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 10:

Introducción a Java

Programación Orientada a Objetos (POO)

MisiónTIC2022

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Explicar el desarrollo de una clase padre con varios constructores con parámetros y el constructor por defecto
2. Explicar el desarrollo de un constructor a partir de la clase padre con super sin paso de parámetros y con paso de parámetros
3. Construir programas en Java con la aplicación de los conceptos de herencia aplicando el método super
4. Sobreescribir métodos heredados en la clase hija utilizando el método super

Clases con dos o más constructores

```
public class Vehiculo {
    private String matricula;
    private int potencia;

    public Vehiculo (String nombreMatricula) {   //CONSTRUCTOR 1
       matricula = nombreMatricula;
       potencia = 0;
    }

    public Vehiculo () {   //CONSTRUCTOR2
       matricula = "";
       potencia = 0;
    }

    public String getMatricula () {
         return matricula;
    }  //Cierre del método
} //Cierre de la clase
```

Clases con dos o más constructores

- Se ha definido la clase Vehículo, que permite crear objetos de tipo Vehículo.
- Todo objeto de tipo Vehículo estará definido por dos atributos: matricula (tipo String) y potencia (tipo entero), y admitirá un método: getMatricula().
- Al ejecutar nombreDelObjeto.getMatricula() se obtendrá el atributo correspondiente.

Clases con dos o más constructores

La clase tiene dos constructores. Lo que significa que se podrán crear Vehículos de dos maneras diferentes:

a) Vehículos que se creen con el constructor 1: habrá de indicarse, además del nombre del objeto, el parámetro que transmite el valor de la matrícula.

b) Vehículos que se creen con el constructor 2: no requieren parámetros para su creación y se inicializan a unos valores por defecto (matrícula cadena vacía y potencia cero).

Clases con dos o más constructores

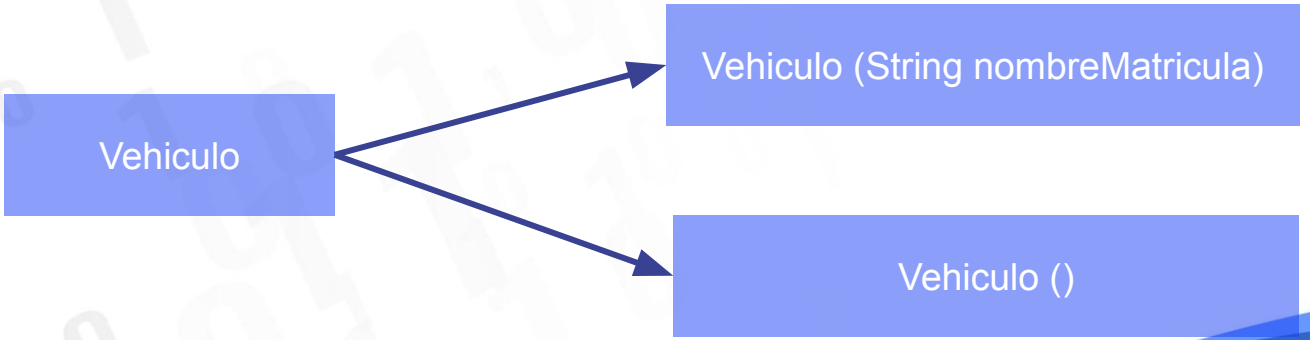
- Cuando más de un constructor o método tienen el mismo nombre pero distintos parámetros se dice que el constructor o método está sobrecargado.

- La sobrecarga de constructores o métodos permite llevar a cabo una tarea de distintas maneras (por ejemplo crear un objeto Vehículo con un nombre ya establecido o crearlo sin nombre establecido).

Sobrecarga (Overloading) de constructores

- La sobrecarga de constructor es una técnica en Java en la que una clase puede tener cualquier cantidad de constructores que sean diferentes en la lista de parámetros.

- El compilador diferencia estos constructores teniendo en cuenta el número de parámetros en la lista y su tipo.



Palabra clave super: llamar métodos de superclases desde subclases

- La palabra clave **super** se puede usar para acceder al constructor de la superclase.
- ”**super**” puede llamar constructores tanto con parámetros como sin parámetros dependiendo de la situación.

Palabra clave super: llamar métodos de superclases desde subclases – Ejemplo_1

```
/* superclase Vehiculo */
Public class Vehiculo
{
    public Vehiculo();
    {
        System.out.println("Constructor de la clase Vehiculo");
    }
}
/* subclase Taxi extiende de la clase Vehiculo */
Public class Taxi extends Vehiculo
{
    public Taxi()
    {
        super(); // llama al constructor de la superclase
        System.out.println("Constructor de la clase Taxi");
    }
}
```

Palabra clave super: llamar métodos de superclases desde subclases – Ejemplo_2

```
Public class Vehiculo {
    private String matricula;
    public Vehiculo(String matricula) {
        this.matricula = matricula;
    }
    public String getMatricula() {
        return matricula;
    }
    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }
}
Public class Taxi extends Vehiculo {
    private int numsillas;
    public Taxi(String matricula, int numsillas)
    {
        super(Matricula); // llama al constructor de la superclase con parámetros
        this.numsillas = numsillas;  // variable  propia de  la clase taxi
    }
}
```

Palabra clave super: llamar métodos de superclases desde subclases

- La llamada a super() debe ser la primera instrucción en el constructor de la subclase (Taxi en el ejemplo).
- Si un constructor de una subclase no llama explícitamente un constructor de la superclase, el compilador de Java inserta automáticamente una llamada al constructor sin argumento de la superclase.
- Si la superclase no tiene un constructor sin argumentos, obtendrá un error en tiempo de compilación.

Herencia – Superclases y Subclases Sintaxis

La sintaxis para declarar subclases es:

```
public class SubClase extends SuperClase{
    ...
}
```

Sobrescritura

- Modificación de los elementos de la superclase dentro de la subclase.
- La subclase puede definir:
  - Un atributo con el mismo nombre que uno de la superclase (Ocultación de atributos)
  - Un método con el mismo nombre que uno de la superclase (Redefinición de métodos)
- La aplicación más común de la sobreescritura es cuando se reescriba un método.

Sobrescritura - Sintaxis

```
Public class ClaseA
{
   void miMetodo(int var1, int var2)
   {
        ...
   }
   String miOtroMetodo( )
   {
        ...
   }
}
Public class ClaseB extends ClaseA
{   /* Estos métodos sobreescriben a los métodos de la superclase */
   void miMetodo (int var1 ,int var2)
   { ... }

   String miOtroMetodo( )
   { ... }
}
```

Sobrescritura de métodos heredados Ejemplo

```
Public class Padre
{
    int a = 100;
    void Mostrar()
    {
        System.out.println(a);
    }
}
Public class Hija extends Padre
{
    int a = 200;
    void Mostrar()
    {
         super.Mostrar(); // Llamar método Show desde una clase base
         System.out.println(a);
    }
    public static void Main(String[] args)
    {
        Hija miHija = new Hija();
        miHija.Mostrar();
    }
}
```

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

Ejercicios para practicar

Mision TIC 2022

El futuro digital es de todos

MinTIC

UNIVERSIDAD DEL NORTE

Vigilada Mineducación

¡GRACIAS

POR SER PARTE DE ESTA EXPERIENCIA DE APRENDIZAJE!

Mision TIC 2022