

Taller 4 Java Ejercicio 2

Emmanuel Valencia Lopera.

Ejercicio 1

- a. **¿Qué pasaría si se pierde la referencia de un objeto de tipo Persona?**
Este objeto sería eliminado de la memoria por el garbage collector.
- b. **¿Cómo podríamos conocer el nombre del dueño de la variable auto de la línea 11 del método main?**
Lo podríamos conocer con la invocación: `auto.getDueño().getNombre()`, primero se llama el método `getDueño()` que es un método de instancia de la clase `Vehículo` el cual devuelve el atributo `dueño` que es de tipo `Persona`, este luego llama al método `getNombre()` que es un método de instancia de la clase `Persona` el cual nos retornará el nombre del dueño.
- c. **¿De qué manera podemos agregar un dueño al Vehículo de la línea 13 del método main?**
Podemos invocar el método `setDueño()` y dentro del parámetro de `setDueño()` colocar un objeto de tipo `Persona`.
- d. **Usando la variable auto2 de la línea 13 del método main, obtenga el valor del atributo velocidadMáxima del motor del vehículo. Adjunte su propuesta**
Solución: `auto2.getMotor().getVelocidadMaxima();`
- e. **Suponga que, al momento de perder la referencia al objeto, se borra del sistema, es decir el garbage collector es muy eficiente, ¿Qué imprimiría al ejecutar el método main por consola?**
Imprimiría:
Matando a: Alejandro
Matando a: Jaime
Matando a: Alexander
Matando a: Santiago
Soy Santiago
- f. **¿Qué ocurre al momento de ejecutar la siguiente línea después de la línea 16 `System.out.println(personas[1])` ? Explique**
Imprime: Soy Daniel. Ya que `personas[1]` ahora toma el valor de `personas[2]`.
- g. **¿Qué modificación al código debo hacer para que al momento de ejecutar `System.out.println(auto2)`, me aparezca la placa del vehículo y el dueño del vehículo?**
Se debería añadir el método `toString()` en la clase `Vehículo` de la siguiente manera:
`public String toString(){`

```
return placa + dueno;  
}
```

- h. Usando la variable auto de la línea 11 del método main, y usando el atributo dueno, asigne de mejorAmigo al tercer elemento del listado personas. Adjunte su propuesta.

Solución: `auto.getDueno().setMejorAmigo(personas[2]);`

Ejercicio 2

- a. Según el siguiente código, indique qué se imprime por consola y explique el porqué de cada línea donde se imprime.

Primero se debe tener en cuenta la jerarquía de tipo de datos primitivos:

-byte

-short

-char

-int

-long

-float

-double

Se imprime:

char : Entra a int: 103

Se imprime esto por consola ya que se llama el método `funcion(c)` donde `c` es un dato de tipo `char` el cual por sobrecarga del método `funcion()` toma el método donde la firma del método tiene como parámetro un dato de tipo `int`, ya que por jerarquía el dato de tipo `char` tomaría el método del dato que le siga en la jerarquía. Se imprime lo que hay en el método y se imprime 103 ya que este es el número de el carácter 'g' en código ASCII.

short : Entra a int: 2

Se imprime esto por consola ya que se llama el método `funcion(s)` donde `s` es un dato de tipo `short` el cual por sobrecarga del método `funcion()` toma el método donde la firma del método tiene como parámetro un dato de tipo `int`, ya que por jerarquía el dato de tipo `short` tomaría el método del dato que le siga en la jerarquía.

byte : Entra a int: 1

Se imprime esto por consola ya que se llama el método `funcion(b)` donde `b` es un dato de tipo `byte` el cual por sobrecarga del método `funcion()` toma el método donde la firma del método tiene como parámetro un dato de tipo `int`, ya que por jerarquía el dato de tipo `byte` tomaría el método del dato que le siga en la jerarquía.

long : Entra a double: 9.99999999E8

Se imprime esto por consola ya que se llama el método `funcion(l)` donde `l` es un dato de tipo `long` el cual por sobrecarga del método `funcion()` toma el método donde la firma del método tiene como parámetro un dato de tipo `double`, ya que por jerarquía el dato de tipo `long` tomaría el método del dato que le siga en la jerarquía.

integer : Entra a int: 51232

Se imprime esto por consola ya que se llama el método `funcion(i)` donde `i` es un dato de tipo `int` el cual toma el método donde la firma del método tiene como parámetro un dato de tipo `int`.

double : Entra a double: 12.4

Se imprime esto por consola ya que se llama el método `funcion(d)` donde `d` es un dato de tipo `double` el cual toma el método donde la firma del método tiene como parámetro un dato de tipo `double`.

float : Entra a double: 5.650000095367432

Se imprime esto por consola ya que se llama el método `funcion(f)` donde `f` es un dato de tipo `float` el cual por sobrecarga del método `funcion()` toma el método donde la firma del método tiene como parámetro un dato de tipo `double`, ya que por jerarquía el dato de tipo `float` tomaría el método del dato que le siga en la jerarquía.

- b. Realice los siguientes cambios, teniendo siempre como referencia el código inicial. Explique cómo y por qué cambia lo que se imprime por pantalla.**

- **Active la función que recibe un short.**

Cuando se desea imprimir el método `funcion()` que toma un tipo de dato de jerarquía menor igual que el `short` (en este caso `byte` y `short`) ya no se pasa por jerarquía al método con parámetro de tipo `int` que estaba preestablecido en el código original, sino que ahora pasa a tomar el método con el parámetro de tipo `short`, así cambiarían algunos outputs:

`short : Entra a short: 2`

`byte : Entra a short: 1`

- **Active la función que recibe un float.**

Cuando se desea imprimir el método `funcion()` que toma un tipo de dato de jerarquía menor igual que el `float` (en este caso `long` y `float`) ya no se pasa por jerarquía al método con parámetro de tipo `double` que estaba preestablecido en el código original, sino que ahora pasa a tomar el método con el parámetro de tipo `float`, así cambiarían algunos outputs:

`long` : Entra a float: 1.0E9

`float` : Entra a float: 5.65

- **Comente la función que recibe un double y active la que recibe un float.**

Imprimiría error ya que cuando se quiera llamar el método `funcion(d)` con un parámetro `d` de tipo `double` por jerarquía no habría un método para este tipo de dato y habría una ambigüedad.

- **Comente todas las funciones, excepto la que recibe un double.**

Se imprimiría todo correctamente de la siguiente manera:

`char` : Entra a double: 103.0

`short` : Entra a double: 2.0

`byte` : Entra a double: 1.0

`long` : Entra a double: 9.99999999E8

`integer` : Entra a double: 51232.0

`double` : Entra a double: 12.4

`float` : Entra a double: 5.650000095367432

Esto porque cuando se llama en estas líneas el método `funcion()` el parámetro es de tipo `double` y este tipo de dato es el más alto en la jerarquía por lo que todos los tipos de datos primitivos incluyendo el `double` pueden entrar a este método sin que haya ambigüedad.