UTE EC

EDUCACIÓN EN LÍNEA

Computación paralela y distribuida

Métodos por defecto(default) en interfaces

Integrantes:
Carlos Cadena
Esteban Perugachi
Jonathan Sánchez

Clase abstracta

- Es una clases que declara la existencia del método pero no su implementación.
- Puede contener métodos no abstractos pero al menos uno debe ser abstracto.
- No se pueden instanciar pero si se pueden heredar, lo que las clases hijas le darán la funcionalidad.

Clase abstracta

```
package ejemploabs;
public abstract class Figuras {
    int base,altura;
    public void IngresarDato(int x,int y){
        base=x;
        altura=y;
    }
    public void ImprimirDatos(){
        System.out.println("Base= " + base);
        System.out.println("Altura= "+altura);
    }
    public abstract float Area();// metodo ABSTRACTO
}
```

Implementación del método abstracto

```
package ejemploabs;
public class Rectangulo extends Figuras{
    @Override
    public float Area(){
        System.out.println("Rectangulo");
        return base*altura;
    }
}
```

Implementación del método abstracto

```
package ejemploabs;
public class Triangulo extends Figuras {
    @Override
    public float Area(){
        System.out.println("Triangulo");
        return (base*altura)/2;
    }
}
```

Clase principal

```
public static void main(String[] args) {
        System.out.println("Clases y metodos Abstractos");
        int x=3,y=10;
        //Rectangulo
        Rectangulo r= new Rectangulo();
        r.IngresarDato(x, y);
        System.out.println(r.Area());
        r.ImprimirDatos();
        //Triangulo
        Triangulo t= new Triangulo();
        t.IngresarDato(x, y);
        System.out.println(t.Area());
        t.ImprimirDatos();
    }
```

Interfaces

- Es una colección de métodos abstractos y propiedades constantes.
- Es un variante de la clase abstracta, pero no puede tener métodos que no sean abstractos
- Simula la herencia múltiple.
- Se pueden implementar varias interfaces en una clase.

Primera interfaz

```
package interfaz;

public interface Animal {
    public void comunicar();
    public void come();

}
```

Segunda Interfaz

```
package interfaz;


public interface Acuatico {
    public void nadar();
}
```

Herencia múltiple

```
package interfaz;
public class Delfin implements Animal,Acuatico {
    @Override
    public void comunicar() {
        System.out.println("Se comunica con Silvidos");
    }
    @Override
    public void come(){
        System.out.println("Se alimenta de peces");
    }
    @Override
    public void nadar() {
        System.out.println("El delfin nada");
    }
}
```

Método predeterminado (default)

- Por ejemplo, si varias clases como A, B, C y D implementan una interfaz X, entonces si añadimos un nuevo método a la interfaz X, tenemos que cambiar el código en todas las clases (A, B, C y D) que implementan esta interfaz. En este ejemplo sólo tenemos cuatro clases que implementan la interfaz que queremos cambiar pero imaginemos que si hay cientos de clases implementando una interfaz entonces sería casi imposible cambiar el código en todas esas clases.

Método predeterminado (default)

- Tenemos que saber que los métodos default de Java sólo se permiten introducir en interfaces, por lo tanto no existen en clases. Estos métodos, al igual que todos los demás en las interfaces, son de manera implícita públicos. Su principal diferencia es que no son abstractos como el resto y necesitan proporcionar una implementación para pasar la fase de compilación.
- La idea principal de estos métodos es añadir nuevas funcionalidades a nuestras interfaces manteniendo la compatibilidad con el código ya existente.

