

NOMBRE: KEVIN SHAGÑAY

NRC: 17087

Actividad de aprendizaje Contacto Docente n.º 2

- **Tema de la actividad:**
 - Refuerzo de conceptos generales
- **Tipo de actividad:**
 - Resolución de ejercicios
- **Descripción de la actividad:**
 - Resolver los siguientes códigos en el lenguaje de programación JAVA
- **Formato de entrega del archivo**
 - (WORD, PDF, ZIP).
- **Nombre del archivo**
 - Nombre_Apellido_NRC_CD2
- **Criterios de evaluación (promedio de los ejercicios)**
 - UML (5)
 - Atributos bien definidos conforme lo solicitado (5)
 - Métodos bien utilizados conforme a los Atributos (5)
 - Main realizando lo solicitado (5)
 - Resumen indicando lo siguiente. **(ESTE RESUMEN SINO LO CONTIENE NO SE REVISARÁ EL TRABAJO Y LA CALIFICACIÓN SERÁ DE 0 PUNTOS SIN DERECHO A RECLAMO)**
 - Los atributos tienen el acceso x (público, privado, protegido, default), debido que en el ejercicio o en la resolución planteada y diseñada por el UML se indica, el constructor utilizado es el siguiente por los siguientes motivos:
 - 1
 - 2
 - 3

Para el ingreso de la información se lo realiza en la línea número #, y se lo realiza de la siguiente manera (ejemplo:
Persona persona1 = new Persona ();
Persona1.setNombre();) debido que el atributo nombre su acceso es (público, privado, protegido).
El proceso realizado luego del ingreso de los datos es el siguiente

Extensión máxima

3 de Junio del 2024

EJERCICIOS A RESOLVER

Ejercicio 1: Clase Persona y Clase Estudiante

Enunciado:

Crea una clase base llamada Persona con los siguientes atributos públicos:

- +nombre (tipo String)
- +edad (tipo int)

La clase Persona debe tener los siguientes métodos:

- presentarse(): que imprima un mensaje indicando el nombre y la edad de la persona.

Luego, crea una clase derivada llamada Estudiante que herede de Persona y añada los siguientes atributos públicos:

- ■ matricula (tipo String)
- +carrera (tipo String)

