+AUTOEVALUACIÓN UT5.-UML DIAGRAMAS DE CLASES.

1- Relaciona los siguientes **conceptos de orientación a objetos** con su definición:

Concepto.		Definición.
Abstracción - **d**		Aísla las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas.
Encapsulación - **f**		Consiste en reunir bajo el mismo nombre comportamientos diferentes. La selección de uno u otro depende del objeto que lo ejecute.
Principio de ocultación - **a**		Relación que se establece entre objetos en los que unos utilizan las propiedades y comportamientos de otros formando una jerarquía
Polimorfismo - **b**		Permite capturar las características y comportamientos similares de un conjunto de objetos con el objetivo de darles una descripción formal
Herencia - **c**		Técnica por la cual el entorno de objetos se encarga de destruir automáticamente los objetos
Recolección de basura - **e**		Significa reunir todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción

2- ¿Cuál es la afirmación más adecuada al **paradigma de orientación a objetos**?

a. Permite crear aplicaciones basadas en módulos de software que representan objetos del entorno del sistema, por lo que no son apropiados para dar solución a otros problemas.
b. Tiene como objetivo la creación de aplicaciones basadas en abstracciones de datos estáticas y de difícil ampliación
c. Permite crear aplicaciones cuyo mantenimiento es complicado porque las modificaciones influyen a todos los objetos del sistema
d. Permite crear aplicaciones basadas en módulos que pueden reutilizarse, de fácil modificación y que permiten su ampliación en función del crecimiento del sistema.

3- Un **objeto** es una concreción de una clase, es decir, en un objeto se concretan valores para los atributos definidos en la clase, y además, estos valores podrán modificarse a través del paso de mensajes al objeto.
   a. Verdadero.
   b. Falso.

4- Relaciona los siguientes conceptos con su definición relacionados con **una clase**.

Concepto	Definición
Miembros - **d**	Conjunto de características asociadas a una clase
Atributos - **a**	Resultado de cierta acción efectuada por un objeto
Método - **c**	Procedimiento o función que se invoca para actuar sobre un objeto.
Mensaje – **b**	Conjunto de elementos que describen una clase.
Protocolo - **e**	Operaciones (métodos, mensajes) que manipulan el estado.

5- Relaciona los siguientes tipos de **visibilidad** que se permiten a atributos y métodos en un **diagrama de clases** con el acceso permitido:

Visibilidad	Acceso
Público – **c**	Sólo se pueden acceder desde operaciones de la clase
Privado – **a**	Sólo se pueden acceder desde operaciones de la clase o de clases derivadas en cualquier nivel.
Protegido – **b**	Se pueden acceder desde cualquier clase y cualquier parte del programa.
Paquete - **d**	Se puede acceder desde las operaciones de las clases que pertenecen al mismo paquete que la clase que estamos definiendo

6- ¿Desde dónde se puede **acceder al estado** de una clase?
   a. Desde cualquier zona de la aplicación.
   b. Desde la clase y sus clases derivadas.
   c. Solo desde los métodos de la clase.
   d. Ninguna de las anteriores.

7- ¿Qué es **UML**?
   a. Es un conjunto de herramientas que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.
   b. Unified Modeling Language.
   c. Lenguaje Unificado de Modelado.
   d. Todas las anteriores.

8- Relaciona los dos **tipos de diagramas UML** que existen con lo que representan:

Diagrama
Estructural – **b**
De comportamiento - **a**

Descripción
Muestran la conducta en tiempo de ejecución.
Representan la visión estática del sistema.

9- El **diagrama de clases** pertenece al siguiente tipo de diagrama UML:
   a. De comportamiento.
   b. Estructural.
   c. De interacción.
   d. Ninguna de las anteriores.

10- Las **herramienta CASE** para la elaboración de diagramas UML sirven solo para la generación de los diagramas asociados al análisis y diseño de una aplicación.
   a. Verdadero.
   b. Falso.

11- Al **crear una clase** en el diagrama de clases es obligatorio definir nombre, atributos y métodos.
   a. Verdadero.
   b. Falso.

12- ¿Cómo sabemos que los atributos tienen **visibilidad privada** en el diagrama?
   a. Porque aparecen acompañados del símbolo más “+”
   b. Porque aparecen acompañados del símbolo almohadilla “#”
   c. Porque aparecen acompañados del símbolo “~”
   d. Porque aparece acompañado del símbolo menos “-”

13- ¿Cuál es el **método** que no devuelve ningún tipo de dato?
   a. El constructor
   b. Todos los métodos devuelven algo, aunque sea void
   c. ~<nombre_clase>
   d. Ninguna de las anteriores.

14- Relaciona el **tipo de relaciones** entre clases con su descripción:

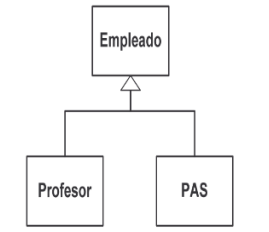
Relación	Descripción
De herencia - **c**	Es una asociación binaria que representa una relación todo-parte. Los elementos parte pueden existir sin el elemento contenedor y no son propiedad suya.
De composición - **b**	Es una agregación fuerte en la que una instancia 'parte' está relacionada, como máximo, con una instancia 'todo' en un momento dado, de forma que cuando un objeto 'todo' es eliminado, también son eliminados sus objetos 'parte'.
De agregación - **a**	Propiedad que permite a los objetos ser construidos a partir de otros objetos.

15- Se ha creado una clase persona cuyos atributos son Nombre, fechaContratación y numeroCuenta. De esta clase derivan por herencia la clase Empleado y JefeDepartamento. ¿Cómo debe declararse un **método** en la clase Persona que se llame CalculaAntigüedad que se usa sólo para calcular el sueldo de los empleados y jefes de departamento?

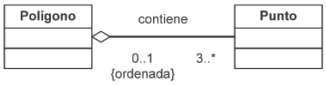
a. Público
b. Privada
c. Protegida
d. Paquete

16- Relaciona las siguientes relaciones entre clases con el **tipo de relación** que utiliza:

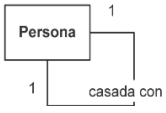
a.



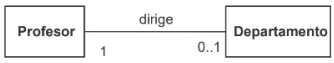
b.



c.



d.



e.

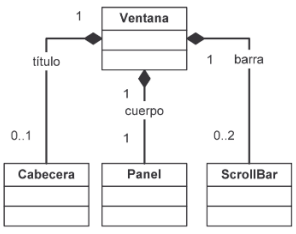


Figura
a
b
c
d
e

Tipo de relación
De Asociación - **d**
De Herencia (Generalización y Especialización) - **a**
De Composición -**b**
De Agregación - **e**
Involutiva -**c**

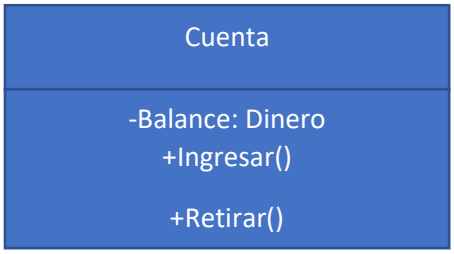
17- Dado el siguiente código en Java, se solicita **crear la clase correspondiente** son sus atributos y métodos asociados:

```
class Cuenta
{
  private Dinero balance;

  public void ingresar (Dinero cantidad)
  {
    balance += cantidad;
  }

  public void retirar (Dinero cantidad)
  {
    balance -= cantidad;
  }

  public Dinero getSaldo ()
  {
    return balance;
  }
}
```



18- Sabiendo que un **interfaz** es una clase totalmente abstracta, es decir, que no tiene atributos y todos sus métodos son abstractos y públicos, sin desarrollar. Se pide realizar el diagrama de clases a partir del siguiente código Java:

```
public interface Animal {
        public void comer();
        public void comunicarse();
        public void reproducirse();
}
public class Calamar implements Animal{
        public Calamar(){}
        public void reproducirse(){}
        public void comunicarse(){}
        public void comer(){}
}
public class Perro implements Animal{
        public Perro(){}
        public void reproducirse(){}
        public void comunicarse(){}
        public void comer(){}
}
public class Gallina implements Animal{
        public Gallina(){}
```

```
        public void reproducirse(){}
        public void comunicarse(){}
        public void comer(){}
}
```

19- Dado el siguiente código Java, se pide **realizar el diagrama de clases** correspondiente:

```
public class Persona{
        private int dni;
        private char nombre;
        private char sexo;
        private char fechanacimiento;
        public Persona(){}
}
public class Alumno extends Persona{
        private int numatricula;
        private int curso;
        public Alumno(){}
}
public class Empleado extends Persona{
        private char numerosegsocial;
        private char puestotrabajo;
        private int salario;
        public Empleado(){}
}
```

20- Indicar cuál de las siguientes **herramientas** se pueden utilizar para realizar diagramas UML:

a. ArgoUML
b. EasyUML
c. VisualParadigm
d. Todas las anteriores.
e. Ninguna de las anteriores