Tarea hito 2 - POO variables, arrays, clases, packages

MANEJO DE CONCEPTOS

En este apartado se responden las preguntas dadas en la defensa en base a lo avanzado en clases

MANEJO DE CONCEPTOS

01

1. ¿A que se refiere cuando se habla de POO?

Nos referimos a programación orientada a objetos

02

2. ¿Cuáles son los 4 componentes que componen POO?

- Objetos
- Clases
- Métodos
- Atributos

03

3. ¿Cuáles son los pilares de POO?.

- Abstracción,
- Encapsulamiento
- Polimorfismo
- Herencia

MANEJO DE CONCEPTOS

04

4. ¿Qué es Encapsulamiento y muestre un ejemplo?

Son los moificadores de acceso
como: public, private y protected

```java
package claseporv;
import java.util.*;
public class Provincia
{
 private String nombre;

 public Provincia()
 {
  this.nombre = "Chulumani y chacaltalla";
 }

 public String getNombre()
 {
  return nombre;
 }

 public void setNombre(String nombre)
 {
  this.nombre = nombre;
 }

 public void mostrar()
 {
  System.out.println("Mostrando Clase Provincia");
  System.out.println("Nombre: " + this.getNombre());
 }
}
```

MANEJO DE CONCEPTOS

05

5. ¿Qué es Abstracción y muestre un ejemplo?

La abstracción **es un pilar o característica de la programación orientada a objetos que va a permitir que los objetos puedan interactuar sin necesidad de conocer los detalles del funcionamiento**

```
public abstract class FiguraGeometrica
{
 public abstract double calcularArea();
 public abstract double calcularPerimetro();
}
```

```
public class Circulo extends FiguraGeometrica
{
 private double radio;
 public Circulo(double radio)
 {
  this.radio = radio;
 }
 public double calcularArea()
 {
  return Math.PI * radio * radio;
 }
 public double calcularPerimetro()
 {
  return 2 * Math.PI * radio;
 }
}
```

MANEJO DE CONCEPTOS

06

6. ¿Que es Herencia y muestre un ejemplo?

La herencia permite que una clase herede propiedades y comportamientos de otra clase, lo que permite crear jerarquías de clases y reutilizar código.

```
public class Persona
{
 private String nombre;
 private int edad;
 public Persona(String nombre, int edad)
 {
  this.nombre = nombre;
  this.edad = edad;
 }
 public String getNombre()
 {
  return nombre;
 }
 public int getEdad()
 {
  return edad;
 }
}

public class Empleado extends Persona
{
 private double salario;
 public Empleado(String nombre, int edad, double salario)
 {
  super(nombre, edad);
  this.salario = salario;
 }
 public double getSalario()
 {
  return salario;
 }
}
```

MANEJO DE CONCEPTOS

07

7. ¿Qué es Polimorfismo y muestre un ejemplo?

el polimorfismo permite que un objeto se comporte de diferentes maneras según el contexto en el que se use.

```
public interface Animal
{
 void hacerSonido();
}
public class Perro implements Animal
{
 public void hacerSonido()
 {
  System.out.println("Guau!");
 }
}
public class Gato implements Animal
{
 public void hacerSonido()
 {
  System.out.println("Miau!");
 }
}
public class Vaca implements Animal
{
 public void hacerSonido()
 {
  System.out.println("Muu!");
 }
}
```

MANEJO DE CONCEPTOS

08

8. ¿Que es un ARRAY?

Es un conjunto de valores del mismo tipo almacenados de manera ordenada y ascendentes

09

9. ¿Qué son los paquetes en JAVA?

Son un **conjunto de clases e interfaces relacionadas entre sí, de manera que las clases y métodos no colisionen entre si y mantener un cierto orden.**

10

10.¿Cómo se define una clase main en JAVA y muestra un ejemplo?

debemos crear una clase que contenga un método main() público y estático. El método main() debe tener el siguiente formato:

```
public class MiPrograma
{
 public static void main(String[] args)
 {
  System.out.println("¡Hola, mundo!");
 }
}
```