Estructura de un método default

- Debe comenzar por la palabra reservada default.
- Proporcionará una implementación al método.
- Debe de encontrarse dentro de una interfaz.

```
1 public interface InterfazNormal {
2     void metodoAbstracto();
3     default void metodoDefault() {
4         System.out.println("Hola");
5     }
6 }
```

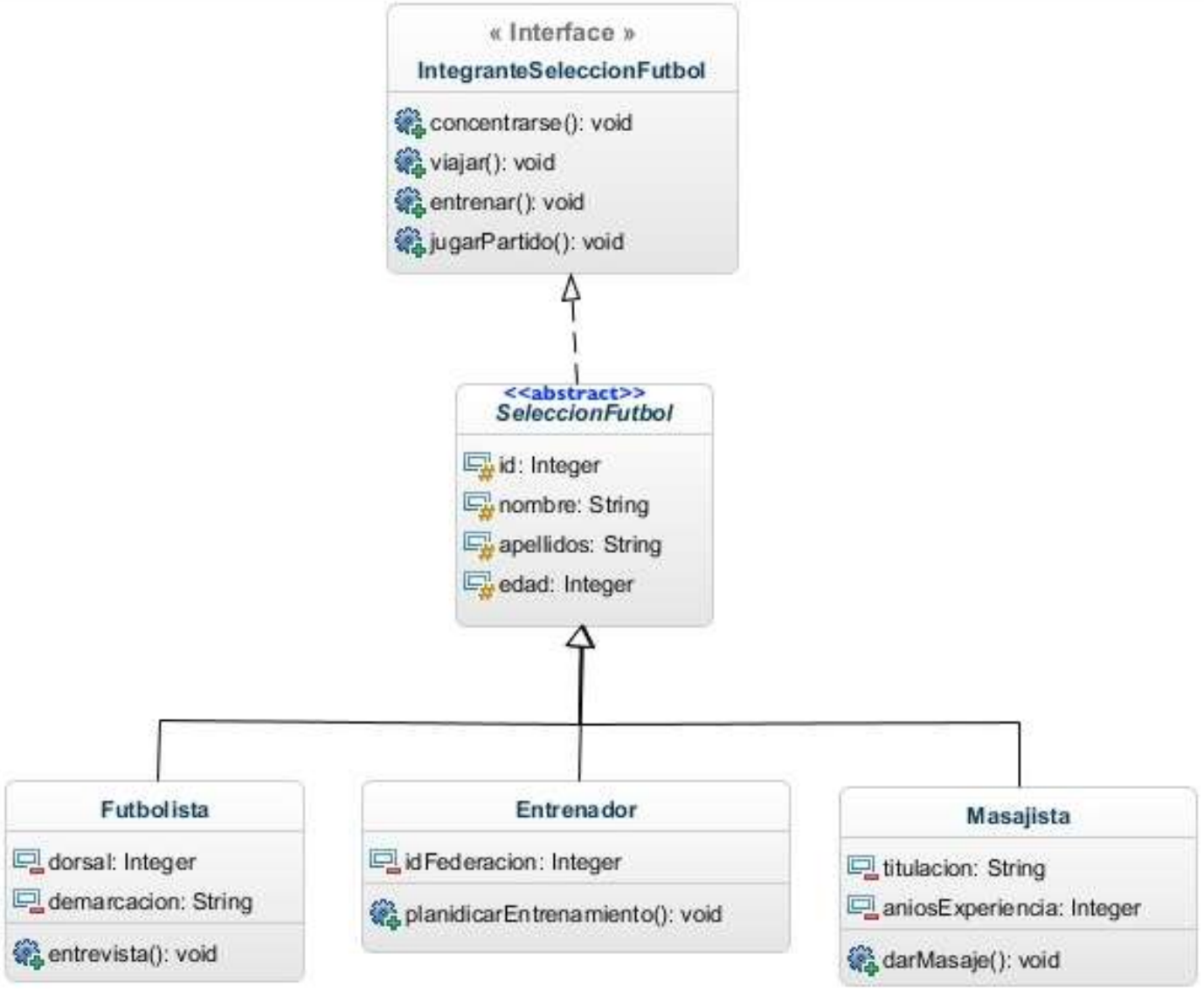
Método estático