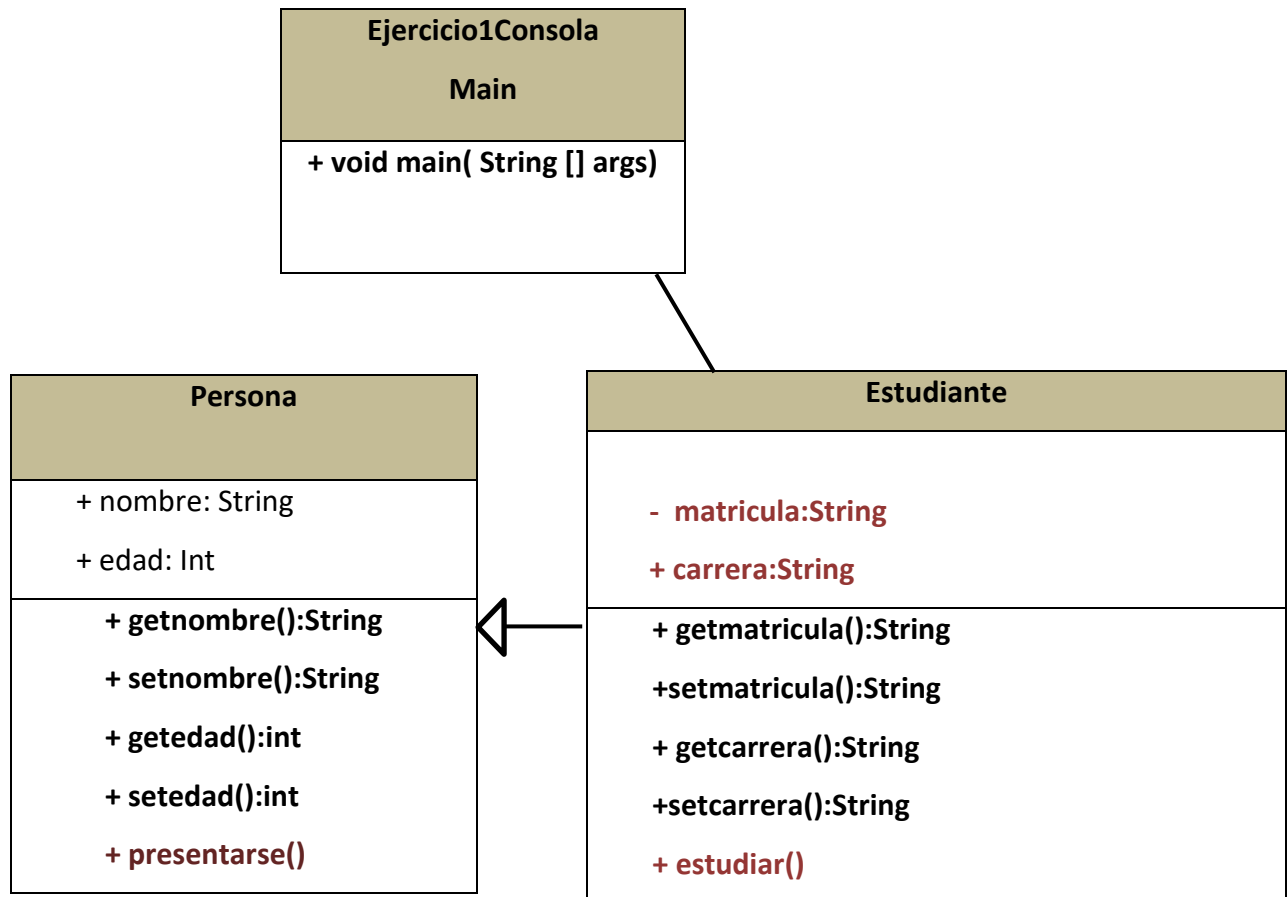
La clase Estudiante debe tener los siguientes métodos:

- +estudiar(): que imprima un mensaje indicando que el estudiante está estudiando.

Instancia un objeto de tipo Estudiante y ejecuta ambos métodos, presentarse() y estudiar().

RESOLUCION EJERCICIO 1

UML



CODIGO

Pantalla

```

--- exec:3.1.0:exec (default-cli) @ Ejercicio1Consola ---
Introduce nombre:
kevin
Introduce edad:
29
Introduce matricula:
C001
Introduce carrera:
TICS
Presentacion:
Nombre:kevin edad:29

El estudiante: kevin esta estudiando
-----
BUILD SUCCESS
    
```

Clase Persona

```
12 public class Persona {
13     String nombre;
14     int edad;
15     //Get y set
16     public String getnombre()
17     {
18         return nombre;
19     }
20     public void setnombre(String nombre)
21     {
22         this.nombre = nombre;
23     }
24     public int getedad()
25     {
26         return edad;
27     }
28     public void setedad(int edad)
29     {
30         this.edad = edad;
31     }
32
33     //metodo
34     public void presentarse() {
35         System.out.println("Presentacion:");
36         System.out.println("Nombre:"+getnombre()+" edad:"+getedad());
37     }
38 }
```

```
public class Persona {
    String nombre;
    int edad;

    //Get y set
    public String getnombre()
    {
        return nombre;
    }
    public void setnombre(String nombre)
    {
        this.nombre = nombre;
    }
    public int getedad()
    {
        return edad;
    }
    public void setedad(int edad)
    {
        this.edad = edad;
    }

    //metodo
    public void presentarse(){
        System.out.println("Presentacion:");
        System.out.println("Nombre:"+getnombre()+"
edad:"+getedad()); }
}
```



