**1. ¿Qué es una Clase? Elegí dos opciones**

* Un conjunto de objetos organizados en categorías x
* Un modelo que define atributos y métodos que son comunes a los objetos de un mismo tipo x
* La representación gráfica de un objeto
* En programación se define como un tipo de dato

**2. ¿Qué es un Objeto? Elegí dos opciones**

* Se considera una instancia de una clase x
* Un diagrama que representa una clase x
* En programación se considera como una variable compleja
* Es un método definido en una clase

**3. Explica de forma breve los elementos que componen el diagrama de clases**

**Clase**

Una clase representa a una agrupación de cosas, es una plantilla para armar un objeto. Las clases son detectadas – en la mayoría de los casos - como sustantivos en singular. Ejemplos de una clase puede ser la clase Auto, que es una clase representante de todos los autos posibles (autos de carrera, autos urbanos, cualquier marca, patente, color, etc.) Otro ejemplo puede ser la clase Animal, representando a todos los animales posibles (mamíferos, herbívoros u otros, cualquier cantidad de patas, color, etc.) Las clases están formadas por atributos y métodos, y por convención, la primera letra debe estar en mayúscula

**Atributos**

Los atributos son características que posee una clase, y determinan el estado que posteriormente tendrá un objeto En el caso de la clase Auto, atributos pueden ser color, marca, modelo, cantidad de puertas y velocidad. Por convención, la primera letra debe estar en minúscula.

**Metodos**

Los métodos son las responsabilidades (o comportamiento) que realiza una clase. Generalmente los métodos son verbos. Por convención, la primera letra debe estar en minúscula.

**Clases abstractas**

Las clases abstractas Las clases abstractas son clases que representan un concepto abstracto, de carácter muy general. Por ejemplo una institución, que tiene los atributos dirección y superficie, pero no es posible determinar que tipo de institución es. Es una clase que no se puede instanciar, es decir que no se puede crear un objeto a partir de ésta clase. Se utiliza como base para otras clases en la relación de generalización, pero no tiene sentido por sí sola. Las clases que no son abstractas se denominan concretas. Por convención, la primera letra debe estar en mayúscula, y la palabra en negrita e itálica.

**Interfaz**

Interfaz A diferencia de la clase, la interfaz define únicamente un comportamiento, es decir un conjunto de métodos que no poseen implementación. Estos métodos deberán ser implementados por las clases que decidan implementar la interfaz. Representa un “contrato” que una clase debe respetar en caso de implementar la interfaz. Por ejemplo, se puede crear una interfaz denominada Volador, que tiene los métodos despegar, aterrizar y volar. Por convención, para d

**4. Indicar que representa los siguientes elementos de una clase:**

* -saldo: double representa los números guardados en la memoria que tienen parte entera y parte decimal
* -nombre: string representa y manipula una secuencia de caracteres
* +calcularSaldo(): void representa un valor vacio

**5. Explicar brevemente (con tus palabras) cual es la diferencia entre clases concretas, clases abstractas e interfaces**

Una clase abstracta puede heredar de una sola clase (abstracta o no) mientras que una interfaz puede extender varias interfaces de una misma vez. Una clase abstracta puede tener métodos que sean abstractos o que no lo sean, mientras que las interfaces sólo y exclusivamente pueden definir métodos abstractos.

Una clase concreta en Java es un tipo de subclase, que implementa todo el método abstracto de su clase súper abstracta a la que se extiende. También tiene implementaciones de todos los métodos de interfaces que implementa.

Una interface no puede hacer nada por sí sola, es prácticamente un contrato, en donde las clases que la implementen deben, obligatoriamente, definir el comportamiento de todos los métodos abstractos

**6. Indicar que tipo de clase o interfaz se quiere representar con la siguiente informacion:**

* Persona clase concreta
* EmpleadoAdministrativo interfaz
* Auto clase abstracta
* Controlable clase abstracta

**7. ¿Qué es la Herencia entre clases? Elegí dos opciones**

* Un método de construcción de clases, donde cada clase se concatena con otra
* Indica que la clase que hereda toma los atributos y métodos de la super clase. x
* Se puede decir que la herencia es una especialización de la super clase x
* Se puede decir que la herencia es una generalización de la clase subclase

**8. Indicar el tipo de relación:**

* Un perro es un animal Asociación
* Una empresa esta compuesta por varios departamentos dependencia
* Un banco tiene varios clientes Dependencia
* alumno es una persona Asociación

**9. Decir que tipo de relaciones son las siguientes expresiones (agregación o composición)**

Departamentos de una empresa **composición**

Clientes de un banco **agregación**

Diccionario Palabra **composición**

Frase Palabra **composición**

**10. Indicar los estereotipos de clase que representan el patrón de diseño MVC.**

1.Modelo: representa la capa de datos o lógica de negocio de la aplicación. En general, el modelo no está asociado con un estereotipo de clase específico, ya que su estructura y funcionalidad dependerán de los requerimientos y características de la aplicación en cuestión.

2.Vista: representa la capa de presentación de la aplicación, es decir, la interfaz de usuario o cualquier otro mecanismo utilizado para mostrar información al usuario final. En este caso, es común utilizar estereotipos de clase asociados con la interfaz de usuario, como Ventana, Panel, Botón, etc.

3.Controlador: representa la capa de control o coordinación entre el modelo y la vista. El controlador se encarga de procesar las solicitudes del usuario, actualizar el modelo y reflejar los cambios en la vista correspondiente. En general, se utilizan estereotipos de clase como Controlador, Gestor o Manejador para identificar las clases responsables de esta tarea.