Comenzado el	Friday, 25 de March de 2022, 18:59
Estado	Finalizado
Finalizado en	Friday, 25 de March de 2022, 19:16
Tiempo empleado	17 minutos 4 segundos
Puntos	12.20/16.00
Calificación	**7.63** de 10.00 (**76**%)

Pregunta **1**

Finalizado

Se puntúa 3.00 sobre 3.00

```
//Clase para objetos de dos dimensiones
class DosDimensiones{
    double base;
    double altura;

    void mostrarDimension(){
        System.out.println("La base y altura es: "+base+" y "+altura);
    }
}
```

```
//Una subclase de DosDimensiones para Triangulo
class Triangulo [ extends ] [ DosDimensiones ] {
    String estilo;

    double area(){
        [ return ] base*altura/2;
    }

    void mostrarEstilo(){
        System.out.println("Triangulo es: "+estilo);
    }
}

class Lados3{
    public static void main(String[] args) {
        Triangulo t1=new Triangulo();
        Triangulo t2=new Triangulo();

        t1.base=4.0;
        t1.altura=4.0;
        t1.estilo="Estilo 1";

        t2.base=8.0;
        t2.altura=12.0;
        t2.estilo="Estilo 2";

        System.out.println("Información para T1: ");
        t1.mostrarEstilo();
        t1.mostrarDimension();
        System.out.println("Su área es: "+t1.area());

        System.out.println();

        System.out.println("Información para T2: ");
        t2.mostrarEstilo();
        t2.mostrarDimension();
        System.out.println("Su área es: "+t2.area());

    }
}
```

DosDimensiones	extends	return	Main	Void	This

Pregunta **2**

Finalizado

Se puntúa 3.00 sobre 3.00

Los Sets y Gets son la forma de acceder a atributos de una clase. Generalmente, se usan con los atributos privados, ya que a los públicos se puede acceder directamente sin tener que acudir a ellos.

Revisa este código, arrastra y responde correctamente según corresponda.

```
public class Persona( ){

  private String nombre;
  private String apellidos;
  private int edad;
  private boolean sexo; // Tomaremos que el valor verdadero significa varon, y el falso mujer.

  public Persona( ){ }

  // Aquí empezamos a declarar gets y sets

  public String getNombre( ){
    return this.nombre;
  }

  public void setNombre(String nombre){
     this.nombre = nombre ;
  }

  public String getApellidos( ){
    return this.apellidos;
  }

  public void setApellidos(String nombre){
    this.nombre = Apellidos ;
```

```
public int getEdad( ){
  return this.edad;
}

public void setNombre(String edad){
    this.edad= edad ;
}

public boolean getSexo( ){
    return this.sexo ;
}

public void setNombre(String boolean ){
  this.boolean = boolean ;
}

}
```

this.nombre = nombre | this.nombre = Apellidos | this.edad= edad | this.return | this

new nombre= nombre

return this.apellidos | return this.sexo | return

Pregunta **3**

Finalizado

Se puntúa 3.00 sobre 3.00

Un ejemplo clásico de **poliformismo** es el siguiente. Responsa de forma correcta segun el programa siguiente:

```
class Animal {
  public void makeSound() {
    System.out.println("Grr...");
  }
}
class Cat [extends] Animal {
  public void makeSound() {
    System.out.println("Meow");
  }
}
class Dog [extends] Animal {
  public void makeSound() {
    System.out.println("Woof");
  }
}
```

Como todos los objetos Gato y Perro son objetos Animales, podemos hacer lo siguiente

```
public static void main(String[ ] args) {
  [Animal] a = new Dog();
  [Animal] b = new Cat();
}
```

```
Ahora, podemos llamar a los métodos makeSound()
```

```
a.makeSound();
[//Outputs "Woof"]
```

```
b.makeSound();
[//Outputs "Meow"]
```

[extends] [Animal] [Main] [This]

[//Outputs "Woof"] [//Outputs "Meow"] [//Outputs "Perro"] [//Outputs "Gato"]

Pregunta **4**

Finalizado

Se puntúa 0.00 sobre 1.00

Qué está haciendo este programa:

```
class Demo{

    public static void main(String[] args) {
        System.out.println("Hello Folks"); // Hello Folks
        Demo.main("Ducks");
    }


    public static void main(String arg1) {
        System.out.println("Hello, " + arg1); // Hello Ducks
        Demo.main("Dogs","Cats");
    }
    public static void main(String arg1, String arg2) {
        System.out.println("Hello, " + arg1 + " and "  + arg2); // Hello Dogs and Cats
    }
}
```

a. Polimorgismo

b. Encapsulamiento

c. Herencia

d. Sobre carga del Main

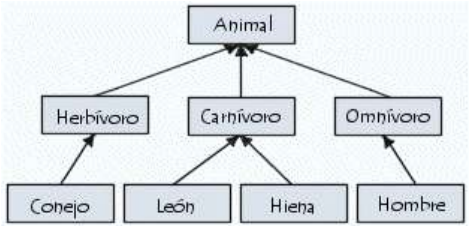
e. Todas son incorrectas

Pregunta **5**

Finalizado

Se puntúa 1.00 sobre 1.00

Este es un ejemplo de



a. Todas son incorrectas

b. Encapsulamiento

c. Herencia

d. Sobre carga métodos

e. Polimorfismo

Pregunta **6**

Finalizado

Se puntúa 0.00 sobre 1.00

Un método Void se caracteriza por

- a. Devolver valores string
- b. Recibir parámetros
- c. Todas son incorrectas
- d. Para crear un constructor
- e. Devolver Valores Numéricos

Pregunta **7**

Finalizado

Se puntúa 1.00 sobre 1.00

Es la posibilidad de crear varios métodos con el mismo nombre

Respuesta:

Polimorfismo

Pregunta **8**

Finalizado

Se puntúa 1.00 sobre 1.00

Es una variable utilizada para recibir valores de entrada en una rutina, subrutina o método.

- a. Todas son incorrectas
- b. Constructor
- c. Método
- d. Parámetro
- e. Instancia

Pregunta **9**

Finalizado

Se puntúa 0.20 sobre 1.00

```
//Atributos

public class Cuento {
private String titular;
private double cantidad;

public class Cuenta {
public Cuenta(String titular) {
    this(titular, 0);

//Constructores
}

public Cuenta(String titular, double cantidad) {
    this.titular = titular;
    //Si la cantidad es menor que cero, lo ponemos a cero
    if (cantidad < 0) {
        this.cantidad = 0;
    } else {
        this.cantidad = cantidad;
    }
}

//Metodos
public String getTitular() {
    return titular;
}

public void setTitular(String titular) {
    this.titular = titular;
}

public double getCantidad() {
    return cantidad;
}

    'Cantidad(double cantidad) {
     = cantidad;
```

```
}
```

//Sobrecarga	public class Persona {	//Parametros

Pregunta **10**

Finalizado

Se puntúa 0.00 sobre 1.00

//Este es un claro ejemplo de:

```
abstract class Piece{
    public abstract void move(byte X, byte Y);
}

class Bishop extends Piece{
 @Override
  public void move(byte X, byte Y){

  }
}
```

- a. Todas son incorrectas
- b. Polimorfismo parametrico
- c. Polimorfismo de sobre carga
- d. Polimorfismo de inclusion
- e. Polimorfismo Bayes

◄ Tarea 3

Ir a...

Foro de dudas ►