Clase Estudiante

```
11 public class Estudiante extends Persona {
12     private String matricula;
13     String carrera;
14
15     //Get y Set
16
17     public void setmatricula(String matricula) {
18         this.matricula = matricula;
19     }
20     public String getmatricula() {
21         return matricula;
22     }
23     public void setcarrera(String carrera) {
24         this.carrera = carrera;
25     }
26     public String getdarrera() {
27         return carrera;
28     }
29
30     public void estudiar(){
31         System.out.println("");
32         System.out.println("El estudiante: "+getnombre()+" esta estudiando" );
33     }
34 }
```

```
public class Estudiante extends Persona {
    private String matricula;
    String carrera;

    //Get y Set

    public void setmatricula(String matricula) {
        this.matricula = matricula;
    }
    public String getmatricula() {
        return matricula;
    }
    public void setcarrera(String carrera) {
        this.carrera = carrera;
    }
    public String getdarrera() {
        return carrera;
    }

    public void estudiar(){
        System.out.println("");
        System.out.println("El estudiante: "+getnombre()+" esta estudiando" );
    }
}
```

Clase Principal

```

5  package com.mycompany.ejercicio1consola;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author Kevin
12   */
13  public class Ejercicio1Consola {
14      public static void main(String[] args) {
15          String nombre, matricula, carrera;
16          int edad;
17          Scanner dato= new Scanner (System.in);
18
19          System.out.println("Introduce nombre:");
20          nombre =dato.nextLine();
21          System.out.println("Introduce edad:");
22          edad =dato.nextInt();
23
24          dato.nextLine();
25          System.out.println("Introduce matricula:");
26          matricula =dato.nextLine();
27          System.out.println("Introduce carrera:");
28          carrera =dato.nextLine();
29
30          Estudiante E1 = new Estudiante();
31          E1.setnombre(nombre);
32          E1.setedad(edad);
33          E1.setmatricula(matricula);
34          E1.setcarrera(carrera);
35
36          E1.presentarse();
37          E1.estudiar();
38
39      }
40  }
41
42

```

```

import java.util.Scanner;

/**
 *
 * @author Kevin
 */
public class Ejercicio1Consola {
    public static void main(String[] args) {
        String nombre, matricula, carrera;
        int edad;
        Scanner dato= new Scanner (System.in);

        System.out.println("Introduce nombre:");
        nombre =dato.nextLine();
        System.out.println("Introduce edad:");
        edad =dato.nextInt();

        dato.nextLine();
        System.out.println("Introduce matricula:");
        matricula =dato.nextLine();
        System.out.println("Introduce carrera:");
        carrera =dato.nextLine();

        Estudiante E1 = new Estudiante();
        E1.setnombre(nombre);
        E1.setedad(edad);
        E1.setmatricula(matricula);
        E1.setcarrera(carrera);

        E1.presentarse();
        E1.estudiar();

    }
}

```

RESUMEN

UML

El diagrama de clases nos ayudan a entender cómo funciona el programa, cómo se comporta y cómo se relacionan las clases, (la herencia entre la clase Estudiante y Persona).

CLASE PERSONA

Esta clase empezamos creando una clase denominada Persona en donde los atributos tienen el acceso público debido a que en el enunciado del ejercicio nos plantea que sean de acceso público.

El metodo utilizado es el siguiente:

Metodos get y set

Motivo:

- 1: con estos metodos nosotros podemos escribir y leer nuestras variables ingresadas por el teclado
- 2: En los set nosotros guardamos la informacion y con los get nosotros podemos recuperar los valores
- 3: Con estos metodos nosotros podremos tener un mejor control a la hora de manejar nuestro codigo

Ejemplo nosotros ingresamos dato del nombre del estudiante por teclado y con el set nosotros guardamos esa informacion para despues imprimirla.

Métodos:

En la clase Persona creamos el metodo: `presentarse()`, para que dentro de los bloques de código reciban los parámetros (variables) con el fin de proporcionar la información para que el método pueda realizar e imprimir su tareas en la programación (estos métodos son llamados desde el Main).” **En este metodo nosotros obtener los datos mediante el metodo get de cada variable “.**

CLASE ESTUDIANTE (Clase heredada)

Creamos una clase (Estudiante) en donde el atributo (matricula) es de acceso privado, y el atributo (carrera) es de acceso publico, debido a que en el enunciado del ejercicio nos plantea que sean de acceso público.

Para que una clase herede los atributos y metodos de otra clase, tenemos que agregar la palabra **extends** en: “public class Estudiante extends Persona”, con eso demostramos que es una clase hija, en esta clase los atributos de Persona (nombre y edad), vienen a formar parte de la clase Estudiante.

El metodo utilizado es el siguiente:

Metodos get y set para obtener la informacion ingresada por teclado.

Métodos:

En la clase Estudiante creamos el método: estudiar (), para que dentro de los bloques de código reciban los parámetros (variables) con el fin de proporcionar la información para que el método pueda realizar e imprimir sus tareas en la programación (estos métodos son llamados desde el Main).” **En este metodo nosotros obtener los datos mediante el metodo get de cada variable”**

CLASE MAIN:

En la clase principal(Main) llamamos a la librería Scanner, esta nos permite la entrada de datos por teclado.

- En la linea #15 y #16 declaramos variable de tipo String y Int, estas variables nos ayudaran a guardar la informacion que ingresemos por teclado
- En la linea #17 creamos un objeto de la clase scanner , esto nos permitira capturar la entrada de los datos.
- En la linea #20 empezamos a guardar los datos ejemplo: nombre =dato.nextLine(), en el objeto nombre que es un String guardaremos los datos ingresados por teclado y capturamos con el objeto “dato ” creado desde la librería scanner, y asi realizamos el mismo procesamiento con las demas variables.

- En la línea #30 instanciamos un objeto de la clase Estudiante: Estudiante E1 = new Estudiante() que no retorna ningún valor ya que vamos a capturarlos desde el método set.
- En la línea 31 con el objeto creado E1. Asignamos el valor ingresado por teclado ejemplo: E1.setnombre(nombre); esto nos permite capturar un nombre escrito por el teclado, de igual manera lo hacemos con todas las variables creadas.

Para luego en la línea #36 ejecutar al objeto E1 con el método presentarse(), en donde podremos imprimir en pantalla un mensaje indicando el nombre y la edad de la persona..

Además en la línea #37 utilizamos el objeto E1 para llamar la función estudiar(), y cuando ejecutemos nos imprima en pantalla un mensaje indicando que el estudiante está estudiando...y con eso finalizamos el primer ejercicio de herencia.

Ejercicio 2: Clase Vehículo y Clase Coche

Enunciado:

Crea una clase base llamada Vehículo con los siguientes atributos públicos:

- marca (tipo String)
- modelo (tipo String)

La clase Vehículo debe tener los siguientes métodos:

- detalles(): que imprima un mensaje con la marca y el modelo del vehículo.

Luego, crea una clase derivada llamada Coche que herede de Vehículo y añada el siguiente atributo público:

- ■ numPuertas (tipo int)