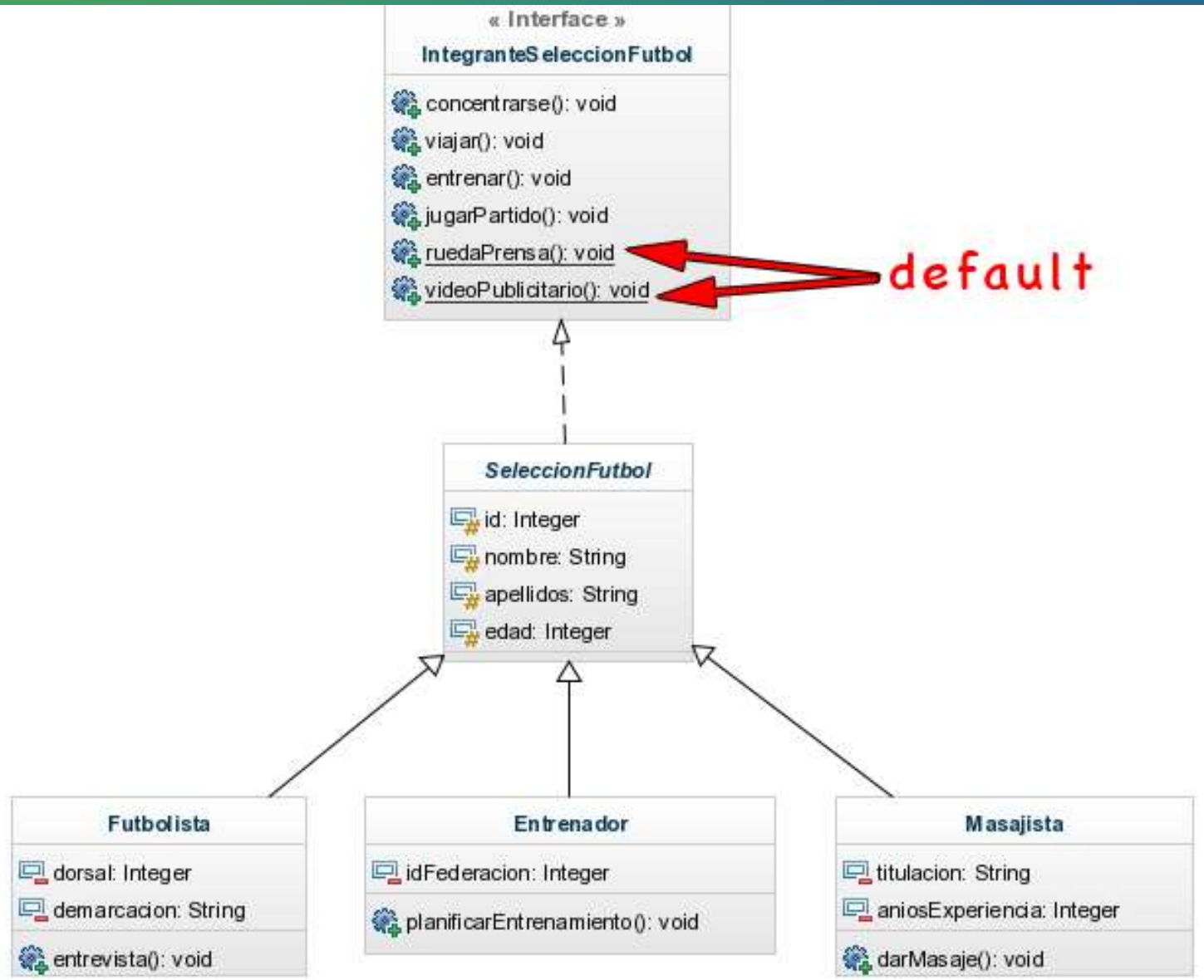
- Los métodos estáticos en la interfaz son similares al método por defecto, por lo que no es necesario implementarlos en las clases de implementación. Podemos añadirlos con seguridad a las interfaces existentes sin cambiar el código en las clases de implementación. Dado que estos métodos son estáticos, no podemos anularlos en las clases de implementación.

Estructura de un método estático

- Para crear un método estático se debe colocar la palabra clave static al principio de la firma del método.

```
1  public interface MyInterface {
2     static void staticMethod(){
3        System.out.println("This is a static method");
4     }
5  }
```


« Interface »
IntegranteSeleccionFutbol
concentrarse(): void
viajar(): void
entrenar(): void
jugarPartido(): void
<<abstract>>
SeleccionFutbol
id: Integer
nombre: String
apellidos: String
edad: Integer
Futbolista
dorsal: Integer
demarcacion: String
entrevista(): void
Entrenador
idFederacion: Integer
planidicarEntrenamiento(): void
Masajista
titulacion: String
aniosExperiencia: Integer
darMasaje(): void


« Interface »
IntegranteSeleccionFutbol
concentrarse(): void
viajar(): void
entrenar(): void
jugarPartido(): void
ruedaPrensa(): void
videoPublicitario(): void
default
SeleccionFutbol
id: Integer
nombre: String
apellidos: String
edad: Integer
Futbolista
dorsal: Integer
demarcacion: String
entrevista(): void
Entrenador
idFederacion: Integer
planificarEntrenamiento(): void
Masajista
titulacion: String
aniosExperiencia: Integer
darMasaje(): void

```java
public interface IntegranteSeleccionFutbol {

        void concentrarse();

        void viajar();

        void entrenar();

        void jugarPartido();

        default void ruedaPrensa(){
                System.out.println(" da una rueda de prensa (desde la Interface)");
        }

        default void videoPublicitario (){
                System.out.println(" hacer un video publicitario (desde la Interface)");
        }

}
```

```java
// ArrayList de objetos SeleccionFutbol. Idenpendientemente de la clase hija a la que pertenezca el objeto
public static ArrayList integrantes = new ArrayList();

public static void main(String[] args) {

        SeleccionFutbol delBosque = new Entrenador(1, "Vicente", "Del Bosque", 60, 28489);
        SeleccionFutbol iniesta = new Futbolista(2, "Andres", "Iniesta", 29, 6, "Interior Derecho");
        SeleccionFutbol raulMartinez = new Masajista(3, "Raúl", "Martinez", 41, "Licenciado en Fisioterapia", 18);

        integrantes.add(delBosque);
        integrantes.add(iniesta);
        integrantes.add(raulMartinez);

        // RUEDA DE PRENSA CON EL MÉTODO DEFAULT
        System.out.println("Todos los integrantes dan una rueda de prensa. (Todos ejecutan el mismo método)");
        for (SeleccionFutbol integrante : integrantes) {
                System.out.print(integrante.getNombre() + " " + integrante.getApellidos() + " -> ");
                integrante.ruedaPrensa();
        }

        // VIDEO PUBLICITARIO CON EL MÉTODO DEFAULT
        System.out.println("Todos los integrantes graban un video publicitario. (Todos ejecutan el mismo método)");
        for (SeleccionFutbol integrante : integrantes) {
                System.out.print(integrante.getNombre() + " " + integrante.getApellidos() + " -> ");
                integrante.videoPublicitario();
        }
```

```
Todos los integrantes dan una rueda de prensa. (Todos ejecutan el mismo método)
Vicente Del Bosque ->  da una rueda de prensa (desde la Interface)
Andres Iniesta ->  da una rueda de prensa (desde la Interface)
Raúl Martinez ->  da una rueda de prensa (desde la Interface)

Todos los integrantes graban un video publicitario. (Todos ejecutan el mismo método)
Vicente Del Bosque ->  hacer un video publicitario (desde la Interface)
Andres Iniesta ->  hacer un video publicitario (desde la Interface)
Raúl Martinez ->  hacer un video publicitario (desde la Interface)
```

Bilbiografía:

- E. (s. f.). 4.5 Interfaces | Curso de IntroducciÃ3n a Java. (c) www.exes.es. Recuperado 28 de octubre de 2020, de https://mundojava.net/interfaces.html?Pg=java_inicial_4_5.html.
- colaboradores de Wikipedia. (2020, 13 mayo). Interfaz (Java). Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Interfaz_(Java)#:%7E:text=Una%20interfaz%20en%20Java%20es,del%20comportamiento%20de%20los%20m%C3%A9todos .
- J. (2019, 13 enero). Métodos default de Java 8. Explicación y Ejemplos. Blog de Juanla. https://juanjavierrg.com/metodos-default-de-java-8/.

UTE EC

¡GRACIAS!

TRAS CENDE MOS