

**Universidad Nacional de San Agustín**  
**Escuela Profesional de Ingeniería de Sistemas**  
**Fundamentos de Programación II**  
**Tema N° 13:**  
**POO – Agregación, Composición y Herencia III**

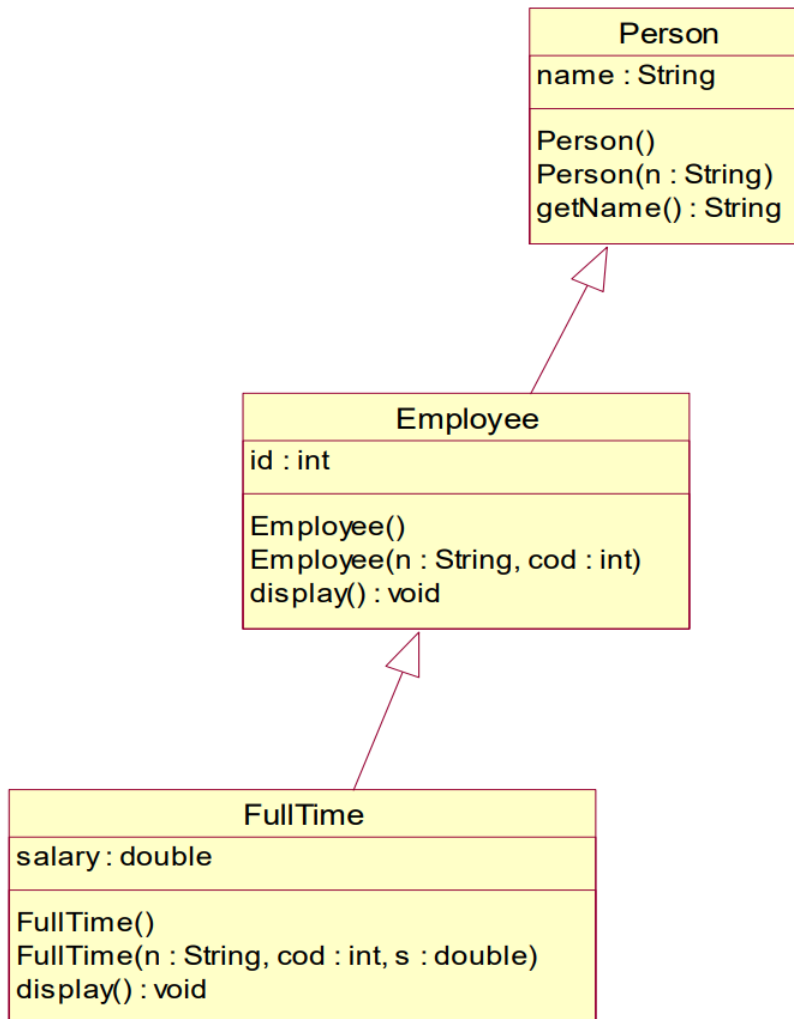
**Nombre:** Jhonatan Benjamin Mamani Céspedes

**CUI:** 20232188

**Link de GitHub:** <https://github.com/JBenjamin01/fp2-24b>

**Ejercicio 1:**

1. Completar la implementación de las clases y un main() que cree y use un objeto de cada clase: `FullTime`, `Employee` y `Person`, y que muestre los valores de sus atributos.



## Clase Person.java:

```
1  public class Person {
2      private String name;
3
4      // En el constructor vacío le estoy añadiendo un valor por defecto
5      public Person() {
6          this.name = "No tiene Nombre";
7      }
8
9      public Person(String n) {
10         this.name = n;
11     }
12
13     public String getName() {
14         return name;
15     }
16
17     public void display() {
18         System.out.println("Nombre: " + name);
19     }
20 }
```

## Clase Employee.java:

```
1  public class Employee extends Person {
2      private int id;
3
4      public Employee() {
5          super(); // Llamo al constructor por defecto de Person (para su nombre)
6          this.id = 0;
7      }
8
9      // Constructor con parámetros
10     public Employee(String n, int cod) {
11         super(n); // Aquí llamo al constructor de Person con parámetro
12         this.id = cod;
13     }
14
15     @Override
16     public void display() {
17         super.display(); // Uso también el método display() de Person
18         System.out.println("ID: " + id);
19     }
20 }
```

## Clase FullTime.java:

```
1  public class FullTime extends Employee {
2      private double salary;
3
4      public FullTime() {
5          super();
6          this.salary = 0.0;
7      }
8
9      // Constructor con parámetros
10     public FullTime(String n, int cod, double s) {
11         super(n, cod);
12         this.salary = s;
13     }
14
15     @Override
16     public void display() {
17         super.display();
18         System.out.println("Salario: " + salary + " soles");
19     }
20 }
```

## Clase Main.java:

```
1  public class Main {
2      public static void main(String[] args) {
3          // Objeto de la clase Person
4          Person p = new Person("Jhonatan");
5          System.out.println("Información del objeto Person:");
6          p.display();
7          System.out.println();
8
9          // Objeto de la clase Employee
10         Employee e = new Employee("Benjamin", 81556132);
11         System.out.println("Información del objeto Employee:");
12         e.display();
13         System.out.println();
14
15         // Objeto de la clase FullTime
16         FullTime f = new FullTime("Pepe", 56713418, 3502.70);
17         System.out.println("Información del objeto FullTime:");
18         f.display();
19     }
20 }
```

## Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad  
ad Nacional de San Agustín\2nd Year\Segundo Semestre\Fundame  
ming\Code\User\workspaceStorage\7345bf8761ac985a6bc8d0abeccd  
Información del objeto Person:  
Nombre: Jhonatan  
  
Información del objeto Employee:  
Nombre: Benjamin  
ID: 81556132  
  
Información del objeto FullTime:  
Nombre: Pepe  
ID: 56713418  
Salario: 3502.7 soles
```

## Diagrama UML:

