UNIVERSIDAD TECNOLÓGICA DE PANAMÁ FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES LICENCIATURA EN CIBERSEGURIDAD

Programación I

Actividad #1

PREPARADO POR:

Angeline Urriola 4-834-1980

PROFESOR:

Napoleón Ibarra

GRUPO:

2S3111

FECHA:

20/08/2025

1.3 Construcción de clase (JAVA)

1. **¿Cuál es su concepto?**
   La construcción de una clase en Java se refiere al proceso de definir una clase, las cuales son una parte fundamental de la POO y son la base para la creación de objetos, ya que es un modelo que define la estructura y el comportamiento de los objetos que se pueden crear a partir de ella.
2. **Importancia y/o relevancia**
   - Ayudan a organizar mejor el programa, separando todo en partes claras y fáciles de manejar.
   - Las clases permiten reutilizar el código
   - Ayuda a mantener las especificaciones de acceso de las variables miembro.
3. **Ventajas y desventajas**
   ***Ventajas*:**
   - Se puede crear código modular y reutilizable
   - Fácil de mantener y ampliar con el tiempo
   - Mejoran la legibilidad

   ***Desventajas*:**
   - Si se crean muchos objetos y clases, el programa puede usar más memoria y procesador de lo que sería con un código más simple.
   - Para tareas sencillas, el uso de clases puede resultar en un código más complejo de lo que se necesita.
   - La creación de objetos puede consumir más memoria y recursos.
4. **Ejemplo de cada subpunto**

```
class Student {
        int id;
        String n;
                public Student(int id, String n) {
                this.id = id;
                this.n=n;
        }
}
```

1.3.1 Miembros de una clase

1. **¿Cuál es su concepto?**
   Los miembros de una clase son un conjunto de elementos que definen a los objetos (propiedades), así como los comportamientos o funciones que maneja el objeto.
2. **Importancia y/o relevancia**
   En Java, los miembros de una clase son esenciales para definir el comportamiento y las características de los objetos creados a partir de esa clase.

3. **Ventajas y desventajas**

***Ventajas*:**

- Los miembros de una clase ayudan a organizar el código, porque agrupan los datos y funciones relacionados.
- Mejoran la modularidad, haciendo que el programa sea más fácil de entender y mantener.
- Permiten proteger los datos mediante el encapsulamiento, decidiendo qué puede verse y qué no.

***Desventajas*:**

- Si se crean demasiados miembros, el código puede volverse confuso o difícil de manejar.
- Planear correctamente los miembros de una clase requiere tiempo antes de empezar a programar.
- El uso excesivo de miembros puede consumir más memoria y recursos de lo necesario en programas simples.

4. **Ejemplo de cada subpunto**

```
public class Perro
{
        private int Peso:
        public int GetPeso ()
        {
                return Peso:
        }
        public void SetPeso (int newPeso)
        {
                if (newPeso > 0)
    {
                Peso newPeso:;
        }
    }
}
```

1.3.2 Modificadores de acceso

1. **¿Cuál es su concepto?**
   Los modificadores de acceso son palabras que determinan quién puede ver o usar los elementos de una clase, como sus atributos y métodos. Básicamente, indican desde dónde se puede acceder a ellos y ayudan a proteger los datos siguiendo el principio de encapsulamiento.
2. **Importancia y/o relevancia**
   - Protegen los datos de la clase, evitando que alguien los cambie por error desde fuera.
   - Ayudan a mantener el código ordenado, dejando claro qué se puede usar y qué no.
   - Facilitan que el programa sea más seguro y fácil de mantener, porque se sigue el principio de encapsulamiento.
3. **Ventajas y desventajas**
   ***Ventajas*:**
   - Controlan quién puede usar los miembros de la clase, aumentando la seguridad del código.
   - Permiten mantener los datos y funciones organizados, facilitando su comprensión.
   - Ayudan a trabajar en equipo, porque cada clase puede exponer solo lo necesario sin afectar otras partes del programa.

   Desventajas:

   - Requieren planificación, ya que hay que decidir qué miembros serán públicos, privados o protegidos.
   - Pueden limitar el acceso demasiado, dificultando el uso de ciertos atributos o métodos si no se planifica bien.
   - Pueden confundir a los principiantes, especialmente cuando se combinan varios modificadores en un programa grande.
4. **Ejemplo de cada subpunto**

```
public class Persona {

   private String nombre;

   public int edad;


   public Persona(String n, int e) {

      nombre = n;

      edad = e;

   }
```

```
    public void mostrar() {
        System.out.println(nombre + ", " + edad);
    }
}

public class Main {
    public static void main(String[] args) {
        Persona p = new Persona("Angeline", 20);
        p.mostrar();  // solo se puede acceder al método público
    }
}
```

1.3.3 Otras opciones

Además de los miembros y los modificadores de acceso, existen otros elementos importantes en la construcción de clases en Java que permiten mejorar el diseño, la seguridad y el funcionamiento del código, como:

1. **Métodos getter y setter**

- **Concepto**: Métodos que permiten acceder (leer) o modificar (escribir) atributos privados de una clase.
- **Importancia**: Son esenciales para mantener el encapsulamiento, ya que controlan cómo se accede a los datos sin exponerlos directamente.
- **Ventajas/Desventajas**: Mejoran la seguridad y mantenibilidad del código, pero pueden aumentar su longitud si se usan en exceso.
- **Ejemplo**:

```
class Persona {
    private int edad;
    public void setEdad(int e) { edad = e; }
    public int getEdad() { return edad; }
}
```

```
public class Main {
    public static void main(String[] args) {
        Persona p = new Persona();
        p.setEdad(25);
        System.out.println(p.getEdad());
    }
}
```

Referencias

Ramuglia, G. (2024, February 20). Declaring and Constructing Java Classes: A How-TO Guide. Linux Dedicated Server Blog. https://ioflood.com/blog/java-classes/#:~:text=La%20importancia%20de%20las%20clases,organizaci%C3%B3n%20jer%C3%A1rquica%20de%20las%20clases.

GeeksforGeeks. (2025, August 5). Classes and objects in Java. GeeksforGeeks. https://www.geeksforgeeks.org/java/classes-objects-java/

Blasco, J. L. (2023, October 27). Introducción a POO en Java: Objetos y clases. OpenWebinars.net. https://openwebinars.net/blog/introduccion-a-poo-en-java-objetos-y-clases/#qu%C3%A9-son-las-clases-en-java

GeeksforGeeks. (2025a, July 11). Understanding classes and objects in Java. GeeksforGeeks. https://www.geeksforgeeks.org/java/understanding-classes-and-objects-in-java/

Concepts, M. (2023, March 17). Java Objects and Classes: Creating efficient and Modular code. https://www.linkedin.com/pulse/java-objects-classes-creating-efficient-modular-code#:~:text=By%20using%20classes%20and%20objects,easy%20to%20understand%20and%20manage.

Rodriguez, G. J. (2021, April 27). Programación orientada a objetos. Northware. https://www.northware.mx/blog/programacion-orientada-a-objetos/#:~:text=Los%20miembros%20de%20una%20clase,que%20est%C3%A1%20hecho%20un%20objeto.