

CLASE PRINCIPAL

package pooinvercapiparim;

import javax.swing.JOptionPane;

public class Pooinvercapiparimpar {

public static void main(String[] args) {

int opcion;

boolean continuar = true;

while(continuar){

String num= JOptionPane.showInputDialog("MANEJO DE MENUS \nSELECIONE UNA OPCION " + "\n\n1.- Numero Invertido" + "\n2.- Numero Capicua \n3.- Cantidad de Pares e Impares en un nùmero" + "\n4.-Salir");

opcion=Integer.parseInt(num);

switch (opcion){

case 1:

String xx=JOptionPane.showInputDialog("Introduzca el numero");

int x=Integer.parseInt(xx);

InvertirObj inv=new InvertirObj(x);

//JOptionPane.showInputDialog(null,"El num inver es"+inv.invertir());

JOptionPane.showMessageDialog(null, "El número invertido es: " + inv.invertir());

break;

case 2:

String yy=JOptionPane.showInputDialog("Introduz el numero");

int y=Integer.parseInt(yy);

CapicuaObj cap=new CapicuaObj(y);

// cap.CapicuaObj(y);

JOptionPane.showMessageDialog(null,"Si el Capicua saldra Tue caso contrario False : "+cap.capicua());

break;

case 3:

String nn=JOptionPane.showInputDialog("Introduz el numero");

int n=Integer.parseInt(nn);

ParImparObj numero=new ParImparObj(n);

JOptionPane.showMessageDialog(null," "+numero.ParImpar());

break;

case 4:

continuar = false;

}

}

}

}

InvertirObje.java

package pooinvercapiparim;

public class InvertirObj {

private int numero;

// Constructor

public InvertirObj(int num) {

//this.numero = new Numero(num);

this.numero = num;

}

// Getter y Setter para el número

public int getNumero() {

return numero;

}

public void setNumero(int num) {

this.numero=num;

}

// Método para obtener el número invertido

public int invertir() {

int dig,reversedNum=0;

while (numero > 0) {

dig = numero % 10;

numero = numero / 10;

reversedNum = (reversedNum \* 10) + dig;

}

return reversedNum;

}

}

**CapicuaObj.java**

package pooinvercapiparim;

public class InvertirObj {

private int numero;

// Constructor

public InvertirObj(int num) {

//this.numero = new Numero(num);

this.numero = num;

}

// Getter y Setter para el número

public int getNumero() {

return numero;

}

public void setNumero(int num) {

this.numero=num;

}

// Método para obtener el número invertido

public int invertir() {

int dig,reversedNum=0;

while (numero > 0) {

dig = numero % 10;

numero = numero / 10;

reversedNum = (reversedNum \* 10) + dig;

}

return reversedNum;

}

}

ParImparObj.java

package pooinvercapiparim;

public class ParImparObj {

private int numero;

// Constructor

public ParImparObj(int num) {

this.numero = num;

}

// Método para contar los números pares e impares

public String ParImpar() {

int dig;

int pares = 0; // Inicializamos el contador de pares

int impares = 0; // Inicializamos el contador de impares

int tempNum = numero;

while (tempNum > 0) {

dig = tempNum % 10; // Extraemos el último dígito

if (dig % 2 == 0) { // Si el dígito es par

pares++;

} else { // Si el dígito es impar

impares++;

}

tempNum = tempNum / 10; // Eliminamos el último dígito

}

return "Pares: " + pares + ", Impares: " + impares;

}

// Getter y Setter para el número

public int getNumero() {

return this.numero;

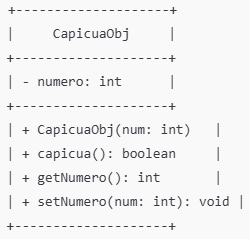
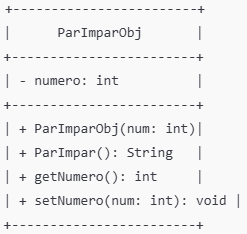
}

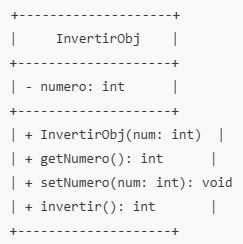
public void setNumero(int num) {

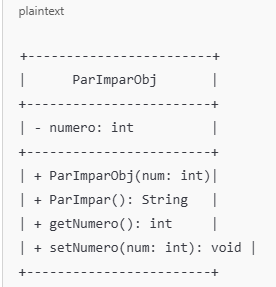
this.numero = num;

}

}

DIAGRAMAS ULM

º

**Pasos para realizar un Diagrama UML:**

**1. Identificación de Clases**

* **Paso 1**: Identifica las clases en el código que deseas representar en el diagrama UML. En tu caso, tienes clases como InvertirObj, CapicuaObj y ParImparObj.
* **Paso 2**: Escribe el nombre de cada clase en el diagrama. El nombre de la clase siempre debe ir en la parte superior del rectángulo del diagrama, en una caja separada.

**2. Atributos (Propiedades o Variables)**

* **Paso 1**: Identifica los atributos de cada clase. Los atributos son las variables que almacenan datos dentro de la clase.
* **Paso 2**: Escribe los atributos debajo del nombre de la clase, con el siguiente formato:
  + **Visibilidad** (+ para público, - para privado, # para protegido)
  + **Tipo de dato** (por ejemplo, int, String, etc.)
  + **Nombre del atributo** (por ejemplo, numero).
  + Ejemplo: - numero: int (significa que numero es un atributo privado de tipo entero).

**3. Métodos (Funciones o Acciones)**

* **Paso 1**: Identifica los métodos en cada clase. Los métodos son funciones que realizan acciones en la clase.
* **Paso 2**: Escribe los métodos debajo de los atributos, con el siguiente formato:
  + **Visibilidad** (como en los atributos).
  + **Tipo de retorno** (por ejemplo, void, int, String).
  + **Nombre del método** (por ejemplo, invertir()).
  + **Parámetros** (si tiene, como en setNumero(num: int)).
  + Ejemplo: + invertir(): int (significa que invertir() es un método público que retorna un valor entero).

**4. Relaciones entre Clases (si es necesario)**

* **Paso 1**: Identifica las relaciones entre las clases (por ejemplo, herencia, composición o asociación).
* **Paso 2**: Dibuja las relaciones entre las clases usando líneas:
  + **Línea continua** con un triángulo apuntando hacia la clase base (para herencia).
  + **Línea continua** con rombos (para composición o agregación).
  + **Línea continua sin rombos ni triángulos** (para asociación).
* Ejemplo: Si una clase usa otra (asociación), dibuja una línea entre ellas.