

RESUMEN GENERAL DE PROG.III

¿Qué es OOPS?

Las siglas OOPS definen el Sistema de Programación Orientada a Objetos (Object Oriented Programming system), con el cual los programas pasan a ser considerados una colección de objetos. Cada uno de estos objetos es una parte de una clase.

¿Cuáles son conceptos básicos de OOPS?

Los siguientes son los conceptos de OOPS:

1. Encapsulación
2. Herencia
3. Polimorfismo
4. Abstracción.

¿Qué es una función en línea (inline)?

La función en línea es la técnica usada por compiladores y que indica que introduzcas el cuerpo entero de la función, siempre que la función se use el código que es fuente del programa.

¿Qué es una función virtual?

La función virtual es una función parte de una clase y su función puede ser eliminada en su clase derivada. Esta función se puede implementar usando una palabra clave que se denomina virtual, y se puede presentar durante la declaración de la función. Un ejemplo de función virtual, puede ser un token en C ++, el cual se puede lograr en lenguaje C, a través del uso de punteros de función (function pointers).

¿Qué es una 'friend function'?

Esta función es el 'amigo' de una clase a la que se le deja entrar a datos públicos, protegidos o privados, en esa misma clase. Los amigos de clase se pueden verse afectados por la palabra clave de control de entrada como, público, privado o protegido.

¿Qué es la sobrecarga de funciones?

La función de sobrecarga es una función normal, pero puede realizar varias tareas. Asimismo, permite crear varios métodos con un mismo nombre que solo los va a distinguir el tipo de entrada y salida que tiene la función.

```
void add(int&a, int&b);
```

```
void add(double&a, double&b);
```

```
void add(struct bob&a, struct bob&b);
```

¿Qué es la sobrecarga del operador?

Es una función donde se aplican varios operadores y que depende de los argumentos. Operator, -, * puede ser usado para pasar a través de la función y tiene su propia prioridad para ejecutar.

¿Qué es una clase abstracta?

Una clase abstracta es aquella clase que no puede ser instanciada. No se puede crear un objeto usando una clase abstracta pero puede ser heredado. Una clase abstracta solo puede contener el método abstracto. Por esta razón, Java permite solamente el método abstracto en la clase abstracta, sin embargo, para otros lenguajes sólo permite el método no abstracto.

¿Qué es un operador ternario (ternary)?

El operador ternario es el que toma tres argumentos, también se llama un operador condicional y toma resultados y argumentos. Los argumentos y los resultados son de diferentes tipos de datos, y dependen de la función.

¿Cuál es el uso del método de finalización (finalize)?

Este método ayuda a hacer operaciones de limpieza en recursos que no son utilizados en la actualidad. El método de finalización está protegido y solo se puede acceder por medio de esta clase o una derivada.

¿Cuáles son los diferentes tipos de argumentos?

Hay dos tipos de argumentos.

- Llamada por valor (call by value): el valor que se haya pasado se modifica solo dentro de la función, y reitera el mismo valor que sea el cual se va a pasar a la función.
- Llamada por referencia (call by reference): el valor pasado cambia tanto dentro como fuera de la función y arroja el mismo valor o distinto.

¿Que hace la palabra clave super?

La palabra clave super es la que se utiliza para llamar el método overridden, el cual anula uno de sus métodos de superclase. La palabra clave deja entrar a métodos sobrescritos y a miembros escondidos de la superclase. Asimismo, reenvía una llamada de un constructor a otro constructor.

¿Qué es el método de anulación (overriding)?

Se conoce como método de anulación a una característica que deja que una subclase suministre la implementación de un método que anula en la clase principal. Se realiza la anulación de la implementación en la superclase al dar el mismo nombre de método, parámetro y tipo de retorno.

¿Qué son los tokens?

El token es reconocido por un compilador y no puede ser dividido en elementos-componentes. Entre los tokens podemos ubicar: las palabras clave, los identificadores, los constantes, los literales de string y los operadores. Sin embargo, también son considerados tokens, las comas, paréntesis, llaves y corchetes.

¿Cuál es la diferencia entre overloading y overriding?

Overloading es un enlace estático, y la overriding es un enlace dinámico. Overloading es el mismo método con distintos argumentos, y puede devolver o no, el mismo valor en la misma clase. En cambio, overriding es el mismo nombre del método, y con los mismos argumentos y tipos de devolución asociados a la clase y a su clase secundaria.

¿Cuál es la diferencia entre clase y objeto?

El objeto es una instancia de una clase. Los objetos tienen información múltiple, las clases no guardan ningún tipo de información. La definición de propiedades y funciones se realiza en clase y puede utilizarla el objeto. La clase posee subclases, pero el objeto no posee subobjetos.

¿Qué son los modificadores de acceso?

Los modificadores de acceso son aquellos que determinan el alcance que tiene el método o las variables a las cuales se entran a través de otros objetos o clases. Hay 5 tipos de modificadores de acceso:

- Private
- Protected
- Public
- Friend
- Protected Friend

¿Qué son modificadores sellados?

Los modificadores sellados (sealed modifiers) son los modificadores de acceso a los cuales no se les permiten ser heredados por los métodos. Estos también se pueden aplicar a eventos, propiedades y métodos. Es un modificador que no se puede aplicar a miembros estáticos.

¿Cómo podemos llamar al método base sin crear una instancia?

Se puede llamar “método estático”

¿Cuál es la diferencia entre new y override?

El modificador ‘new’ le va a indicar al compilador que utilice la nueva implementación y no la función de clase base. En cambio, el modificador override permite anular la función de clase base.

¿Cuáles son los distintos tipos de constructores?

Los constructores, son de tres tipos:

- El constructor por defecto / sin parámetros.
- El constructor paramétrico / con parámetros: Este constructor pasa argumentos en forma simultánea y puede crear una nueva instancia de una clase
- El constructor copy: este constructor puede crear un nuevo objeto como una copia de un objeto que ya existe.

¿Qué es early y late binding?

La vinculación temprana es la asignación de valores a variables durante el tiempo de diseño, y la vinculación tardía es la asignación de valores a las variables en el tiempo de ejecución.

¿Cuál es el modificador de acceso predeterminado en una clase?

Por defecto, es el privado (private).

¿Qué es una función virtual pura?

Una función virtual pura es aquella función que puede ser anulada en la clase derivada, pero no puede ser definida. La función virtual es pura al utilizar el operador = 0.

Ejemplo:

```
Virtual void function1() // virtual no pura  
Virtual void function2() = 0 //virtual pura
```

¿Cuáles son todos los operadores que no pueden ser sobrecargados (overloaded)?

Los operadores que no se pueden sobrecargar son

- Scope Resolution.
- Member Selection.
- Member selection a través de un puntero a una función.

¿Qué es el polimorfismo dinámico o de runtime?

El polimorfismo dinámico o en tiempo de ejecución, es el método de cancelación en el cual la llamada a una función anulada se resolverá durante el tiempo de ejecución.

¿Necesitamos un parámetro para los constructores?

No se requiere de ninguno.

¿Qué es un constructor copy?

Es un constructor especial, el cual se usa para crear un nuevo objeto como copia de un objeto que ya existe. Es decir, que habrá un solo constructor de copia que va a ser definido ya sea por el usuario o por el sistema.

¿Cuántas instancias se pueden crear para una clase abstracta?

Para una clase abstracta, no se puede crear ninguna clase.

¿Qué palabra clave se puede utilizar para sobrecargar?

La palabra clave operator se usa para sobrecargar (overloading)

¿Cuál es el especificador de acceso predeterminado en una definición de clase?

El especificador de acceso privado (private access).

¿Qué concepto OOPS se utiliza como mecanismo de reutilización?

La herencia.

- 1. Enumere y explique los aspectos más relevantes que hacen que un software de gran magnitud sea complejo.** Los aspectos más relevantes que hacen que un software sea complejos son: la complejidad del dominio del problema, la dificultad de gestionar el proceso de desarrollo, la flexibilidad que se puede alcanzar a través del software y los problemas de caracterizar el comportamiento de sistemas discretos.
- 2. ¿Consideraría Ud. al diseño orientado a objetos un desarrollo evolutivo o revolucionario? Justifique.**

Al diseño OO lo considero un desarrollo evolutivo ya que, con el correr del tiempo, se va perfeccionando y optimizando al reutilizarse código programado anteriormente.

¿Cuántos y cuáles son los modelos básicos que se manejan en el diseño orientado a objetos?

Los modelos básicos son 7: abstracción, encapsulación, modularidad, jerarquía, tipos, concurrencia y persistencia.

¿Qué es la programación orientada a objetos?

La programación orientada a objetos es un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada una de las cuales representa una instancia de alguna clase, y cuya clase son, todas ellas, miembros de una jerarquía de clases unidas mediante relaciones de herencia

SELECCIONA LA LETRA QUE CONTENGA LA RESPUESTA CORRECTA.

1.- Es una unidad dentro de un programa informático que tiene un estado y un comportamiento.

- a) Clase
- b) Objeto**
- c) Entidad
- d) Encapsulamiento

2.- Es un paradigma de programación que define los programas en término de clases y objetos.

- a) Programación Estructurada
- c) Programación OO**
- b) Compilador
- d) Programa fuente

3.- Es un conjunto de componentes del Software que pueden ser usados por los programadores para acceder a datos y servicios de datos.

- a) Dataset
- c) SqlConnection
- b) DataReader
- d) ADO.NET**

4.- Es una tecnología desarrollada por Microsoft usada para tener acceso a diferentes fuentes de información, o bases de datos, de manera uniforme.

- a) OleDb**
- c) System Data
- b) Command.ExecuteReader
- d) Namespace

5.- Identifique el signo que fue inspirado para representar el lenguaje de programación C sharp.

- a) C
- c) C#**
- b) MS
- d) GNU

6. - Son objetos y contenidos dentro de objetos de formularios.

- a) Navegadores
- c) Controles**
- b) Instancia
- d) Plantillas

7.- Representa un control que muestra los datos en una cuadrícula Personalizable

- a) DataGridView**
- c) DataList
- b) RadioButtonList
- d) CheckBoxList

8.- Son los elementos que proporcionan códigos auxiliares reutilizable que ofrecen a los usuarios códigos y estructura básicos que se pueden personalizar.

- a) Los Menú
- c) Mnemónico
- b) Plantillas del Proyecto**
- d) Teclas de Acceso rápido

9.- Es un conjunto de imágenes que se utilizan como opciones de menú para navegar dentro de una web.

- a) Controles
- c) Alineación
- b) Barra de Navegación**
- d) TargetType

10.- Estos configuran las propiedades de un control.

- a) Los Estilos**
- c) ListView
- b) Estructura
- d) Interfaz

Programación orientada a objetos

La programación orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Muchos de los objetos prediseñados de los lenguajes de programación actuales permiten la agrupación en bibliotecas o librerías, sin embargo, muchos de estos lenguajes permiten al usuario la creación de sus propias bibliotecas.



Está basada en varias técnicas, como las siguientes:

- *herencia.*
- cohesión.
- *abstracción.*
- *polimorfismo.*
- acoplamiento.
- *encapsulación.*

La POO tiene sus raíces en la década del 60 con el lenguaje de programación `Simula` que en 1967, el cual fue el primer lenguaje que posee las características principales de un lenguaje orientado a objetos.

Smalltalk (de 1972 a 1980) es posiblemente el ejemplo canónico, y con el que gran parte de la teoría de la POO se ha desarrollado. Más su uso se popularizó a principios de la década de 1990.

En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

Los objetivos de la POO son:

- Organizar el código fuente, y
- re-usar código fuente en similares contextos.

POO en Python

El mecanismo de clases de Python agrega clases al lenguaje con un mínimo de nuevas sintaxis y semánticas.

En Python las clases es una mezcla de los mecanismos de clase encontrados en C++ y Modula-3.

Como es cierto para los módulos, las clases en Python no ponen una barrera absoluta entre la definición y el usuario, sino que más bien se apoya en la cortesía del usuario de no “forzar la definición”.

Sin embargo, se mantiene el poder completo de las características más importantes de las clases: el mecanismo de la herencia de clases permite múltiples clases base, una clase derivada puede sobrescribir cualquier método de su(s) clase(s) base, y un método puede llamar al método de la clase base con el mismo nombre.

“Los objetos pueden tener una cantidad arbitraria de datos.”

En terminología de C++, todos los miembros de las clases (incluyendo los miembros de datos), son *públicos*, y todas las funciones miembro son *virtuales*.

Como en Modula-3, no hay atajos para hacer referencia a los miembros del objeto desde sus métodos: la función método se declara con un primer argumento explícito que representa al objeto, el cual se provee implícitamente por la llamada.

Como en Smalltalk, las clases mismas son objetos. Esto provee una semántica para importar y renombrar.

A diferencia de C++ y Modula-3, los tipos de datos integrados pueden usarse como clases base para que el usuario los extienda.

También, como en C++ pero a diferencia de Modula-3, la mayoría de los operadores integrados con sintaxis especial (operadores aritméticos, de subíndice, etc.) pueden ser redefinidos por instancias de la clase.

(Sin haber una terminología universalmente aceptada sobre clases, haré uso ocasional de términos de Smalltalk y C++. Usaría términos de Modula-3, ya que su semántica orientada a objetos es más cercana a Python que C++, pero no espero que muchos lectores hayan escuchado hablar de él).

Algunas particularidades de POO en Python son las siguientes:

- Todo es un objeto, incluyendo los tipos y clases.
- Permite herencia múltiple.
- No existen métodos ni atributos privados.
- Los atributos pueden ser modificados directamente.
- Permite “monkey patching”.
- Permite “duck typing”.
- Permite la sobrecarga de operadores.
- Permite la creación de nuevos tipos de datos.

A continuación se procede a definir algunos conceptos necesarios para entender la POO:

Objetos

Los *objetos* son abstracción de Python para data. Toda la data en un programa Python es representado por objetos o por relaciones entre objetos. (En cierto sentido, y en el código modelo de Von Neumann de una “computadora almacenada del programa” también es un código representado por los objetos.)

Cada objeto tiene una identidad, un tipo y un valor. Una identidad de objeto nunca cambia una vez es creada; usted puede pensar eso como la dirección de objeto en memoria. El operador *in* compara la identidad de dos objetos; la función *id()* devuelve un número entero representando la identidad (actualmente implementado como su dirección).

El *tipo* de un objeto también es inmutable. El tipo de un objeto determina las operaciones que admite el objeto (por ejemplo, “¿tiene una longitud?”) Y también define los valores posibles para los objetos de ese tipo. La función “*type()*” devuelve el tipo de un objeto (que es un objeto en sí mismo). El *valor* **de algunos objetos puede cambiar. Se dice que los objetos cuyo valor puede cambiar son *mutables*; los objetos cuyo valor no se puede cambiar una vez que se crean se llaman *immutable*. (El valor de un objeto contenedor inmutable que contiene una referencia a un objeto mutable puede cambiar cuando se cambia el valor de este último; sin embargo, el contenedor todavía se considera inmutable, porque la colección de objetos que contiene no se puede cambiar. Por lo tanto, la inmutabilidad no es estrictamente lo mismo que tener un valor incambiable, es más sutil.) La mutabilidad de un objeto está determinada por su tipo; por ejemplo, los números, las cadenas y las tuplas son inmutables, mientras que los diccionarios y las listas son mutables.

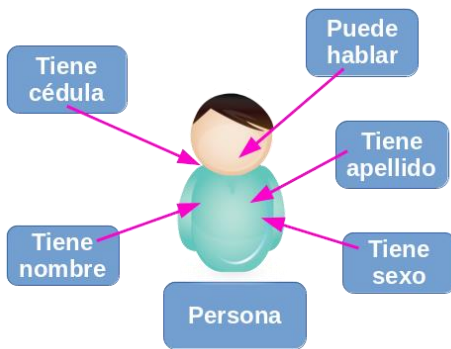
Los objetos son la clave para entender la *POO*. Si mira a nuestro alrededor encontrará un sin fin de objetos de la vida real: perro, escritorio, televisor, bicicleta, etc...

En Python puede definir una clase con la palabra reservada *class*, de la siguiente forma:

```
class Persona:
```

```
    pass
```

En el ejemplo anterior, el nombre de la clase es `Persona` y dentro del bloque de código usa la sentencia `pass`. Aunque no es requerido por el intérprete, los nombres de las clases se escriben por convención capitalizados. Las clases pueden (y siempre deberían) tener comentarios.



Estado de un objeto

El conjunto de datos y objetos relacionados con un objeto en un momento dado, se le conoce como “estado”. Un objeto puede tener múltiples estados a lo largo de su existencia conforme se relaciona con su entorno y otros objetos.

Por hacer

TODO explicar el concepto Estado de un objeto.

9.3.3. Atributos

Los atributos o propiedades de los objetos son las características que puede tener un objeto, como el color. Si el objeto es `Persona`, los atributos podrían ser: `cedula`, `nombre`, `apellido`, `sexo`, etc...

Los atributos describen el estado de un objeto. Pueden ser de cualquier tipo de dato.

```
class Persona:

    """Clase que representa una Persona"""

    cedula = "V-13458796"

    nombre = "Leonardo"

    apellido = "Caballero"

    sexo = "M"
```

Métodos

Los métodos describen el comportamiento de los objetos de una clase. Estos representan las operaciones que se pueden realizar con los objetos de la clase,

La ejecución de un método puede conducir a cambiar el estado del objeto.

Se definen de la misma forma que las funciones normales pero deben declararse dentro de la clase y su primer argumento siempre referencia a la instancia que la llama, de esta forma se afirma que los métodos son *funciones*, adjuntadas a *objetos*.

Nota

Usted puede encontrar ejemplos en las funciones de *cadena de caracteres*, *listas*, *diccionarios*, etc.

Si el objeto es `Persona`, los métodos pueden ser: hablar, caminar, comer, dormir, etc.

```
class Persona:

    """Clase que representa una Persona"""

    cedula = "V-13458796"

    nombre = "Leonardo"

    apellido = "Caballero"

    sexo = "M"

    def hablar(self, mensaje):
```

```
"""Mostrar mensaje de saludo de Persona"""
```

```
return mensaje
```

Ámbito de los métodos¶

Los métodos cuentan con un espacio de nombres propio. En caso de no encontrar un nombre en su ámbito local, buscará en el ámbito superior hasta encontrar alguna coincidencia.

Los métodos pueden acceder y crear atributos dentro del objeto al que pertenecen, anteponiendo la palabra `self` y el operador de atributo `."` antes del nombre del atributo en cuestión.

Métodos especiales¶

Las clases en Python cuentan con múltiples métodos especiales, los cuales se encuentran entre dobles guiones bajos `__<metodo>__()`.

Los métodos especiales más utilizados son `__init__()`, `__str__()` y `__del__()`.

El método `__str__()` es un método especial, el cual se ejecuta al momento en el cual un objeto se manda a mostrar, es decir es una cadena representativa de la clase, la cual puede incluir formatos personalizados de presentación del mismo.

```
def __str__(self):  
    """Devuelve una cadena representativa de Persona"""  
    return "%s: %s, %s %s, %s." % (  
        self.__doc__[25:34], str(self.cedula), self.nombre,  
        self.apellido, self.getGenero(self.sexo))
```

`__del__()`

El método `__del__()` es un método especial, el cual se ejecuta al momento en el cual un objeto es descartado por el intérprete. El comportamiento de `__del__()` es muy similar a los “destructores” en otros lenguajes.

Métodos de clase

En ocasiones es necesario contar con métodos que interactúen con elementos de la clase de la cual el objeto es instanciado. Python permite definir métodos de clase para esto.

Los métodos de clase son aquellos que están ligados directamente con los atributos definidos en la clase que los contiene. Para definir un método de clase se utiliza el decorador `@classmethod` y por convención se utiliza `cls` como argumento inicial en lugar de `self`.

Del mismo modo, los métodos de clase utilizan el prefijo `cls` para referirse a los atributos de la clase.

```
class <Clase>(object):  
  
    ...  
  
    ...  
  
    @classmethod  
    def <metodo>(cls, <argumentos>):  
  
        ...  
  
        ...
```

Métodos estáticos

Los métodos estáticos hacen referencia a las instancias y métodos de una clase. Para definir un método estático se utiliza el decorador `@staticmethod` y no utiliza ningún argumento inicial.

Al no utilizar `self`, los métodos estáticos no pueden interactuar con los atributos y métodos de la instancia.

Para referirse a los elementos de la clase, se debe utilizar el nombre de la clase como prefijo.

```
class <Clase>(object):  
  
    ...  
  
    ...  
  
    @staticmethod  
    def <metodo>(<argumentos>):  
  
        ...  
  
        ...
```

ESCRIBE F (FALSO) Ó V (VERDADERO) DE ACUERDO A SU CONSIDERACIÓN.

- 1.- En todo programa en C# debe existir un bloque llamado **Main**__v__
- 2.- La orden que permite escribir en pantalla en C# es **Console.WriteLine**_v__
- 3.-En C#, para declarar una variable que vaya almacenar números enteros se de usar el tipo de datos **entire** __f__ (**int**)
- 4.- Por legibilidad, cada vez que se escribe una llave de apertura, se debería tabular a la derecha el siguiente bloque.__v__
- 5.- La Abstracción consiste en aislar un elemento de su contexto o del resto de los elementos que lo acompañan_V__
- 6.- La Herencia es capacidad que tiene una clase en convertirse en un nuevo objeto sin cambiar su esencia y luego volver al objeto original__f__ (**polimorfismo**)
- 7.- Una matriz es una estructura de datos que contiene una serie de variables del mismo tipo_v__
- 8.-Los **Items** Permite que el usuario modifique directamente las etiquetas de los elementos__f__ (**LabelEdit**)
- 9.- La aplicación Windows forms me permite crear un formulario__v__
- 10.- **switch** es considerada una estructura de control_v__

-COMPLETA LA RAYA EN BLANCO A LAS SIGUIENTES PREGUNTAS.

- 1.-_____C#_____es un lenguaje orientado a objetos moderno, simple y poderoso.
- 2.-_____Microsoft_____ Empresa que desarrolló el lenguaje de programación C#
- 3.- _____.NET_____una nueva arquitectura tecnológica, desarrollada por Microsoft para la creación y distribución del software como un servicio.
- 4.-Se llama _____instanciar_____ crear un objeto a partir de una clase.
- 5.- Una _Interfaz gráfica_____ contiene definiciones para un grupo de funcionalidades relacionadas que una clase no abstracta o una estructura deben implementar
- 6.- _____SQL Server_____ Es el manejador de base de datos ideal para trabajar un proyecto en C#.
- 7.- El método _____**Main**_____ es el punto de entrada cuando se inicia una aplicación en C#.
- 8.- ____**Lenguaje C**_____ se le considera el lenguaje de programación estructurado donde nace el Lenguaje C#.
- 9.- Un_____ **Método** _____es un bloque de código que contiene una serie de instrucciones.

10.- La instrucción return termina la ejecución del método en el que aparece y devuelve el control al método de llamada.

(APAREA) COLOCA EL NUMERO DE LA IZQUIERDA EN LA RAYITA DE LA DERECHA.

- | | |
|---|--|
| 1. Modificabilidad | <u>10</u> Es nombre dado al momento de autenticación al ingresar a un servicio o sistema. |
| 2. La Ejecución
Prog.OO es más lenta una | <u>4</u> una cadena de caracteres, palabras, ristra de caracteres o frase es secuencia ordenada. |
| 3. Atributos | <u>3</u> Es una especificación que define una propiedad de un objeto, elemento o archivo. |
| 4.- String | <u>9</u> Representa los flujos de entrada, salida y error estándar para aplicaciones de consola. |
| 5. Console.Readkey | <u>5</u> Este método se utiliza para obtener la tecla presionada por el usuario en la ventana de la consola. |
| 6.- Programación
Orientada Obj
consiste | <u>8</u> Es una herramienta gráfica en la interfaz de aplicaciones que de una lista de opciones. |
| 7. Programación
Estructurada
claridad,
computadora | <u>7</u> Es un paradigma de programación orientado a mejorar la calidad y tiempo de desarrollo de un programa de |
| 8.- Menú | <u>6</u> Es un paradigma de programación que viene a innovar las formas de obtener resultados. |
| 9.- Console | <u>1</u> Es una ventaja de la programación Orientada a Objetos. |
| 10.-Login | <u>2</u> Una desventajas de la Programación Orientada a Objeto. |

Los siguientes términos relacionado a la programación Orientada a Objetos.

1.- ¿Qué es un DataTable?

Es el que Proporciona información de referencia sobre el objeto **DataTable** .

2. ¿Qué tipos de errores pueden ocurrir en un programa C#?

Error de Sintaxis, Error de Lógica, Error de Ejecución.

3.- ¿Cuál es la diferencia entre una clase y un registro?

Mientras que las clases son pasadas por referencia, los registros son pasados por valor. Las clases pueden ser heredadas, pero los registros no.

4.- ¿Qué es un constructor?

Un constructor es el método de una clase que se llama cuando un objeto de esa clase se crea.

5.- ¿Cuáles son las ventajas de C# sobre C, C++ o Java?

Y lo más importante, C# interactúa excepcionalmente bien con otros lenguajes de la plataforma .NET. Por esta razón, C# es una mejor opción para construir aplicaciones para .NET.

6.- ¿Qué es SqlConnection?

Es una clase que nos sirve para construir objetos de conexión a una base de datos de Sql Server

7.-¿Menciones dos características del Lenguaje C#?

Sintaxis sencilla. La sintaxis de C# es muy similar a Java, lo que simplifica al desarrollador a la hora de escribir código.

Espacio de nombres. Puedes agrupar o aislar código mediante namespaces, parecido al `package` de Java.

8.-¿Qué una Interfaz?

Dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro.

9.- ¿Qué es un proyecto informático?

Es un sistema de cursos de acción simultánea y/o secuencial que incluye personas, equipamientos de hardware, software y comunicaciones, enfocados en obtener uno o más resultados deseables sobre un sistema de información.

10.- ¿Para que utilizamos la declaración “using”?

Using se utiliza para asegurarse de liberar los recursos del objeto utilizado, ya que siempre llama “Dispose” cuando el termina su bloque de código.

SELECCIÓN MULTIPLE) SELECCIONA LA LETRA QUE CONTENGA LA RESPUESTA CORRECTA.

1.- Esta palabra clave hace referencia a la instancia actual de la clase y también se usa como modificador.

- a) **this** b) static c) Void d) Using

2.- Es un modificador de acceso para tipos y miembros de tipo, es el nivel de acceso mas permisivo.

- a) const **b) public** c) True d) While

3.- Esta instrucción ejecuta una instrucción o un bloque de instrucciones mientras una expresión booleana especificada se evalúa como true.

- a) Main **b) For** c) Ren d) Lock

4.- Es un literal que representa una referencia nula que no hace referencia a ningún objeto.

- a) Override b) void **c) null** d) short

5.- Finaliza la ejecución del bucle contenedor más próximo o de la instrucción switch.

- a) break** b) char c) operator d) sizeof

6.-Es el que contiene todos los componentes visuales que nos permiten elaborar nuestro formulario.

- a) Propiedades **b) Cuadro Herramientas** c) Exp.soluciones d)TextBox

7.- Contienen elementos que representan los scripts, las consultas, la información de conexión y los archivos necesarios para crear la solución de base de datos.

- a) Eventlong b) ImageList **c) Exp.de Soluciones** d) Messages

8.- Son los elementos que colocaremos en el formulario y este en sí, es también un objeto.

- a) Objetos** b) Ventanas c) Eventos d) Barras

9.- Permite cambiar el color del fondo del formulario.

- a) Cursor **b) BackColor** c) ControlBox d) Enabled

10.- Este evento indica que cuando pase el puntero del mouse por el formulario se ejecutará lo que este codificado en él.

- a) Click b) FormClosing **c) MouseMove** **d)**
CursorChanged

¿Cuáles son los elementos de la programación?

Algoritmos e Instrucciones. Los algoritmos son uno de los **elementos** esenciales de un programa. Un algoritmo se puede ver como la solución de un problema muy preciso y pequeño, en el cual se define la secuencia de instrucciones que se debe seguir para resolverlo.

Hay diferentes **Tipos de Programación** que son muy importantes para un computador, según el problema que se necesite resolver. Entre los que podemos encontrar: La **Programación Estructurada**, La **Programación Modular**, La **Programación Concurrente**, La **Programación Lógica** y La **Programación Funcional**.

Que hace la palabra Static? Bueno una clase, metodo o campo declarado como estatico puede ser accedido o invocado sin la necesidad de tener que instanciar un objeto de la clase.

Que son las Excepciones? Una excepcion es un evento que ocurre durante la ejecucion del programa, interrumpiendo el flujo normal.

Que tipo de excepciones conoces?

Error: Indican problemas muy graves que suelen ser no recuperables.

Exception: no definitivas pero se detectan fuera del tiempo de ejecucion.

RoutineException: son excepciones que se dan durante la ejecucion del programa.

Con que sentencia se maneja las excepciones?

Se maneja con la sentencia try-catch-finally

puntero (pointer) 'this' define al objeto actual de una clase. Esta palabra clave es utilizada como puntero que va a distinguir entre el objeto actual y el objeto global. Es decir, está referido al objeto actual.

Qué es el método de anulación (overriding)?

Se conoce como método de anulación a una característica que deja que una subclase suministre la implementación de un método que anula en la clase principal. Se realiza la anulación de la implementación en la superclase al dar el mismo nombre de método, parámetro y tipo de retorno.

¿Cómo podemos llamar al método base sin crear una instancia?

Se puede llamar "método estático"

¿Una clase puede manejar objetos?

Sí, una clase puede manejar objetos. Aunque esta frase es bastante ambigua, apréndetela de memoria para el examen y no le des más vueltas.

¿Pueden crearse objetos del tipo definido por una clase sin constructor declarado?

Sí, en caso de que no haya constructor declarado, java aplica por defecto el criterio de permitir la creación de objetos como si existiera el constructor vacío (sin código)

¿Una clase puede tener como nombre un número o su nombre empezar por un número?

No. Al igual que no es válido que el nombre de una variable empiece con un número, tampoco el nombre de una clase puede empezar con un número.

¿Este código compila y si compila, qué sentido tiene?

```
private class MiClase1 { }
```

No, el código no compila. Solo se admite public. Podría compilar si se tratara de una clase interna dentro de otra clase, pero tal y como está no.

ALGUNAS PREGUNTAS RELACIONADO AL LENGUAJE JAVA.

¿Qué símbolos se utilizan para comentarios de una sola línea?

```
//
```

¿Qué se puede utilizar como identificador?

\$

Menciona un ejemplo de palabras clave y reservadas.

Cast future generic

Que tipo de literales existen?

Enteros, booleanos, cadenas, reales y caracteres

Menciona 5 tipos de operadores:

/.*.-+.=

¿Cuántos tipos de separadores se pueden utilizar?

6

¿Qué es un objeto en java?

Conjunto de variables y funciones relacionadas con estas variables.

¿Cuántos tipos de datos simples existen en el lenguaje de programación de java?

Son 9 tipos de datos simples.

¿Cómo se llama el tipo de dato entero muy pequeño?

Byte

¿Cuánto mide el valor lógico de tipo boolean?

1 bit

PREGUNTAS DE OPCION MULTIPLE

1.- Tiene al menos un método abstracto:

a) Final **b) abstract** c) public

2.- se declara como la clase que termina la cadena de una herencia:

a) Final b) abstract c) public

3.- son accesibles desde otras clases bien sea directamente o por herencia:

a) Final b) abstract **c) public**

4.-pueden ir precedidas en su declaración por uno de los modificadores de acceso:

a) Variables miembro de objeto b) variables miembro de clase c) variables finales

5.- se crean anteponiendo la palabra **static** a su declaración:

- a) Variables miembro de objeto b) variables miembro de clase c) variables finales
- 6.- no puede cambiar su valor durante la ejecución del programa:
a) Variables miembro de objeto b) variables miembro de clase c) variables finales
- 7.- en java la entrada desde teclado y la salida hasta pantalla están reguladas por la clase system. Esta clase pertenece al package...
a)java.io b)java.lang c)java.net
- 8.- ¿Cuál de los siguientes package no existen en el lenguaje de programación de java?
a)java.io b)java.pet c)java.applet
- 9.-objeto de la clase inputstream preparado para recibir datos desde la entrada estándar del sistema:
a)system.in b)system.out c)system.else
- 10.-objeto de la clase printstream que imprimirá los datos en la salida estándar del sistema:
a)system.in b)system.out c)system.else
- 11.-los packages se utilizan con las siguientes finalidades excepto una. ¿Cuál es?
a)para agrupar clases relacionadas b)para evitar conflictos de nombres c)de decoración
- 12.-¿Cuántos tipos de variables miembro existen?
a)51 b)6 c)3
- 13.-¿Qué palabra clave permite agrupar clases e interfaces?
a)compact b)boogle c)package
- 14.- ¿cuantas formas de utilizar import. Para una clase y para todo un package existen?
a)2 b)5 c)10
- 15.- Si no se especifica una superclase se asume que se hereda de la clase:
a) Object b) public c) implements

Interfaces

La forma en que los métodos de un objeto pueden ser accedidos por otros objetos se conoce como “interfaz”. Una interfaz bien definida permite a objetos de distinta índole interactuar entre sí de forma modular. La interfaz define el modo en que los objetos intercambian información.

Por hacer

TODO explicar el concepto de Interface

Implementaciones

Una implementación corresponde al mecanismo interno que se desencadena en un método cuando éste es llamado. Las implementaciones procesan las entradas proveniente de las interfaces y actúan en consecuencia ya sea:

- Modificando el estado del objeto.
- Transfiriendo la información resultante del proceso interno a través de la interfaces.

Por hacer

TODO explicar el concepto de Implementaciones.

Clases

Las clases definen las características del *objeto*.

Con todos los conceptos anteriores explicados, se puede decir que una clase es una plantilla genérica de un *objeto*. La clase proporciona variables iniciales de estado (donde se guardan los *atributos*) e implementaciones de comportamiento (*métodos*) necesarias para crear nuevos objetos, son los modelos sobre los cuáles serán construidos.

Instancias

Ya sabe que una clase es una estructura general del objeto. Por ejemplo, puede decir que la clase `Persona` necesita tener una cedula, un nombre, un apellido y una sexo, pero no va a decir cual es cedula, nombre, apellido y sexo, es aquí donde entran las instancias.

Una instancia es una copia específica de la clase con todo su contenido. Por ejemplo:

```
>>> persona1 = