El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

CICLO II:

Programación Básica en Java

MisionTIC 2022

El futuro digital es de todos | MinTIC

UNIVERSIDAD DEL NORTE
Vigilada Mineducación

Sesión 11:
Introducción a Java

Programación Orientada a Objetos (POO)

MisiónTIC2022

Objetivos de la sesión

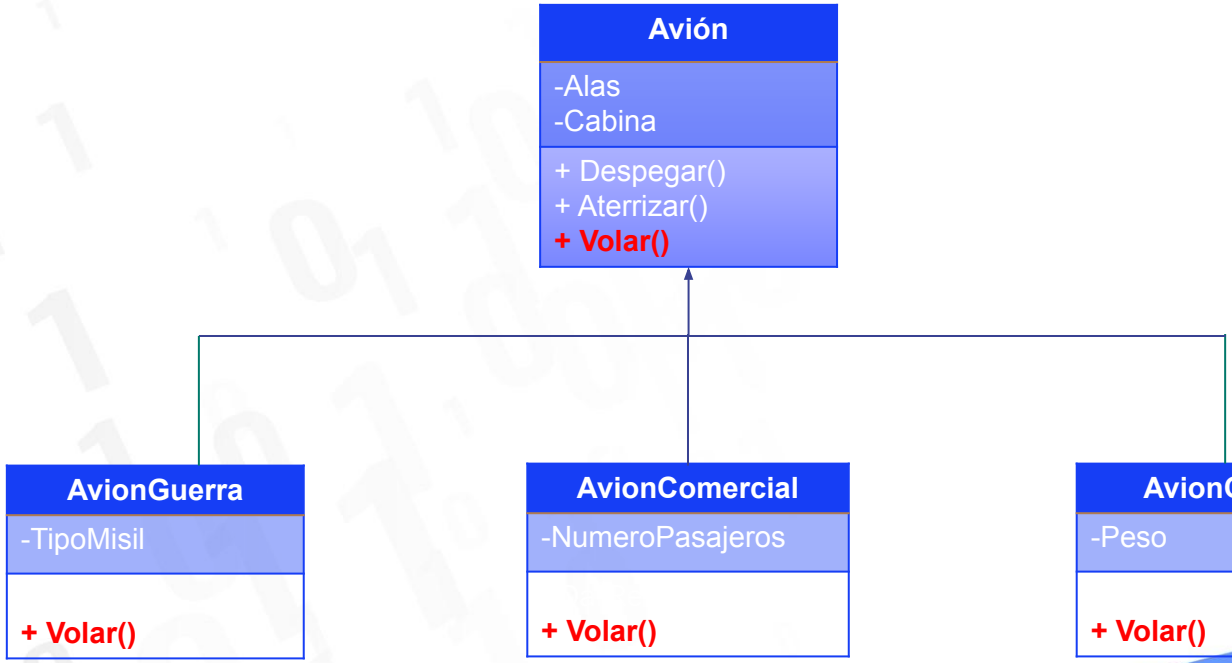
Al finalizar esta sesión estarás en capacidad de:

1. Explicar el polimorfismo básico y su relación con la herencia en Java
2. Explicar el polimorfismo utilizando los conceptos de clase abstracta y la sobreescritura de métodos (@OverEditor)
3. Construir programas que usen el polimorfismo básico con el apoyo de la herencia y con el uso de la clase abstracta y la sobreescritura de métodos

Polimorfismo

Volar

Polimorfismo



Polimorfismo

¿Qué es?

- Muchas Formas que puede tomar un objeto

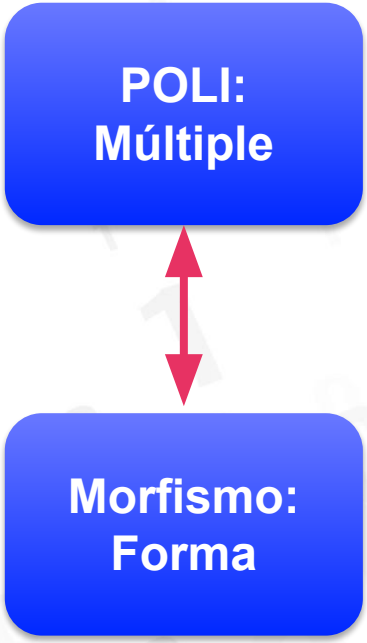
¿Por qué?

- Los objetos pueden tener distinto comportamiento dependiendo de su interacción

¿Para qué?

- Código limpio
- Ahorrar tiempo

Polimorfismo



- En java, el polimorfismo se refiere a la capacidad que tiene un objeto de comportarse de múltiples formas o de decidir qué método aplicar, dependiendo de la clase a la que pertenece. El polimorfismo en Java se implementa por medio de la herencia.
- Esto significa que dos clases que tengan un método con el mismo nombre y que reciban los mismo parámetros, ejecutarán acciones distintas.
- Una llamada a un método genérico de una Superclase ejecuta la implementación correspondiente del método dependiendo de la clase del objeto que se creó.

Polimorfismo - Ejemplo

Se tienen las clases **Entero** y **Char**, cada una responderá de manera diferente al método "**Sucesor**"

Polimorfismo – Ejemplo Java

```
Public class Avion {
  public void Volar() {
    System.out.println(“Con toda");
  }
}
Public class AvionComercial extends Avion {
  public void Volar() {
    System.out.println(“Con Pasajeros");
  }
}
Public class AvionCarga extends Avion {
  public void Volar() {
    System.out.println(“Con Productos");
  }
}
```

```
public static void main(String[ ] args) {
  Avion a = new Avion Comercial();
  Avion b = new AvionCarga();
  a.Volar();
  a.Volar();
}
```

Clases abstractas

- Cuando hay herencia se puede crear una clase abstracta.
- Se denomina **clase abstracta** a una clase que tiene algún método sin implementar.
- Para definir una clase abstracta se debe tener en cuenta:
    - En la declaración de la clase añadir abstract.
    - Dejar sólo la declaración del método, añadiendo igualmente abstract.
- Una subclase de una clase abstracta debe:
    - Implementar todos los métodos abstractos heredados, o bien
    - Ser a su vez declarada abstracta.

Clases Abstractas

- Una clase abstracta no puede ser instanciada
- Puede contener métodos abstractos, a ser implementados en subclases
- Puede contener métodos concretos

```
public abstract class Padre {
    ...
    public abstract void metodo();
}

public class Hija extends Padre {
    public void metodo() {
        ...
    }
}
```

Clase Padre es abstracta: si se intenta instanciarla, se produce un error de compilación

Si la clase Hija no provee una implementación del método metodo(), se produce un error de compilación

Clases Abstractas – Sobreescritura de métodos

- La sobrescritura tiene sentido si la subclase tiene características que hagan que ese método deba ejecutar acciones adicionales.
- Para sobrescribir se utiliza @Override justo encima de la definición del método.
- El método de la subclase debe tener la misma definición que el de la superclase.
- En caso de no sobrescribir un método, se ejecutará el de la superclase.

Clases abstractas, @Override – Polimorfismo - Ejemplo

```
//Clase padre
public abstract class Vehiculo {
        public abstract void abastecer(double litros);
}
//Clase hija
public class Taxi extends Vehiculo{
        public Taxi() {
        }
@Override
public void abastecer(double litros){
}
```

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

¡GRACIAS

POR SER PARTE DE ESTA EXPERIENCIA DE APRENDIZAJE!

Mision TIC 2022