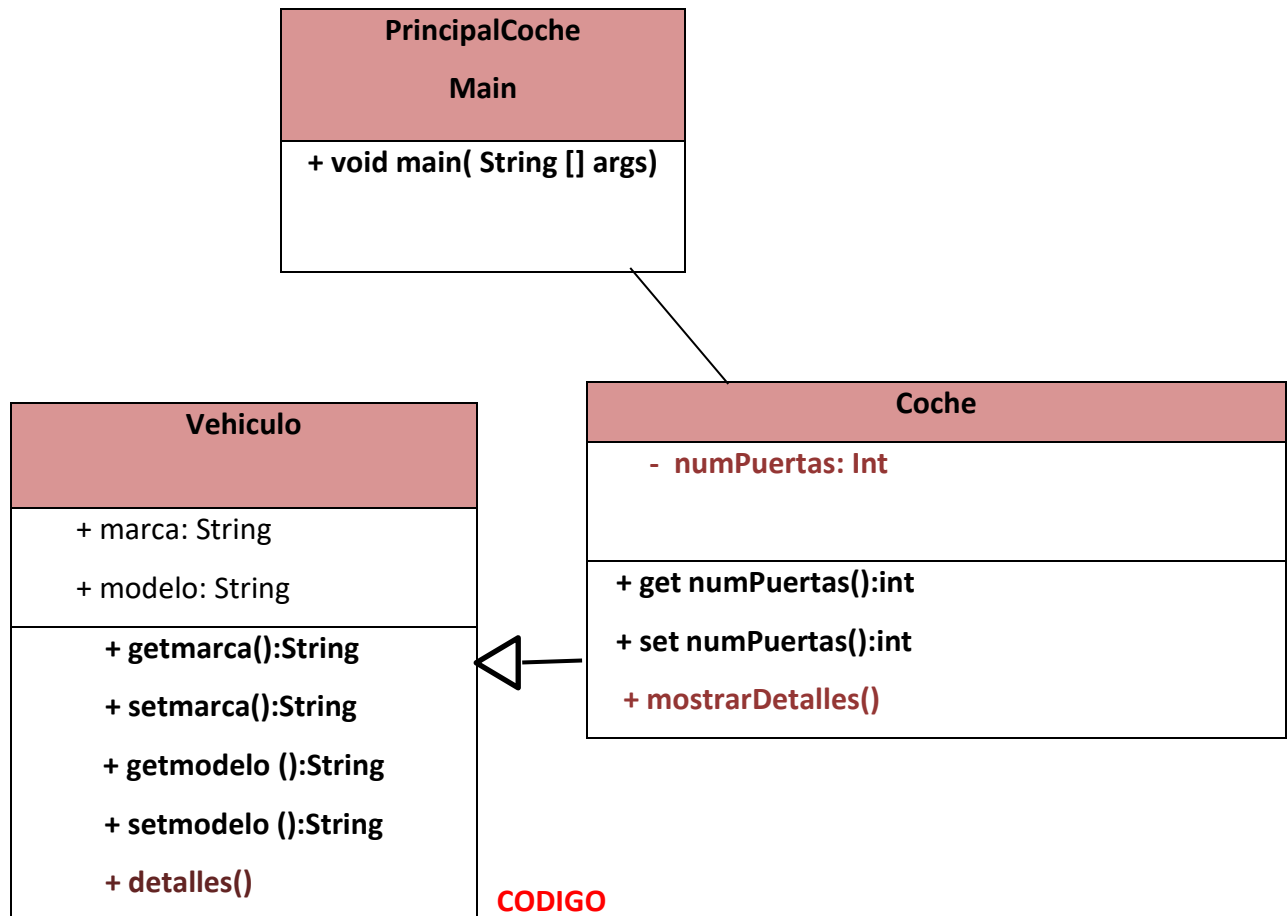
La clase Coche debe tener los siguientes métodos:

- mostrarDetalles(): que imprima los detalles del coche incluyendo el número de puertas.

Instancia un objeto de tipo Coche y ejecuta ambos métodos, detalles() y mostrarDetalles().

RESOLUCION EJERCICIO 2

UML



CODIGO

Pantalla

```

--- exec:3.1.0:exec (default-cli) @ Principal_Coche ---
Introduce la marca del carro:
Kia
Introduce el modelo del carro:
Picanto
Introduce el numero de puertas:
5
Metodo que solo muestra marca y modelo:
Marca:Kia
Modelo:Picanto
Metodo que muestra detalles completos del coche
Marca:Kia
Modelo:Picanto
Numero de puertas: 5
-----
BUILD SUCCESS
    
```



Clase Vehiculo

```
public class Vehiculo {  
    String marca;  
    String modelo;  
  
    // Get y Set  
    public String getmarca()  
    {  
        return marca;  
    }  
    public void setmarca(String marca)  
    {  
        this.marca = marca;  
    }  
    public String getmodelo()  
    {  
        return modelo;  
    }  
    public void setmodelo(String modelo)  
    {  
        this.modelo = modelo;  
    }  
  
    //metodo  
    public void detalles(){  
        System.out.println("Marca:"+getmarca()+"\nModelo:"+getmodelo() );  
    }  
}
```

```
public class Vehiculo {  
    String marca;  
    String modelo;  
  
    // Get y Set  
    public String getmarca()  
    {  
        return marca;  
    }  
    public void setmarca(String marca)  
    {  
        this.marca = marca;  
    }  
    public String getmodelo()  
    {  
        return modelo;  
    }  
    public void setmodelo(String modelo)  
    {  
        this.modelo = modelo;  
    }  
  
    //metodo  
    public void detalles(){  
        System.out.println("Marca:"+getmarca()+"\nModelo:"+getmodelo() );  
    }  
}
```

Clase Coche

```
public class Coche extends Vehiculo{  
    public int numPuertas;  
    //metodo get y set  
    public int getnumPuertas()  
    {  
        return numPuertas;  
    }  
    public void setnumPuertas(int numPuertas)  
    {  
        this.numPuertas = numPuertas;  
    }  
  
    public void mostrarDetalles(){  
        //metodo mostradettales  
        detalles();  
        System.out.println("Numero de puertas: "+getnumPuertas() );  
    }  
}
```

```
public class Coche extends Vehiculo{  
  
    public int numPuertas;  
    //metodo get y set  
  
    public int getnumPuertas()  
    {  
        return numPuertas;  
    }  
    public void setnumPuertas(int numPuertas)  
    {  
        this.numPuertas = numPuertas;  
    }  
  
    public void mostrarDetalles(){  
        //metodo mostradettales  
        detalles();  
        System.out.println("Numero de puertas: "+getnumPuertas() );  
    }  
}
```

Clase Main (Principal_Coche)

```
7 import java.util.Scanner;
8
9 /**
10  *
11  * @author Kevin
12  */
13 public class Principal_Coche {
14
15     public static void main(String[] args) {
16         String marca, modelo;
17         int numPuertas;
18         Scanner dato= new Scanner (System.in);
19
20         System.out.println("Introduce la marca del carro:");
21         marca=dato.nextLine();
22         System.out.println("Introduce el modelo del carro:");
23         modelo =dato.nextLine();
24
25         System.out.println("Introduce el numero de puertas:");
26         numPuertas =dato.nextInt();
27
28         Coche Cl=new Coche ( );
29         Cl.setmarca(marca);
30         Cl.setmodelo(modelo);
31         Cl.setnumPuertas(numPuertas);
32
33         System.out.println("Metodo que solo muestra marca y modelo:" );
34         Cl.detalles();
35         System.out.println("Metodo que muestra detalles completos del coche " );
36         Cl.mostrarDetalles();
37     }
38 }
39
40
```

```
import java.util.Scanner;

/**
 *
 * @author Kevin
 */
public class Principal_Coche {

    public static void main(String[] args) {
        String marca, modelo;
        int numPuertas;
        Scanner dato= new Scanner (System.in);

        System.out.println("Introduce la marca del carro:");
        marca=dato.nextLine();
        System.out.println("Introduce el modelo del carro:");
        modelo =dato.nextLine();

        System.out.println("Introduce el numero de puertas:");
        numPuertas =dato.nextInt();

        Coche C1=new Coche ( );
        C1.setmarca(marca);
        C1.setmodelo(modelo);
        C1.setnumPuertas(numPuertas);

        System.out.println("Metodo que solo muestra marca y modelo:" );
        C1.detalles();
        System.out.println("Metodo que muestra detalles completos del coche " );
        C1.mostrarDetalles();

    }
}
```

RESUMEN

UML

El diagrama de clases nos ayudan a entender cómo funciona el programa, cómo se comporta y cómo se relacionan las clases, (la herencia entre la clase Coche y Vehiculo).