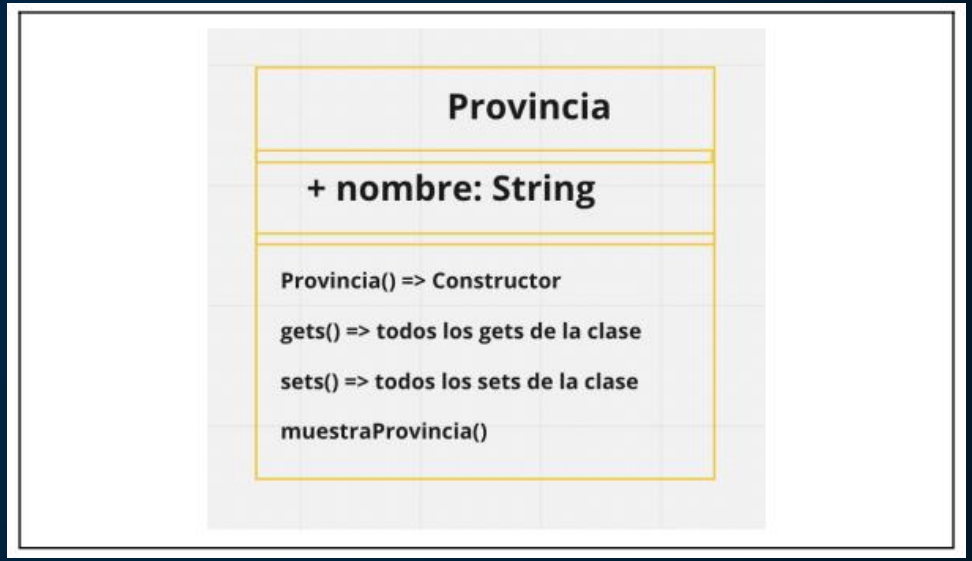
PARTE PRACTICA

Provincia.java

11

11. Generar la clase Provincia.

- Crear una clase MAIN
  - Crear todos los gets y sets de la clase.
  - El constructor no recibe parámetros.
  - Crear una instancia de la clase Provincia
  - Mostrar los datos de una provincia
- Adjuntar el código JAVA generado.



```java
package TareaH2;
public class Provincia {
private String nombre;
public Provincia() {
}
public String getNombre() {
return nombre;
}
public void setNombre(String nombre) {
this.nombre = nombre;
}
public void mostrarProvincia() {
System.out.println("Nombre de la provincia:
" + getNombre());
}
}
```

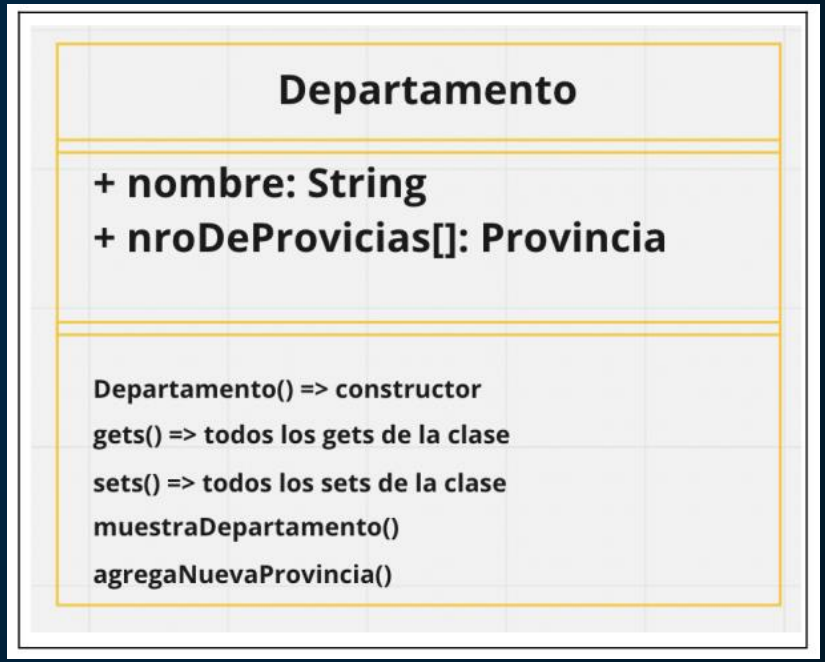
Departamento.java

PARTE PRACTICA

12

12.Generar la clase Departamento.

- ○ Diseño.
- ○ Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)
  - ■ Crear todos los gets y sets de la clase.
  - ■ El constructor no recibe parámetros.
  - ■ Crear una instancia de la clase Departamento.
  - ■ Omitir el método agregaNuevaProvincia()
  - ■ Mostrar los datos de los departamentos.
- ○ Adjuntar el código JAVA generado.



```java
package TareaH2;
public class Departamento {
private String nombre;
private int nroProv;
private Provincia[] provincias;
public Departamento() {
}
public String getNombre() {
return nombre;
}
public void setNombre(String nombre) {
this.nombre = nombre;
}
public int getNroProv() {
return nroProv;
} this.nroProv = nroProv;
}
public Provincia[] getProvincias() {
return provincias;
}

public void setNroProv(int nroProv) {

public void setProvincias(Provincia[] provincias) {
this.provincias = provincias;
}
```

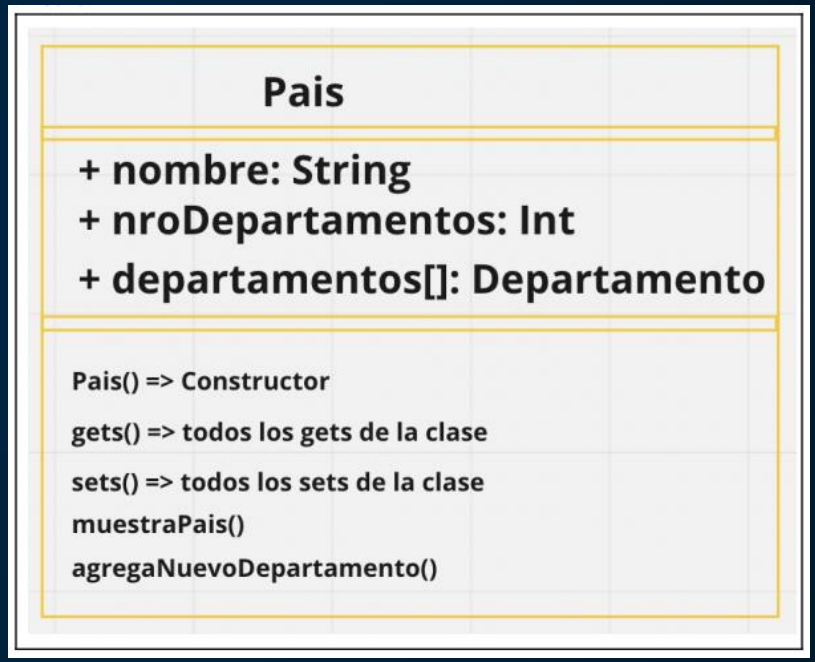
```
public void mostrarDepartamento() {
System.out.println("Nombre del Departamento: " + getNombre());
System.out.println("Nombre del Nro de Provincias: " + getNroProv());
for(int i=0;i<this.getProvincias().length; i++) {
provincias[i].mostrarProvincia();
}
}
public void agregarNuevaProvincia(Provincia nuevaProvincia) {
Provincia[] nueProvincia = new Provincia[provincias.length +1 ];
for(int i=0;i<provincias.length; i++) {
nueProvincia[i] = provincias[i];
}
nueProvincia[nueProvincia.length - 1] = nuevaProvincia;
setProvincias(nueProvincia);
setNroProv(nueProvincia.length);
}
}
```

PARTE PRACTICA

Pais.java

13 **13.Generar la clase País.**

- ○ Diseño.
- ○ Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)
  - ■ Crear una instancia de la clase País
  - ■ El constructor no recibe parámetros.
  - ■ Crear una instancia de la clase Departamento.
  - ■ Omitir el método agregaNuevoDepartamento()
  - ■ Mostrar los datos del País.
- ○ Adjuntar el código JAVA generado.



```java
package TareaH2;
public class Pais {
private int nroDeps;
private Departamento[] departamentos;
private String nombre;
public Pais() {
}
public int getNroDeps() {
return nroDeps;
}
public void setNroDeps(int nroDeps) {
this.nroDeps = nroDeps;
}
public Departamento[] getDepartamentos() {
return departamentos;
}
public void setDepartamentos(Departamento[] departamentos) {
this.departamentos = departamentos;
}
public String getNombre() {
return nombre;
}
public void setNombre(String nombre) {
this.nombre = nombre;
}
```

```
public void mostrarPais() {
System.out.println("Nombre del Pais: " + getNombre());
System.out.println("Nombre del Nro de Departamentos: " + getNroDeps());
for(int i=0;i<this.getDepartamentos().length; i++) {
departamentos[i].mostrarDepartamento();
}
}
public void agregarNuevoDepartamento(Departamento departamento) {
Departamento[] nuevoDepartamentos = new Departamento[departamentos.length
+ 1];
for (int i = 0; i < departamentos.length; i++) {
nuevoDepartamentos[i] = departamentos[i];
}
nuevoDepartamentos[nuevoDepartamentos.length - 1] = departamento;
setDepartamentos(nuevoDepartamentos);
setNroDeps(getNroDeps() + 1);
}
}
```

PARTE PRACTICA

main.java

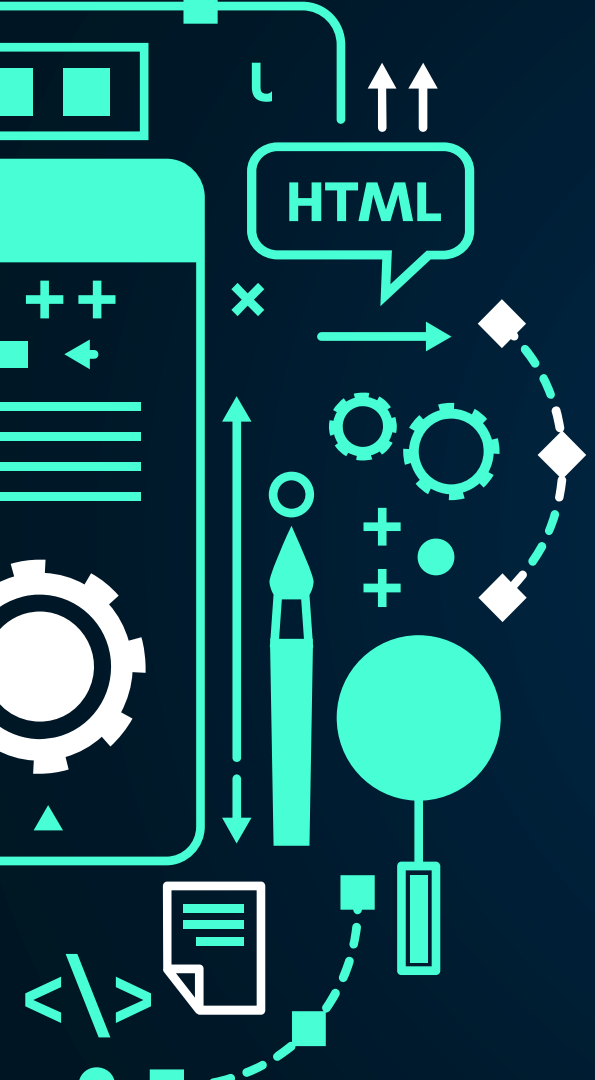
14 14.Crear el diseño completo de las clases.

- Diseño.
- Crear todos gets y sets de cada clase.
- Implementar los métodos agregarNuevoDepartamento(), agregarNuevaProvincia(), es decir todos los métodos.
- El método agregarNuevoDepartamento permite ingresar un nuevo departamento a un país.
- El método agregarNuevaProvincia permite ingresar una nueva provincia a un departamento.
- La clase Main debe mostrar lo siguiente:
  - Crear el PAÍS Bolivia
  - Al país Bolivia agregarle 3 departamentos.
  - Cada departamento deberá tener 2 provincias.
- Adjuntar el código JAVA generado

```java
package TareaH2;
public class main {
public static void main(String[] args) {
// TODO Auto-generated method stub
// Crear el país Bolivia
Provincia prov1 = new Provincia();
prov1.setNombre("Prov 1");
Provincia prov2 = new Provincia();
prov2.setNombre("Prov 2");
Provincia[] provlapaz = new Provincia[2];
provlapaz[0] = prov1;
provlapaz[1] = prov2;
Provincia prov3 = new Provincia();
prov3.setNombre("Prov 3");
Provincia prov4 = new Provincia();
prov4.setNombre("Prov 4");
Provincia[] provoruro = new Provincia[2];
provoruro[0] = prov3;
provoruro[1] = prov4;
Provincia prov5 = new Provincia();
prov5.setNombre("Prov 5");
Provincia prov6 = new Provincia();
prov6.setNombre("Prov 6");
Provincia[] provpando = new Provincia[2];
provpando[0] = prov5;
provpando[1] = prov6;
```

```java
Departamento dep1 = new Departamento();
dep1.setNombre("La Paz");
dep1.setNroProv(2);
dep1.setProvincias(provlapaz);
Departamento dep2 = new Departamento();
dep2.setNombre("Oruro");
dep2.setNroProv(2);
dep2.setProvincias(provoruro);
Departamento dep3 = new Departamento();
dep3.setNombre("Pando");
dep3.setNroProv(2);
dep3.setProvincias(provpando);
Departamento[] dep = new Departamento[3];
dep[0] = dep1;
dep[1] = dep2;
dep[2] = dep3;
Pais p = new Pais();
p.setNombre("Bolivia");
p.setNroDeps(3);
p.setDepartamentos(dep);
p.mostrarPais();
```

```
/*
*/
Departamento depinsert = new Departamento();
Provincia prov7 = new Provincia();
prov7.setNombre("Prov 4");
Provincia prov8 = new Provincia();
prov8.setNombre("Prov 4");
Provincia[] provTarija = new Provincia[2];
provTarija[0] = prov7;
provTarija[1] = prov8;
depinsert.setNombre("Tarija");
depinsert.setNroProv(2);
depinsert.setProvincias(provTarija);
depinsert.agregarNuevaProvincia(prov6);
System.out.println("\n\n");
depinsert.mostrarDepartamento();
Provincia[] provSC = new Provincia[0];
Departamento depnew = new Departamento();
depnew.setNombre("Santa Cruz");
depnew.setNroProv(0);
depnew.setProvincias(provSC);
p.agregarNuevoDepartamento(depnew);
System.out.println("\n\n");
p.mostrarPais();
}
}
```



THANKS!

ESTRUCTURA DE DATOS

Email: mijailchoque35@Gmail.com
GITHUB: https://github.com/choqueoliver35/ESTRUCTURA-DE-DATOS