando gestor de base de datos

Semana 13: Sesión 2

Programas con Puentes entre base de datos y programa (control e interfaz gráfica)

Semana 14: Sesión 2

Programas con Sentencias SQL, ACCESS, ORACLE

Semana 15: Sesión 2

Administración de información para el proyecto grupal

Semana 16: Sesión 2

Trabajo Grupal/proyecto: aplicando las técnicas de programación orientada a objetos, interacción hombre-computador, programación visual y gestión de bases de datos

Referencias

Presentación

La presente guía de trabajo, tiene como propósito fortalecer los fundamentos teóricos a través de prácticas de ejercicios de problemas propuestos, y que permita orientar el desarrollo de los contenidos en las sesiones síncronas.

Los contenidos son: Conceptos: entrada y salida de datos, objetos y clases, fundamentos orientado a objetos, clase interface, cohesión, acoplamiento, expresiones Lambda. Programación visual: manejo de errores, Multithread, delegación de eventos, formularios y menú de opciones. Acceso a datos: API JDBC, sentencias CRUD. Interacción hombre computador: modelos y estilos de interacción, proceso de diseño, prototipos, usabilidad, diseño centrado en el usuario.

El Resultado de Aprendizaje de la asignatura busca que el estudiante sea capaz de elaborar programas en un nivel inicial a partir de las etapas del diseño centrado en el usuario, identificando los fundamentos orientados a objetos con acceso a datos, que usará en un lenguaje de programación, considerando el trabajo en equipo. Por lo que, el estudiante en la Unidad I será capaz de aplicar los conceptos de Clase, Objeto, Herencia, Polimorfismo. En la Unidad II será capaz de desarrollar el modelado de un sistema. En la unidad III será capaz de manejar información a través de la programación visual y manejo de eventos. En la Unidad IV será capaz de conectar a una base de datos administrando la información mediante interfaz gráfica.

Se recomienda al estudiante, revisar los materiales de estudios antes de iniciar cada semana de clase (Flipped Classroom), así como su aula virtual. Participe activamente en clase, ya sea para absolver dudas o para dar aportes a los temas tratados.

Carol Roxana Rojas Moreno

Primera **Unidad**

Principios esenciales de la Programación Orientada a Objetos

Semana 1: Sesión 2

Lenguaje de Programación Orientada a Objetos: Entrada y Salida de Datos

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 1

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

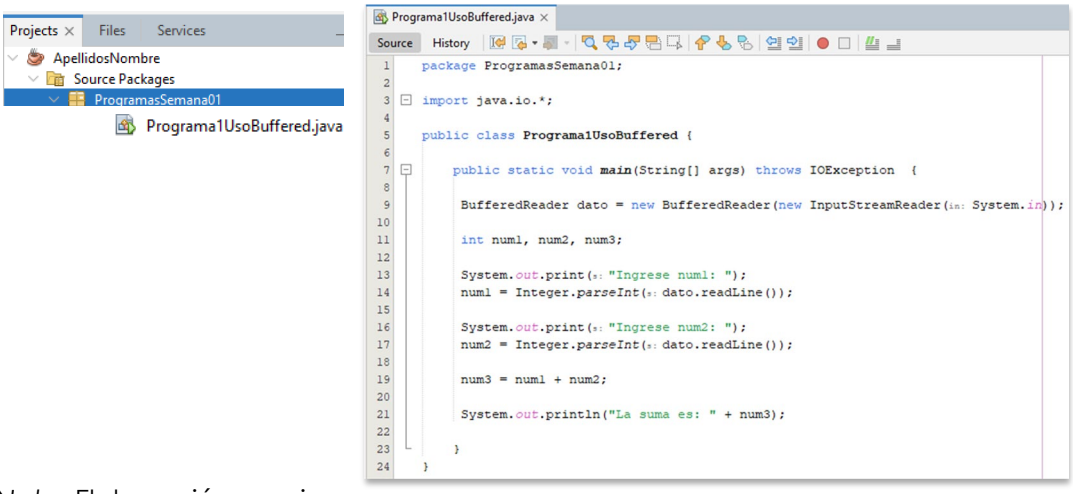
Elaborar programas de cómputo usando el lenguaje de programación orientada a objetos: entrada y salida de datos.

II. Descripción de la actividad por realizar

1. Elabore el siguiente programa ejemplo, en el paquete ProgramasSemana01:

   1.a. Usando *BufferedReader*, elabora el programa para sumar dos números.

Figura 1: Programa UsoBuffered

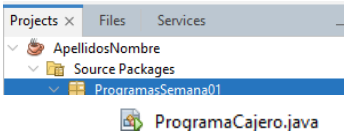


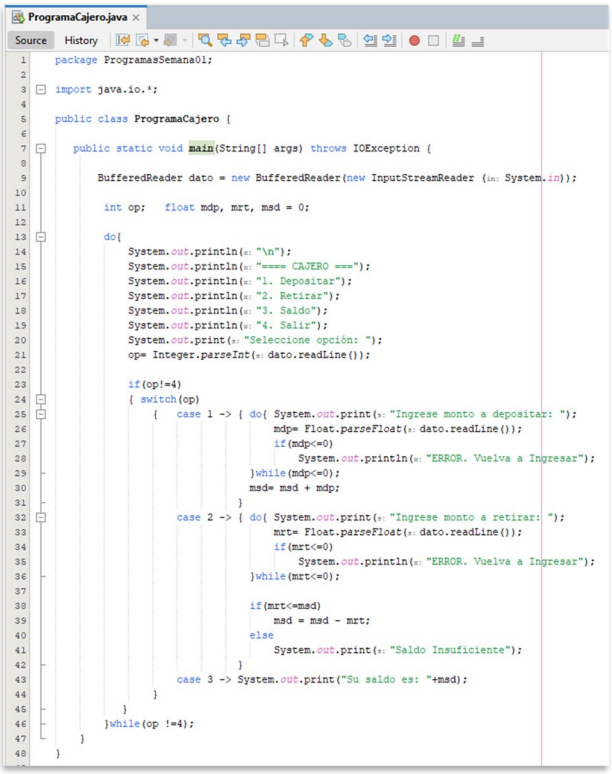
Nota: Elaboración propia

2. Elabore el siguiente programa propuesto, en el paquete ProgramasSemana01:

2.a. Usando *JOptionPane*, elabora el programa para el cajero automático. (tome como referencia el programa elaborado con *BufferedeReader*)

Figura 2: Programa JOptionPane






```java
package ProgramasSemana01;

import java.io.*;

public class ProgramaCajero {

  public static void main(String[] args) throws IOException {

      BufferedReader dato = new BufferedReader(new InputStreamReader (in: System.in));

       int op;   float mdp, mrt, msd = 0;

       do{
           System.out.println(x: "\n");
           System.out.println(x: "==== CAJERO ===");
           System.out.println(x: "1. Depositar");
           System.out.println(x: "2. Retirar");
           System.out.println(x: "3. Saldo");
           System.out.println(x: "4. Salir");
           System.out.print(s: "Seleccione opción: ");
           op= Integer.parseInt(s: dato.readLine());

           if(op!=4)
           { switch(op)
               {   case 1 -> { do{ System.out.print(x: "Ingrese monto a depositar: ");
                                   mdp= Float.parseFloat(s: dato.readLine());
                                   if(mdp<=0)
                                       System.out.println(x: "ERROR. Vuelva a Ingresar");
                               }while(mdp<=0);
                               msd= msd + mdp;
                             }
                   case 2 -> { do{ System.out.print(x: "Ingrese monto a retirar: ");
                                   mrt= Float.parseFloat(s: dato.readLine());
                                   if(mrt<=0)
                                       System.out.println(x: "ERROR. Vuelva a Ingresar");
                               }while(mrt<=0);

                               if(mrt<=msd)
                                   msd = msd - mrt;
                               else
                                   System.out.print(x: "Saldo Insuficiente");
                             }
                   case 3 -> System.out.print("Su saldo es: "+msd);
               }
           }
       }while(op !=4);
   }
}
```

Nota: Elaboración propia

Semana 2: Sesión 2

Programas con Clases, subclases, atributos, operaciones (métodos), encapsulamiento

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 1

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

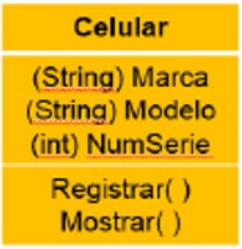
Elaborar programas de cómputo usando Clases, subclases, atributos, operaciones (métodos), encapsulamiento.

II. Descripción de la actividad por realizar

1. Elabore los siguientes programas propuestos, para ingresar y mostrar:

1.a. Elaborar el programa, usando las clases: Celular, PrincipalCelular y leerCadena, en el paquete ProgramasSemana02.Celular.

Figura 3: Clase Celular



1.b. Elaborar el programa, usando las clases: Boleta, PrincipalBoleta y leerCadena, en el paquete ProgramasSemana02.Boleta.

Figura 4: Clase Boleta Venta

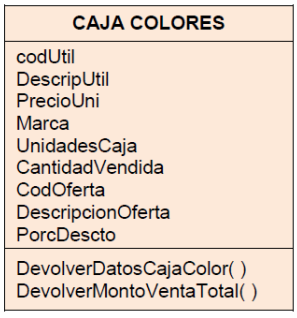
Considerar que, a diferencia de los demás atributos, el atributo Monto Total se asigna a partir del siguiente cálculo:

MontoTotal = Precio * Cantidad

2. Elabore los siguientes programas propuestos, para ingresar y mostrar:

2.a. Elaborar el programa, usando las clases: CajaColores, PrincipalCajaColores y leerCadena, en el paquete ProgramasSemana02.Colores

Figura 5: Clase Caja Colores

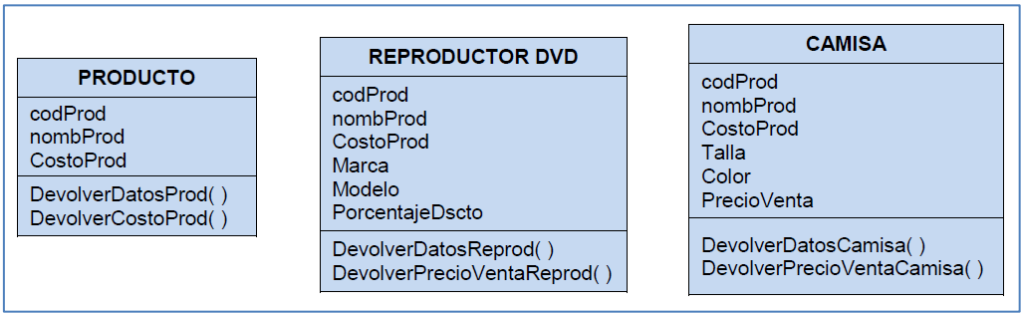


Monto Venta Total = PrecioUni *Cantidad – Monto Descuento

Considere El descuento:
Si Descripción de Oferta es "A", el descuento es de 5 %
Si Descripción de Oferta es "B", el descuento es de 10 %
Si Descripción de Oferta es "C", el descuento es de 15 %

2.b. Elaborar el programa, usando las clases: Producto, ReproductorDVD, Camisa, PrincipalProducto y leerCadena, en el paquete ProgramasSemana02.Almacen

Figura 6: Clase Producto, Reproductor, Camisa



Semana 3: Sesión 2

Programas con Meta clase, herencia, clase abstracta, polimorfismo, herencia, agregación y composición, Instancias y menú de opciones

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 1

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

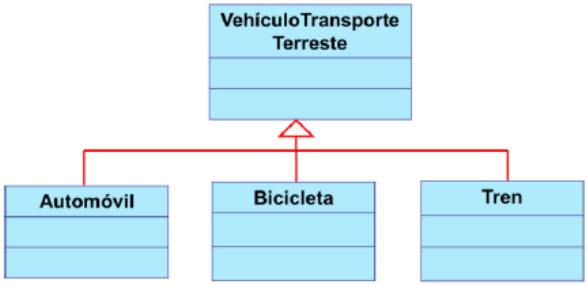
Elaborar programas de cómputo usando meta clase, herencia, clase abstracta, polimorfismo, herencia, agregación y composición, Instancias y menú de opciones.

II. Descripción de la actividad por realizar

1. Elabore los siguientes programas propuestos, para ingresar y mostrar, usando clases abstracta, herencia, menú de opciones.

   1.a. Elaborar el programa, usando clase abstracta, defina los atributos y métodos para cada clase, ingresar y mostrar datos. Calcular y reportar el costo de depreciación de cada vehículo según el porcentaje y el Valor Adquirido de cada vehículo: Automóvil: 25 %, Bicicleta: 5 %, Tren: 45 %, en el paquete ProgramasSemana03.HerenciaAbstractaVehiculo.

Figura 7: Diagrama de Clases con herencia: Vehículo



1ro Elaborar
Clase Abstracta Padre Vehículo

Figura 8: Código Clase Padre Vehículo

Vehiculo.java

```
package ProgramasSemana03.HerenciaAbstractaVehiculo;

public abstract class Vehiculo {

    public float precioIni;
    public float MontoDepre;
    public float Total;

    public abstract float calcularPrecioFinal();
}
```

2do Elaborar
Clase Hiias Automóvil – Bicicleta - Tren

Figura 9: Código Clases Hijas: Automóvil, Bicicleta, Tren

Automovil.java

```
package ProgramasSemana03.HerenciaAbstractaVehiculo;

public class Automovil extends Vehiculo{

    float PorceDep = 0.25f;

    public Automovil(float precio_auto){
      precioIni = precio_auto;
    }

    public float calcularPrecioFinal(){
        MontoDepre =( float)(precioIni*PorceDep);
        Total = precioIni - MontoDepre;
        return Total;
    }

    public float depreciacion(){
        MontoDepre =( float)(precioIni*PorceDep);
        return MontoDepre;
    }
}
```

Bicicleta.java

```
package ProgramasSemana03.HerenciaAbstractaVehiculo;

public  class Bicicleta extends Vehiculo{

    float PorceDep = 0.05f;

    public Bicicleta(float precio_bici){
        precioIni = precio_bici;
    }

    public float calcularPrecioFinal(){
        MontoDepre =( float)(precioIni*PorceDep);
        Total = precioIni - MontoDepre;
        return Total;
    }

    public float depreciacion(){
        MontoDepre =( float)(precioIni*PorceDep);
        return MontoDepre;
    }
}
```

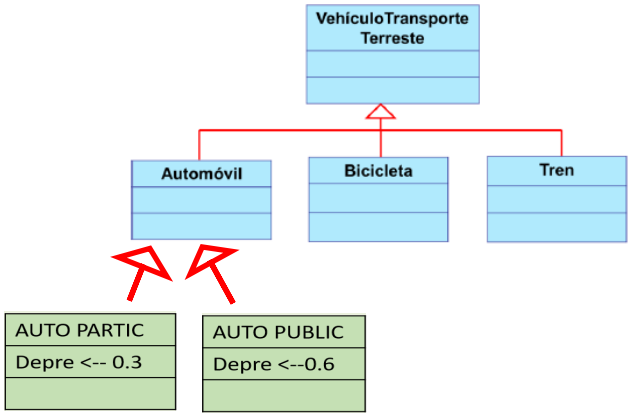
Tren.java

```
package ProgramasSemana03.HerenciaAbstractaVehiculo;

public class Tren extends Vehiculo {

    float PorceDep = 0.45f;

    public Tren(float precio_tren){
        precioIni = precio_tren;
    }

    public float calcularPrecioFinal(){
        MontoDepre =( float)(precioIni*PorceDep);
        Total = precioIni - MontoDepre;
        return Total;
    }

    public float depreciacion(){
        MontoDepre =( float)(precioIni*PorceDep);
        return MontoDepre;
    }
}
```

2. Elabore los siguientes programas propuestos, para ingresar y mostrar, usando clases, clase abstracta, menú de opciones:

2.a. Elaborar el programa, basándose el programa anterior, agregando subclases a la clase Automóvil, en el paquete ProgramasSemana03.HerenciaAbstractaVehiculoParticPublic

Figura 10: Diagrama de Clases herencia: Clases Hijas Particular y Público



1ro Cambiar Automóvil
(ya no implementa el método abstracto)

Figura 11: Código de clase padre Automóvil: antes y después

Antes

Automovil.java

Source History

```
1  package ProgramasSemana03.HerenciaAbstractaVehiculo;
2
3  public class Automovil extends Vehiculo{
4
5      float PorceDep = 0.25f;
6
7      public Automovil(float precio_auto){
8        precioIni = precio_auto;
9      }
10
11     public float calcularPrecioFinal(){
12         MontoDepre =( float)(precioIni*PorceDep);
13         Total = precioIni - MontoDepre;
14         return Total;
15     }
16
17     public float depreciacion(){
18         MontoDepre =( float)(precioIni*PorceDep);
19         return MontoDepre;
20     }
21 }
```

Después

Automovil.java

Source History

```
1  package ProgramasSemana03.HerenciaAbstractaVehiculoParticPublic;
2
3  public class Automovil extends Vehiculo{
4
5      @Override
6      public float calcularPrecioFinal() {
7
8          throw new UnsupportedOperationException(message:"Not supported yet.");
9
10         // AQUI no implementa el método abstracto creado en VEHICULO
11         // AQUI el método abstracto está vacio,
12         // por que se va a impelnetar en sus clases hijas: AutoPartic y AutoPublic
13     }
14 }
```

2do Elaborar
Clase Hijas AutoPartic – AutoPublic

Figura 12: Código de clases hijas Particular y Público

AutoParticular.java

```
package ProgramasSemana03.HerenciaAbstractaVehiculoParticPublic;

public class AutoParticular extends Automovil {

    float PorceDep = 0.6f;

    public AutoParticular(float precio_auto){
      precioIni = precio_auto;
    }

    public float calcularPrecioFinal(){
        MontoDepre =( float)(precioIni*PorceDep);
        Total = precioIni - MontoDepre;
        return Total;
    }

    public float depreciacion(){
        MontoDepre =( float)(precioIni*PorceDep);
        return MontoDepre;
    }
}
```

AutoPublico.java

```
package ProgramasSemana03.HerenciaAbstractaVehiculoParticPublic;

public class AutoPublico extends Automovil{
    float PorceDep = 0.3f;

    public AutoPublico(float precio_auto){
       precioIni = precio_auto;
    }

    public float calcularPrecioFinal(){
        MontoDepre =( float)(precioIni*PorceDep);
        Total = precioIni - MontoDepre;
        return Total;
    }

    public float depreciacion(){
        MontoDepre =( float)(precioIni*PorceDep);
        return MontoDepre;
    }
}
```

Semana 4: Sesión 2

Programas usando los principios esenciales de la programación orientada a objetos

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 1

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

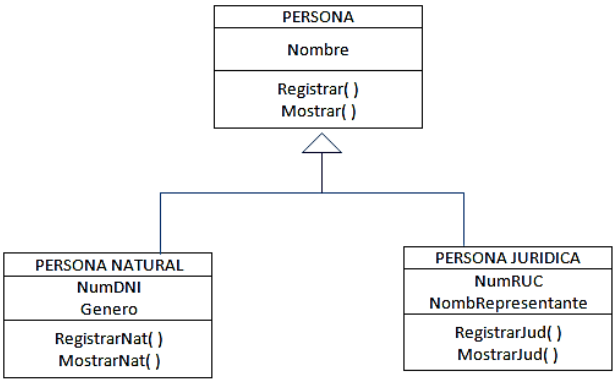
Elaborar programas de cómputo usando los principios esenciales de la programación orientada a objetos.