CLASE VEHICULO

Esta clase empezamos creando una clase denominada Vehiculo en donde los atributos tienen el acceso público debido a que en el enunciado del ejercicio nos plantea que sean de acceso público.

El metodo utilizado es el siguiente:

Metodos get y set

Motivo:

- 1: con estos metodos nosotros podemos escribir y leer nuestras variables ingresadas por el teclado
- 2: En los set nosotros guardamos la informacion y con los get nosotros podemos recuperar los valores
- 3: Con estos metodos nosotros podremos tener un mejor control a la hora de manejar nuestro codigo

Ejemplo nosotros ingresamos el dato de la marca del coche por teclado y con el set nosotros guardamos esa informacion para despues imprimirla.

Métodos:

En la clase Persona creamos el metodo: detalles(),para que dentro de los bloques de código reciban los parámetros (variables) con el fin de proporcionar la información para que el método pueda realizar e imprimir su tareas en la programación (estos métodos son llamados desde el Main).” **En este metodo nosotros obtener los datos mediante el metodo get de cada variable “.**

CLASE COCHE(Clase heredada)

Creamos una clase (Coche) en donde el atributo (numPuertas) es de acceso privado, debido a que en el enunciado del ejercicio nos plantea que sean de acceso público.

Para que una clase herede los atributos y metodos de otra clase, tenemos que agregar la palabra **extends** en: “public class Coche extends Vehiculo”, con eso demostramos que es una clase hija, en esta clase los atributos de Vehiculo (marca y modelo), vienen a formar parte de la clase Coche.

El metodo utilizado es el siguiente:

Metodos get y set para obtener la informacion ingresada por teclado.

Métodos:

En la clase Estudiante creamos el método: `mostrarDetalles()`, para que dentro de los bloques de código reciban los parámetros (variables) con el fin de proporcionar la información para que el método pueda realizar e imprimir sus tareas en la programación (estos métodos son llamados desde el Main).” **En este metodo nosotros obtener los datos mediante el metodo get de cada variable”**

CLASE MAIN:

En la clase principal(Main) llamamos a la librería Scanner, esta nos permite la entrada de datos por teclado.

- En la línea #16 y #17 declaramos variable de tipo String y Int, estas variables nos ayudaran a guardar la informacion que ingresemos por teclado
- En la línea #18 creamos un objeto de la clase scanner , esto nos permitira capturar la entrada de los datos.
- En la línea #21 empezamos a guardar los datos ejemplo: `marca=dato.nextLine()`, en el objeto marca que es un String guardaremos los datos ingresados por teclado y capturamos con el objeto “dato ” creado desde la librería scanner, y asi realizamos el mismo procesidmiento con las demas variables.
- En la línea #28 instanciamos un objeto de la clase Coche: `Coche C1=new Coche ()`, que no retorna ningun valor ya que vamos a cpaturarlos desde el metodo set.
- En la línea 29 con el objeto creado C1. Asignamos el valor ingresado por teclado ejemplo: `C1.setmarca(marca)`, esto nos permite capturar un nombre escrito por el teclado, de igual manera lo hacemos con todas las variables creadas.

Para luego en la línea #34 ejecutar al objeto C1 con el método `detalle()`, en donde podremos imprimir en pantalla un mensaje con la marca y el modelo del vehículo.

Ademas en la línea #36 utilizamos el objeto C1 para llamar la funcion `mostrarDetalles()`, y cuando ejecutemos nos imprima en pantalla un mensaje indicando los detalles del coche incluyendo el número de puertas, y con eso finalizamos el primer ejercicio de herencia.

