**HOJA INTERACTIVA DE APRENDIZAJE Nº 02**

**Apellidos y Nombres:** Casas Moreno Christian Joel

**Docente:** Nestor Audante Ramos **Fecha de entrega:** 24/04/2021

**INTRODUCCIÓN**

1. **Indicadores:**
   1. Identifica errores en la sintaxis Java en una clase.
   2. Identifica el uso de miembros estáticos y no estáticos.
   3. Identifica el uso del constructor, atributos y métodos.
   4. Crea clases y las asigna en paquetes en la solución de un caso de estudio.
2. **Actividades para desarrollar en forma individual:** 
   1. Complete la tabla sobre la sintaxis Java:

|  |  |
| --- | --- |
| **Código fuente** | **Errores** |
| public class T{  datos: String;  static edad int;  } | * La forma correcta de colocar los datos es primero el String. * El int debe ir antes de edad, pues el tipo de dato va antes que el dato en sí. |
| public class T{  datos: String;  static Integer marcar(String this.datos){  return 0;  }  } | * La forma correcta de colocar los datos es primero el String. * No se pone el this, pues se trata de un parámetro de un método. |
| package com.tesla.\*;  import java.util.List;  class T{  datos: String;  static edad int;  } | * El \* no debe ir. * No es necesaria la importación de la biblioteca de listas, pues esta no se utiliza en el código mostrado. * La forma correcta de colocar los datos es primero el String. * El int debe ir antes de edad, pues el tipo de dato va antes que el dato en sí. |
| package com.tesla.\*;  import java.util.\*;  class T{  String datos[ ];  static String[ ] pintar(String[ ] e[ ]){  return this.datos;  }  } | * El \* no debe ir. * Solo es necesario importar la biblioteca de ArrayList para que el código funcione * El método pintar no debe ser estático. |

* 1. Complete la tabla escribiendo dos ejemplos de una clase con métodos estáticos y no estáticos que utilicen atributos estáticos y no estáticos.

|  |
| --- |
| **Código fuente** |
| public class Estaticos1 {  private static float saldofamiliarcomun = 34500;  public Estaticos1(){}  public static void gastoComida(int cant){  saldofamiliarcomun-=cant\*800;  }  public void gastoEntretenimiento(int cant){  saldofamiliarcomun-=cant\*100;  }  public static void main(String[] args) {  Estaticos1 padre = new Estaticos1();  Estaticos1 madre = new Estaticos1();  Estaticos1 hijo = new Estaticos1();  padre.gastoEntretenimiento(7);  hijo.gastoEntretenimiento(10);  Estaticos1.gastoComida(28);  System.out.println(saldofamiliarcomun);  }  } |
| public class Estaticos2 {  static int vidascooperativo=5;  int vidasindividual;  public Estaticos2(int vidasindividual) {  this.vidasindividual = vidasindividual;  }  public void derrota(){  vidasindividual-=1;  vidascooperativo-=1;  }  public static void main(String[] args) {  Estaticos2 player1 = new Estaticos2(7);  Estaticos2 player2 = new Estaticos2(5);  player1.derrota();  player2.derrota();  player1.derrota();  player1.derrota();  System.out.println(Estaticos2.vidascooperativo);  System.out.println(player1.vidasindividual);  System.out.println(player2.vidasindividual);  }  } |

* 1. Complete la tabla corrigiendo el código fuente.

|  |
| --- |
| **Código fuente** |
| class Temp  {    Temp()  {  this(5);  System.out.println("Constructor por defecto");  }  Temp(int x)  {  this(5, 15);  System.out.println(x);  }  Temp(int x, int y)  {  System.out.println(x \* y);  }  public static void main(String args[])  {  Temp r = new Temp();  r.Temp(2,3);  }  } |

|  |
| --- |
| **Código fuente corregido** |
| class Temp  {  private int x;  private int y;    Temp()  {  Super();  System.out.println("Constructor por defecto");  }  Temp(int x)  {  System.out.println(x);  }  Temp(int x, int y)  {  System.out.println(x \* y);  }  public static void main(String args[])  {  Temp r = new Temp(2,3);  Temp p = new Temp(2);  Temp s = new Temp(0);  }  } |

* 1. Caso de estudio:

Toribio es un estudiante de primaria que se aburre de realizar operaciones aritméticas. Un día decide solicitar ayuda para crear una aplicación que le permita realizar operaciones de suma y resta donde se ingresa dos números enteros de distintos sistemas de numeración. Toribio desea mostrar la respuesta siempre en sistema 9.

Contento de haber resuelto el problema anterior asiste a sus clases de Química donde le dejan problemas de conversión de temperaturas Celsius-Kelvin (números enteros) que están expresados en base 8.

Ayude a Toribio en la creación de la aplicación para enfrentarse al nuevo objetivo de conversión de temperaturas reutilizando el código que usó al principio.

* 1. Suba el documento resuelto y la solución del caso de estudio en un directorio llamado lab-2.

1. **Referencia de material** 
   1. Alvarez, Cecilio(2021). Java Sobrecarga de métodos y constructores. Recuperado de https://www.arquitecturajava.com/java-sobrecarga-de-metodos-y-constructores/

1. **Conclusiones y recomendaciones**