**HOJA INTERACTIVA DE APRENDIZAJE Nº 04**

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

**Docente:** Nestor Audante Ramos **Fecha de entrega:** 15/05/2021

**INTRODUCCIÓN**

1. **Indicadores:**
   1. Identifica el uso de sobreescritura de métodos.
   2. Identifica el comportamiento de constructores en subclases.
   3. Crea subclases, sobrecarga y sobre escribe métodos en la solución de un caso de estudio.
   4. Crea clases internas y usa métodos con argumentos variables.
2. **Actividades para desarrollar en forma individual:** 
   1. Complete la tabla sobre la sintaxis Java:

|  |  |
| --- | --- |
| **Código fuente** | **Errores** |
| public class abstract T{  datos: String;  void abstract imprimir(){ }  } | * La forma correcta de colocar los datos es primero el String. |
| public class T2 extends T{  datos: String;  abstract void imprimir(){  }  } | * La forma correcta de colocar los datos es primero el String. |
| package com.tesla.\*;  abstract class T{  private T(){  }  } |  |

* 1. Complete la tabla escribiendo los conceptos aprendidos en clase

|  |
| --- |
| **Conceptos** |
| **Clase abstracta**  Es aquella clase que no se puede instanciar. No se pueden crear objetos a partir de esta clase. La finalidad de una clase abstracta es servir como clase base para otras clases a las que generalmente se conoce como clases "concretas". |
| **Clase concreta**  Es el tipo de clase que, a diferencia de las clases abstractas, sí se pueden instanciar objetos. |
| **Comportamiento en la invocación a constructores de subclase** |
| **Sobrecarga de métodos**  La sobrecarga de métodos es la creación de varios métodos con el mismo nombre, pero con diferente lista de tipos de parámetros. Java utiliza el número y tipo de parámetros para seleccionar cuál definición de método ejecutar. |
| **Sobreescritura de métodos**  La Sobreescritura es la forma por la cual una clase que hereda puede redefinir los métodos de su clase Padre, de esta manera puede crear nuevos métodos con el mismo nombre de su superclase. |
| **Clase interna**  Una clase interna tiene acceso a todas las variables y métodos de su clase externa y puede referirse a ellos directamente de la misma manera que lo hacen otros miembros no estáticos de la clase externa. |
| **Métodos con argumentos variables**  Un método que toma una cantidad variable de argumentos se denomina método varargs. La lista de parámetros para un método varargs no es fija, sino que tiene una longitud variable. Por lo tanto, un método varargs puede tomar una cantidad variable de argumentos.  **Defina sus restricciones de uso**  El vararg siempre debe ir al final entre los parámetros.  Debe haber solo un parámetro varargs. |

* 1. Complete la tabla corrigiendo el código fuente.

|  |
| --- |
| **Código fuente: Cree una clase interna y defina un método sobrecargado usando argumentos variables.** |
| public class Sobrecargado {  public void producto(){  System.out.println("No hay parametros.");  }  public void producto(int x){  System.out.println("Solo se encuentra un valor, el cual es "+x);  }  public void producto(int x, int y){  System.out.println("El resultado es: "+(x\*y));  }  public void producto(int x, int y, int z){  System.out.println("El resultado es: "+(x\*y\*z));  }  public void producto(int... num){  int prod = 1;  for (int k: num){  prod\*=k;  }  System.out.println("El resultado es: "+prod);  }  public static void main(String[] args) {  Sobrecargado operacion = new Sobrecargado();  operacion.producto(3,4);  operacion.producto(7,5,2);  operacion.producto(2);  operacion.producto();  operacion.producto(7,3,5,6,8,7,1);  }  } |

* 1. Caso de estudio: Sobrecarga y sobreescritura de métodos

**Parte I**

Un jugador de cartas desea disponer de un programa que se ejecute en su celular para calcular las oportunidades de ganar el juego R35 para los cual asigna valores a cada carta según la siguiente tabla:

|  |  |
| --- | --- |
| Carta | Puntaje |
| 2-10 | 4 |
| J 11 | 5 |
| Q 12 | 6 |
| K 13 | 8 |
| A 1 | 10 |

En el juego el jugador obtiene siempre dos cartas y se establece las oportunidades de cada como la suma de los puntajes de la tabla anterior.

El jugador quiere ingresar indistintamente como parámetros los valores numéricos del 1 al 13 como también las letras J, Q, K y A.

**Defina las clases necesarias para resolver el pedido del jugador.**

**Parte II**

El jugador se encuentra complacido por el programa anteriormente implementado y desea que realices modificaciones para participar adicionalmente en el juego R36 que dispone de valores modificados establecidos en la siguiente tabla:

|  |  |
| --- | --- |
| Carta | Puntaje |
| 2-10 | 5 |
| J 11 | 6 |
| Q 12 | 4 |
| K 13 | 3 |
| A 1 | 2 |

**Defina las subclases clases necesarias para resolver este nuevo pedido.**

* 1. Suba el documento resuelto y la solución del caso de estudio en un directorio llamado lab-4.

1. **Referencia de material** 
   1. Henao Cristian (2013). Sobreescritura y SobreCarga de Métodos en Java ( Overriding y Overloading ). Recuperado de <http://codejavu.blogspot.com/2014/05/sobreescritura-y-sobrecarga-de-metodos.html>
   2. RaiderTrance(2017). Tutorial Java Básico sobre escritura y sobre carga de métodos (Method overriding / overloading). Recuperado de <https://devs4j.com/2017/07/04/tutorial-java-basico-sobre-escritura-y-sobre-carga-de-metodos-method-overriding-overloading/>

1. **Conclusiones y recomendaciones**