

**Universidad Nacional de San Agustín**  
**Escuela Profesional de Ingeniería de Sistemas**  
**Fundamentos de Programación II**  
**Tema N° 11:**  
**POO – Clases con Miembros de Clase**

**Nombre:** Jhonatan Benjamin Mamani Céspedes

**CUI:** 20232188

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

**Ejercicio 1:**

1. Crear el diagrama de clases de UML y el programa
2. Crear la clase Empleado con atributos:
  - ❖ nombre
  - ❖ apellido
  - ❖ edad
3. Además consignar un atributo de clase que vaya contando la cantidad de empleados que vayamos creando
4. En métodos: 2 constructores sobrecargados y los gets
5. Crear un main que cree varios objetos, que pruebe los gets y que muestre la cantidad de objetos creados

**Clase Empleado.java:**

```
1  public class Empleado {
2      private String nombre;
3      private String apellido;
4      private int edad;
5      private static int contadorEmpleados = 0;
6
7      public Empleado(String nombre, String apellido, int edad) {
8          this.nombre = nombre;
9          this.apellido = apellido;
10         this.edad = edad;
11         contadorEmpleados++;
12     }
13     public Empleado(String nombre, int edad) {
14         this.nombre = nombre;
15         this.apellido = "";
16         this.edad = edad;
17         contadorEmpleados++;
18     }
19     public String getNombre() {
20         return nombre;
21     }
22     public String getApellido() {
23         return apellido;
24     }
25     public int getEdad() {
26         return edad;
27     }
28     public static int getContadorEmpleados() {
29         return contadorEmpleados;
30     }
31 }
```

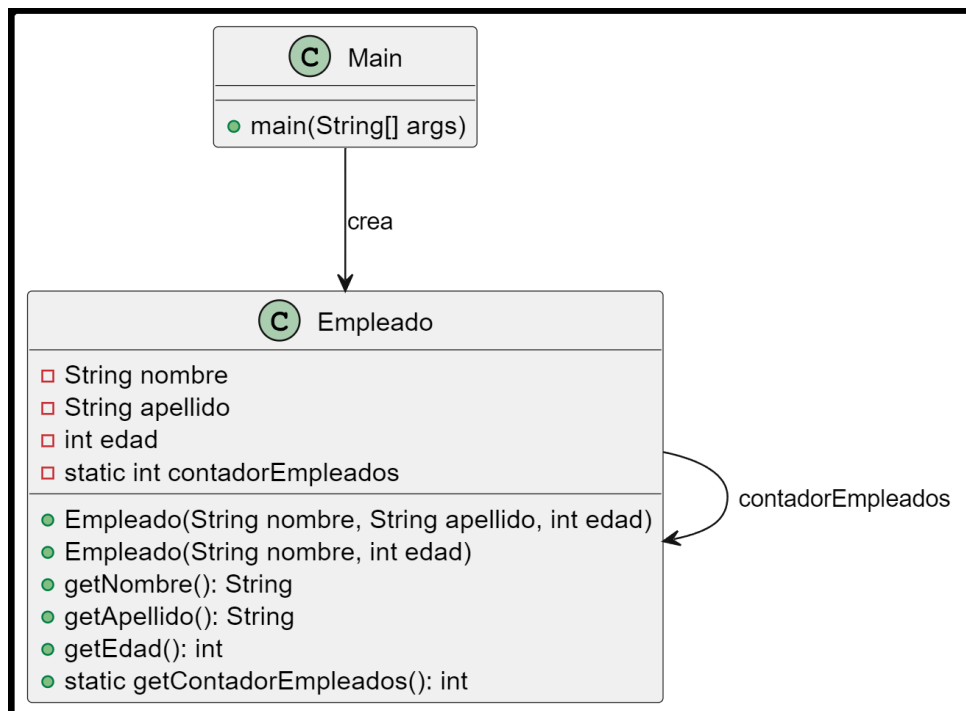
## Clase Main.java:

```
1 public class Main {
2     public static void main(String[] args) {
3         Empleado e1 = new Empleado("Juan", "Perez", 30);
4         Empleado e2 = new Empleado("Ana", "Gomez", 25);
5         Empleado e3 = new Empleado("Luis", 29);
6
7         System.out.println("Empleado 1: " + e1.getNombre() + " "
8                             + e1.getApellido() + ", Edad: " + e1.getEdad());
9         System.out.println("Empleado 2: " + e2.getNombre() + " "
10                            + e2.getApellido() + ", Edad: " + e2.getEdad());
11        System.out.println("Empleado 3: " + e3.getNombre() + ", Edad: " + e3.getEdad());
12
13        System.out.println("Total de empleados creados: " + Empleado.getContadorEmpleados());
14    }
15 }
```

## Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\Uni...
C:\Program Files\Java\jdk-21\bin\java.exe
ac08e3b1b\redhat.java\jdt_ws\Ejercicio 1_...
Empleado 1: Juan Perez, Edad: 30
Empleado 2: Ana Gomez, Edad: 25
Empleado 3: Luis, Edad: 29
Total de empleados creados: 3
```

## Diagrama UML:



## Ejercicio 2:

1. Diagrama de clases y programa de “jarra de monedas”
2. Cuando insertemos una moneda en cualquiera de las jarras, el programa deberá imprimir un mensaje “clink” e incrementar la cuenta de las monedas en la jarra e incrementar también la cantidad total de monedas en todas las jarras que tenga.
3. Cuando el número total de monedas exceda una meta fija preestablecida antes de iniciar el programa, el programa deberá imprimir “tiempo de compras”
4. Crear una aplicación para validar la clase, que permita crear 2 jarras, insertar algunas monedas a cada una usando un menú, imprimir el número de monedas por cada jarra y el número de monedas en total cada vez que se inserte una moneda

PennyJar
<u>+GOAL : int = 10000</u> <u>-allPennies : int = 0</u> <u>-pennies : int = 0</u>
<u>+addPenny() : void</u> <u>+getPennies() : int</u> <u>+getAllPennies() : int</u>

### Clase PennyJar.java:

```
1 public class PennyJar {
2     public static final int GOAL = 10000; // Esta es la meta fija para todas las jarras de mi clase
3     private static int allPennies = 0; // Cuenta total de monedas en todas las jarras
4     private int pennies = 0; // Monedas en la jarra actual
5
6     public void addPenny() {
7         pennies++; // Incrementa el número de monedas en esta jarra
8         allPennies++; // Incrementa el número total de monedas en todas las jarras
9         System.out.println("clink");
10
11         // Verifica si se alcanzó o excedió la meta
12         if (allPennies >= GOAL)
13             System.out.println("¡Tiempo de compras!");
14     }
15
16     // Método para obtener el número de monedas de la jarra actual
17     public int getPennies() {
18         return pennies;
19     }
20
21     // Método para obtener el número total de monedas en todas las jarras
22     public static int getAllPennies() {
23         return allPennies;
24     }
25 }
```

## Clase Main.java

```
1 import java.util.*;
2 public class Main {
3     // Lista para almacenar las jarras creadas
4     private static ArrayList<PennyJar> jarras = new ArrayList<>();
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         boolean exit = false;
9
10        while (!exit) {
11            System.out.println("\nMenú:");
12            System.out.println("1. Crear nueva jarra");
13            System.out.println("2. Insertar moneda en una jarra");
14            System.out.println("3. Ver número de monedas en una jarra específica");
15            System.out.println("4. Ver número total de monedas en todas las jarras");
16            System.out.println("5. Salir");
17            System.out.print("Seleccione una opción: ");
18
19            int option = sc.nextInt();
20
21            switch (option) {
22                case 1:
23                    PennyJar newJar = new PennyJar();
24                    jarras.add(newJar);
25                    System.out.println("¡Jarra creada! Ahora tienes " + jarras.size() + " jarras.");
26                    break;
27                case 2:
28                    if (jarras.isEmpty()) {
29                        System.out.println("No hay jarras disponibles. Crea una jarra primero.");
30                    } else {
31                        System.out.println("Elige una jarra (1 a " + jarras.size() + "): ");
32                        int jarIndex = sc.nextInt();
33                        if (jarIndex > 0 && jarIndex <= jarras.size()) {
34                            jarras.get(jarIndex - 1).addPenny();
35                        } else {
36                            System.out.println("Índice de jarra no válido.");
37                        }
38                    }
39                    break;
40                case 3:
41                    if (jarras.isEmpty()) {
42                        System.out.println("No hay jarras disponibles. Crea una jarra primero.");
43                    } else {
44                        System.out.println("Elige una jarra (1 a " + jarras.size() + "): ");
45                        int jarIndex = sc.nextInt();
46                        if (jarIndex > 0 && jarIndex <= jarras.size()) {
47                            System.out.println("Monedas en jarra " + jarIndex + ": " + jarras.get(jarIndex - 1).getPennies());
48                        } else {
49                            System.out.println("Índice de jarra no válido.");
50                        }
51                    }
52                    break;
53                case 4:
54                    System.out.println("Monedas totales en todas las jarras: " + PennyJar.getAllPennies());
55                    break;
56                case 5:
57                    exit = true;
58                    System.out.println("Saliendo del programa...");
59                    break;
60                default:
61                    System.out.println("Opción no válida.");
62                    break;
63            }
64        }
65    }
66 }
```

## Consola:

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San Agustín> .\a.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\jhona\AppData\Roaming\Java\jdk-11.0.2\bin\java.exe'
```

Menú:

1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir

Seleccione una opción: 1

¡Jarra creada! Ahora tienes 1 jarras.

Menú:

1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir

Seleccione una opción: 1

¡Jarra creada! Ahora tienes 2 jarras.

Menú:

1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir

Seleccione una opción: 1

¡Jarra creada! Ahora tienes 3 jarras.

Menú:

1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir

Seleccione una opción: 3

Elige una jarra (1 a 3):

1

Monedas en jarra 1: 0

Menú:

1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica

```
PS C:\Users\jhona\OneDrive\Documentos\University\Universidad Nacional de San Agustín>
```

2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir

Seleccione una opción: 4

Monedas totales en todas las jarras: 0

Menú:

1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir

Seleccione una opción: 2

Elige una jarra (1 a 3):

1

'clink'

```
Menú:
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: 2

Elige una jarra (1 a 3):

3

'clink'

Menú:

```
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: 4

Monedas totales en todas las jarras: 2

Menú:

```
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: █

```
PS C:\Users\jgonza\OneDrive\documentos\university\universidad nacional de san agustin
```

Menú:

```
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: 1

¡Jarra creada! Ahora tienes 4 jarras.

Menú:

```
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: 2

Elige una jarra (1 a 4):

4

'clink'

Menú:

```
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: 2

Elige una jarra (1 a 4):

4

'clink'

Menú:

```
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
```

Seleccione una opción: 2

Elige una jarra (1 a 4):

4

'clink'

```

PS C:\Users\jrona\OneDrive\documentos\University\Universidad Nacional de San Agustín
Seleccione una opción: 2
Elige una jarra (1 a 4):
4
'clink'

Menú:
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
Seleccione una opción: 2
Elige una jarra (1 a 4):
4
'clink'

Menú:
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
Seleccione una opción: 3
Elige una jarra (1 a 4):
4
Monedas en jarra 4: 3

Menú:
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
Seleccione una opción: 4
Monedas totales en todas las jarras: 5

Menú:
1. Crear nueva jarra
2. Insertar moneda en una jarra
3. Ver número de monedas en una jarra específica
4. Ver número total de monedas en todas las jarras
5. Salir
Seleccione una opción: 5
Saliendo del programa...

```

## Diagrama UML:

