|  |  |
| --- | --- |
| Alumno: Manuel Palmero | **Fecha límite de entrega: 20/3/25** |

El presente práctico tiene por objeto introducir al alumno en la programación orientada a objetos, observar sus particularidades y realizar un primer acercamiento a la codificación en JAVA.

A). Responda:

1. ¿Qué es un TAD?

Son los tipos abstractos de datos que define el programador. Deben ser tratados como a los tipos predefinidos por el lenguaje.

1. ¿Qué es un Objeto o instancia? De 3 ejemplos de objetos.

Un objeto es una instancia de la Clase a la que pertenece. Por ejemplo:

De la Clase Animal se puede crear el objeto Gato.

De la Clase Persona se puede crear el objeto Raúl.  
 De la Clase Comida se puede crear el objeto Ñoquis.

1. ¿Dónde se produce el encapsulamiento?

El encapsulamiento se produce en los atributos de una clase u objeto.

1. ¿Cuáles son las semejanzas y las diferencias entre atributos y métodos?

Las similitudes entre atributos y métodos es que los dos tienen un calificador de acceso y un tipo de dato (o tipo de devolución en los métodos) declarados al inicio de la línea.

La diferencia es que los métodos llevan paréntesis al lado del nombre en el que pueden o no recibir parámetros y también se colocan llaves de apertura y cierre del bloque de código que corresponde a dicho método. Los atributos no tienen ésto.

1. ¿Qué es UML? ¿Y cómo se representa una clase en ese lenguaje?

UML es el Lenguaje Unificado de Modelado, en el que se puede hacer, en el caso de POO, un modelo de qué clases deben estar en un código, qué deberían poseer y qué deberían hacer. Una clase se representa de la siguiente forma:

|  |
| --- |
| Nombre de la Clase |
| + atributo  - atributo  # ATRIBUTO |
| + método  # método |

B). Una con flechas los calificadores de acceso con sus correspondientes:

1. public
2. private No se aplica a Clases
3. protected Sólo se aplica a atributos
4. friendly -default Sólo se aplica a Clases
5. static Se aplica a atributos, métodos y clases
6. final
7. primera letra en minúscula
8. primera letra en mayúscula

C). Realizando abstracción, escriba atributos y sus tipos, intentando que sean diferentes para cada ámbito de aplicación.

|  |  |  |
| --- | --- | --- |
| Clases | Ámbito en los que se aplica | |
| Comida | Para un comercio de venta | Para el experto que lo prepara |
| atributos | float precio | String[] ingredientes |
| String nombre | String nombre |
| String tipo (almuerzo, desayuno, cena, etc.) |  |
| boolean aptoParaCeliacos |  |
|  | Para el cliente que compra | Para la empresa que recicla los sobrantes |
| atributos | String nombre | String[] sobrantes |
| float precio | boolean esReciclable |
| boolean descuento | float cantidad |
|  |  |

D). Responda:

1. ¿Qué es el bytecode?  
    El bytecode de Java es un conjunto de instrucciones que se compila a partir del código fuente de Java. Es una forma intermedia que se usa para traducir el código de Java a código de máquina.
2. ¿Qué función cumple la JVM?   
    La máquina virtual de Java (JVM) es un entorno que permite ejecutar programas escritos en Java en cualquier sistema operativo o navegador.   
   Funciones de la JVM:
   1. Carga los archivos tipo .class y genera los datos binarios que corresponda.
   2. Vincula y prepara el archivo.
   3. Inicializa el programa, asignando las variables a los valores definidos en el código Java.
   4. Gestiona la memoria.
   5. Aísla el código que proviene de otra plataforma.
   6. Restringe a qué recursos del sistema puede acceder cada una de las clases que se cargan.
3. ¿Cuándo se ejecuta el recolector de basura de java?   
    El recolector de basura en Java escanea periódicamente la memoria del montón para encontrar objetos que no se están utilizando. El proceso de recogida de basura implica varios pasos, incluidos el marcado, el barrido y la compactación. La máquina virtual Java (JVM) realiza automáticamente la recogida de basura, por lo que el programador no tiene que gestionar manualmente la memoria. El recolector de basura se ejecuta en un hilo independiente y suele funcionar en segundo plano, por lo que no afecta a la ejecución normal del programa.

E). Describa los calificadores de acceso de los miembros de esta clase. Desarrolle el código de los métodos de set y get para el atributo que corresponda.

**Cajero\_automático**

+ EFECTIVO: double

# acepta: boolean

-devuelve: String

+ procesarPedido(): void

+ pago(double): boolean

F). Indique verdadero (V) o falso (F):

Un constructor…

|  |  |  |  |
| --- | --- | --- | --- |
|  | V/F | Afirmación | Es falso debido a… |
| 1 | F | Es el método principal para ejecutar el programa. | El constructor es el método que crea la instancia de la Clase que lo contiene. |
| 2 | V | Crea objetos. |  |
| 3 | F | Devuelve el valor de un atributo privado. | Eso lo hace el getter. |
| 4 | F | Tiene sentencia return | El constructor no tiene retorno, sólo construye el objeto. |
| 5 | V | Siempre hay uno por defecto, sin parámetros ni inicializaciones de atributos. | (A menos que se declare uno que sobrescriba al que está por defecto y no se declare explícitamente el por defecto) |
| 6 | V | Se puede sobrecargar. |  |
| 7 | V | Su nombre se escribe con mayúscula. |  |
| 8 | F | Su calificador de acceso es static. | Static se usa para métodos o atributos propios de la clase, no de la instancia. |
| 9 | V | Su tipo de devolución no se indica y corresponde a la Clase. |  |

H). Dados las siguientes opciones determine cuál es constructor. (Se subrayaron los correctos)

1. Auto(): void
2. Persona (boolean trabaja) : int
3. Paciente(int doc, int edad)
4. calcularSuma() : void
5. determinarFeriados (): int
6. Juego(int puntaje)

J). Defina para la clase Cajero\_automático el constructor que recibe valores para todos sus atributos