II. Descripción de la actividad por realizar

1. Elabore los siguientes programas propuestos, para ingresar y mostrar:

1.a. Elaborar el programa, usando herencia y subclases, en el paquete ProgramasSemana04.HerenciaPersona.

Figura 13: Herencia de la Clase Persona



Segunda

Unidad

Interacción hombre computador

Semana 5: Sesión 2

Modelado con diagrama de proceso actual y diagrama de casos de uso

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 2

Nombres y apellidos: .....

Instrucciones

El estudiante, formando equipo para el proyecto grupal, elabora los diagramas para el modelado del negocio.

I. Propósito

Elaborar el modelado de un sistema usando Diagrama de proceso actual y Diagrama de Casos de Uso de requerimientos, con trabajo colaborativo en equipo para una empresa real.

II. Descripción de la actividad por realizar

1. Elabore el Diagrama BPMN del proceso actual de su proyecto grupal.
2. Elabore el Diagrama de Casos de Uso del proceso actual de su proyecto grupal.

Semana 6: Sesión 2

Diagramas de clases con estereotipo, con trabajo colaborativo en equipo

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 2

Nombres y apellidos: .....

Instrucciones

Cada estudiante, formando equipo para el proyecto grupal, elabora los diagramas para el modelado del diseño de un sistema.

I. Propósito

Elaborar el modelado de un sistema usando Diagrama de Clases con estereotipo, con trabajo colaborativo en equipo para una empresa real.

II. Descripción de la actividad por realizar

1. Elabore el Diagrama de Clases con estereotipo del proceso actual de su proyecto grupal.
2. Presentar el trabajo realizado en clase o aula virtual.

Semana 7: Sesión 2

Cohesión y acoplamiento. Diagrama arquitectónico

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 2

Nombres y apellidos: .....

Instrucciones

Cada estudiante, formando equipo para el proyecto grupal, elabora los diagramas para el modelado del diseño de un sistema.

I. Propósito

Identifica diseño arquitectónico, con trabajo colaborativo en equipo para una empresa real.

II. Descripción de la actividad por realizar

1. Determine qué tipo de diseño y que diagrama arquitectónico
2. Presentará para su proyecto grupal.

Semana 8: Sesión 2

Presentación de Trabajo Grupal/ proyecto: diagrama de proceso, diagrama de casos de uso y diagrama de clases con estereotipo

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 2

Nombres y apellidos: .....

Instrucciones

El estudiante, formando equipo para el proyecto grupal, elabora los diagramas para el modelado del diseño de un sistema.

I. Propósito

Presenta el modelado de un sistema demostrando la correcta aplicación de cohesión y acoplamiento, conceptos de interacción hombre computador, y la utilización adecuada de diagramas para satisfacer las necesidades específicas de una empresa real.

II. Descripción de la actividad por realizar

1. Presenta al aula virtual el archivo según formato, con la información del proceso y los diagramas elaborados, para su proyecto grupal.
2. Presentar el trabajo realizado en clase o aula virtual.

Tercera **Unidad**

Programación visual, manejo de eventos y expresiones

Semana 9: Sesión 2

Gestión de errores y excepciones. Expresiones Lambda. Programación de tareas multiproceso (multihilo)

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 3

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

Elaborar programas de cómputo usando Gestión de errores y excepciones, Expresiones Lambda, Programación de tareas multiproceso (multihilo).

II. Descripción de la actividad por realizar

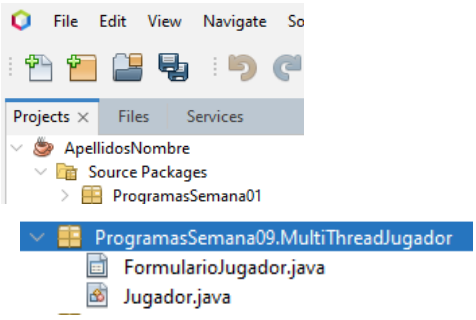
1. Elabore los siguientes programas propuestos, para ingresar y mostrar:

1.a. Elaborar el programa, usando Multithreads para el caso de un jugador, en el paquete ProgramasSemana09.MultiThreadJugador

FUENTE:
https://www.youtube.com/watch?v=mepVH9igIME.

Considere crear las clases Jugador y FormularioJugador:

Figura 14: Paquete y archivos de clases



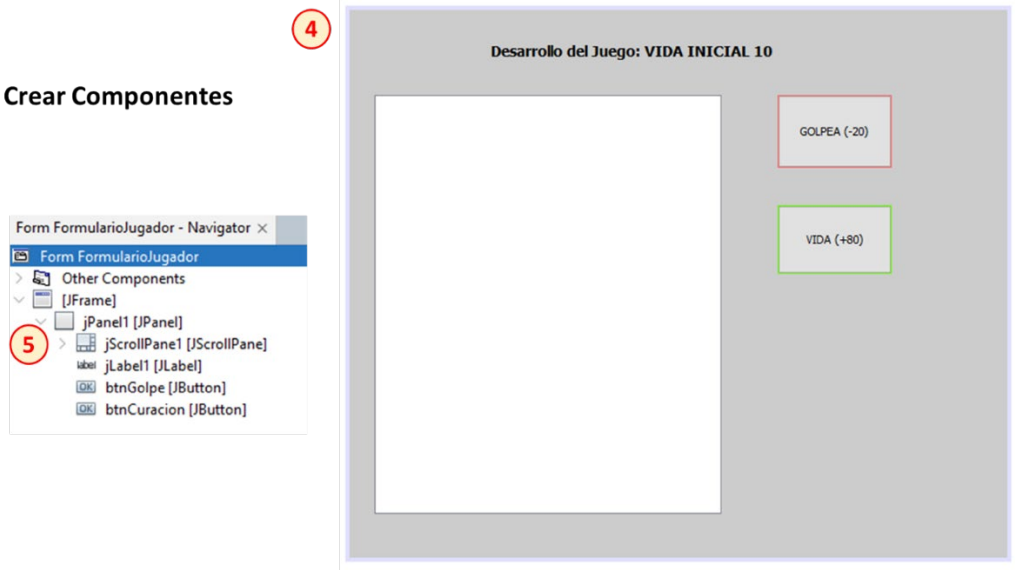
Clase java class: Jugador:

Figura 15: Código de Clase Jugador


```
package ProgramasSemana09.Multihilos;

import java.util.logging.Level;
import java.util.logging.Logger;

public class Jugador {

    int vida;
    int curacion;

    public Jugador(){

        vida = 10;
        curacion = 80;
    }

    synchronized public void RecibirGolpe(int cantidad){

        vida = vida - cantidad;

        try {
            Thread.sleep(5);
        } catch (InterruptedException ex) {
            Logger.getLogger(Jugador.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    synchronized public void RecibirVida(int curacion){

        vida = vida + curacion;

        try {
            Thread.sleep(5);
        } catch (InterruptedException ex) {
            Logger.getLogger(Jugador.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    synchronized public boolean DeclararMuerto(){

        return (vida <= 0);
    }
}
```

Clase jFrame Form: FormularioJugador

Figura 16: Interfaz Gráfica del Juego



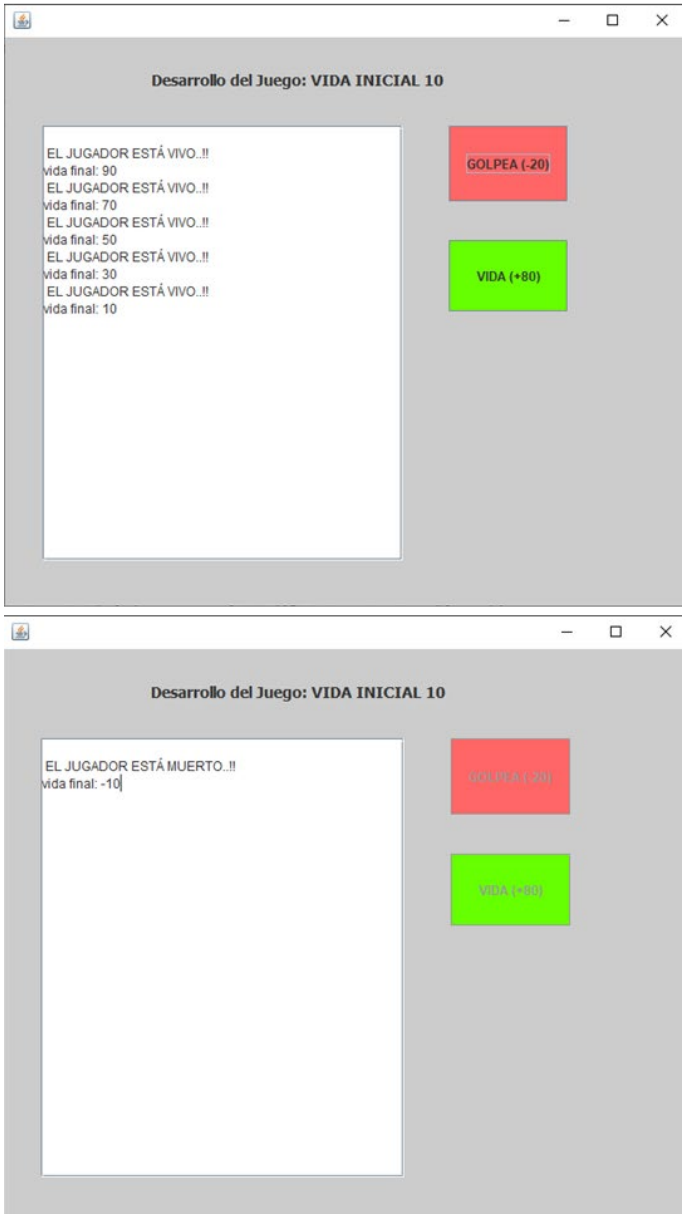
Crear los Hilos por Proceso, e invocar con el objeto hilo1, hilo2, hilo3:

Figura 17: Código de Interfaz Gráfica del juego


```
package ProgramasSemana09.Multihilos;
import java.awt.Color;
import javax.swing.JOptionPane;

public class FormularioJugador extends javax.swing.JFrame {

        Jugador jugadorEjecucion = new Jugador();

        Hilo01 h1 = new Hilo01(jugadorEjecucion);
        Hilo02 h2 = new Hilo02(jugadorEjecucion);
        Hilo03 h3 = new Hilo03(jugadorEjecucion);

    public FormularioJugador() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    Generated Code
    private void btnGolpeActionPerformed(java.awt.event.ActionEvent evt) {
        h1.run();
        h3.run();
    }
    private void btnCuracionActionPerformed(java.awt.event.ActionEvent evt) {
        h2.run();
        h3.run();
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {...6 lines });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton btnCuracion;
    private javax.swing.JButton btnGolpe;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea txtaResultados;
    // End of variables declaration

    //PRIMERO CREAR HILOS:
    class Hilo01 extends Thread{
        Jugador jugad;

        public Hilo01 (Jugador tempJugador){
            jugad = tempJugador;
        }

        public void run(){
            jugad.RecibirGolpe(20);
        }
    }

    class Hilo02 extends Thread{
        Jugador jugad;

        public Hilo02 (Jugador tempJugador){
            jugad = tempJugador;
        }

        public void run(){
            jugad.RecibirVida(80);
        }
    }

    class Hilo03 extends Thread{
        Jugador jugad;

        public Hilo03 (Jugador tempJugador){
            jugad = tempJugador;
        }

        public void run(){

            if(jugad.DeclararMuerto()){
                txtaResultados.setText("\n EL JUGADOR ESTÁ MUERTO..!!\n"+"vida final: "+jugad.vida);
                btnGolpe.setEnabled(false);
                btnCuracion.setEnabled(false);
            }else
                txtaResultados.append("\n EL JUGADOR ESTÁ VIVO..!!\n"+"vida final: "+jugad.vida);
            }
    }
}
```

Figura 18: Interfaz Gráfica del juego en ejecución

Al ejecutar:



Semana 10: Sesión 2

Programas con interfaz gráfica Swing, AWT

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 3

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

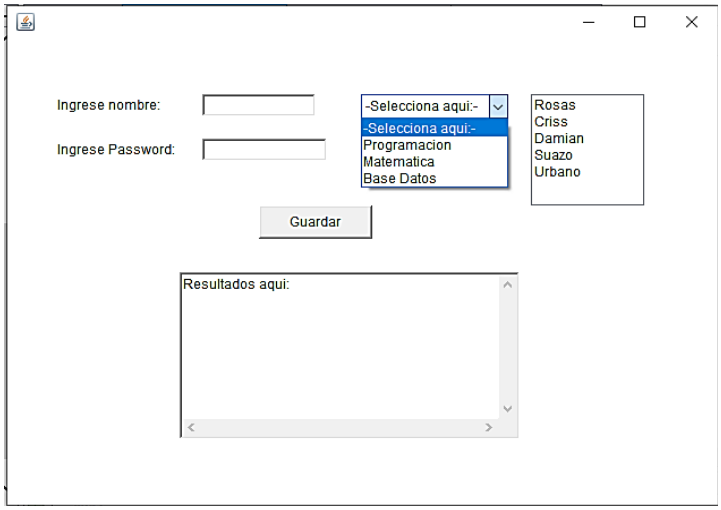
Elaborar programas de cómputo usando Tipos de interfaz gráfica Swing, AWT.

II. Descripción de la actividad por realizar

1. Elabore los siguientes programas propuestos, para ingresar y mostrar:

1.a. Elaborar el programa, usando SWING para crear el siguiente formulario y el código para el botón Guardar que permite mostrar en el TextArea lo ingresado o seleccionado, en el paquete ProgramasSemana10.GuardarDatos.

Figura 19: Interfaz Gráfica para Guardar



Semana 11: Sesión 2

Programas con componente grafico Swing para preparación de interfaces de proyecto final: Menú y formularios

.

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 3

Nombres y apellidos: .....

.

Instrucciones

El estudiante, formando equipo para el proyecto grupal, elabora los formularios de un sistema.

I. Propósito

Elaborar programas de cómputo usando componente grafico Swing para preparación de interfaces de proyecto final: Menú y formularios, para un proyecto grupal de un proceso de una empresa real.

II. Descripción de la actividad por realizar

1. Elabore los siguientes programas propuestos, para formularios:

   1.a. Elaborar el programa, con el formulario del menú principal y formularios de transacciones de su proyecto grupal de un proceso de una empresa real, usando la herramienta de desarrollo de interfaces gráficas.

   Considere para el formulario Menú, como mínimo las opciones de:

   Transacción – Gestión de Datos – Reporte de Datos

   Considere para cada opción de menú, los formularios o sub menús.

Semana 12: Sesión 2

Programas con la programación visual, manejo de eventos y expresiones

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 3

Nombres y apellidos: .....

Instrucciones

El estudiante, formando equipo para el proyecto grupal, elabora los formularios de un sistema.

I. Propósito

Elaborar programas de cómputo usando Programación visual, manejo de eventos y expresiones.

II. Descripción de la actividad por realizar

1. Elabore los siguientes programas propuestos, para formularios:

   1.a. Elaborar el formulario de restar, del formulario menú principal de la semana anterior

   1.b. Implementar el código para el evento que permita calcular la operación de resta, usando la herramienta de desarrollo de interfaces gráficas.

Cuarta **Unidad**

Administración de información usando gestor de base de datos

Semana 13: Sesión 2

Puentes entre base de datos y programa (control e interfaz gráfica)

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 4

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

I. Propósito

Elaborar programas de cómputo usando Puentes entre base de datos y programa con oracle y access.

II. Descripción de la actividad por realizar

1. Elaborar el programa, con un formulario para guardar datos en la tabla Countries, usando la herramienta de desarrollo de interfaces gráficas y el gestor de base de datos Oracle.
2. Presentar el trabajo realizado en clase o aula virtual.

Semana 14: Sesión 2

Sentencias SQL, ACCESS, ORACLE

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 4

Nombres y apellidos: .....

Instrucciones

El estudiante, primero en forma individual, elabora los programas ejemplos: 1), segundo, formando equipo de pares, elabora los programas propuestos: 2).

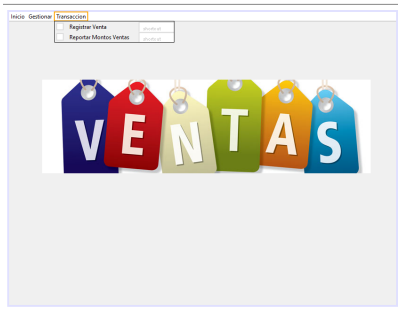
I. Propósito

Elaborar programas de cómputo usando Sentencias SQL, ACCESS, ORACLE.

II. Descripción de la actividad por realizar

1. Revisa el programa dado en clase para usarlo de base para su proyecto grupal.

Figura 20: Interfaz Gráfica para Inicio de Proyecto



2. Realiza el avance de su trabajo grupal.

Semana 15: Sesión 2

Administración de información mediante programa

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 4

Nombres y apellidos: .....

Instrucciones

El estudiante, formando equipo para el proyecto grupal, elabora las interfaces gráficas y la conectividad de un sistema de su trabajo grupal.

I. Propósito

Elabora programas de cómputo usando la administración de información, con trabajo colaborativo en equipo para una empresa real.

II. Descripción de la actividad por realizar

1. Diseña el formulario de menú de opciones y los formularios con conectividad a una base de datos, que presentará para su proyecto grupal, usando la herramienta de desarrollo de interfaces gráficas y el gestor de base de datos Access.

Semana 16: Sesión 2

Presentación de Trabajo Grupal/proyecto: aplicando las técnicas de programación orientada a objetos, interacción hombre-computador, programación visual y gestión de bases de datos

Sección: ..... Fecha: ...../...../..... Duración: 60 minutos Docente: ..... Unidad: 4

Nombres y apellidos: .....

Instrucciones

El estudiante, formando equipo para el proyecto grupal, elabora los diagramas para el modelado del diseño de un sistema.

I. Propósito

Presenta el proyecto final, demostrando la correcta aplicación de técnicas de programación orientada a objetos, interacción hombre-computador, programación visual y gestión de bases de datos, para satisfacer las necesidades específicas de una empresa real.

III. Descripción de la actividad por realizar

1. Presenta al aula virtual, los archivos según formato, con la información del proceso, las imágenes de diagramas, formularios código, base de datos, para su proyecto grupal.

Referencias

Aws. (2023). *¿Qué es diagramación de la arquitectura?* https://aws.amazon.com/es/what-is/architecture-diagramming/

Bermudez P. (2019). *Diseño Centrado en el Usuario*. https://es.slideshare.net/pbermudez10/diseo-centrado-en-el-usuario-149093168

Cosmina, L. (2022). *Java 17 for Absolute Beginners*. Edinburgh, UK: Apress.

Fernando Herrera. (2022, 06 de abril). *Acoplamiento y cohesión* [vídeo]. YouTube. https://youtu.be/PtHgco7oGks

Jaramillo, S. Cardona S. (2018). *Principios de programación orientada a objetos*. Armenia, Colombia: ElizCOM SAS.

JOptionPane: showMessageDialog y showInputDialog. (s/f). Edu4java.com. Recuperado el 28 de febrero de 2024, de http://www.edu4java.com/es/java/joptionpane-showmessagedialog-showinputdialog.html

SQL Developer. (s/f). Software. Recuperado el 28 de febrero de 2024, de https://www.oracle.com/database/sqldeveloper/

SQL SERVER. (S/f). Software. Recuperado el 28 de febrero de 2024, de https://www.mysql.com/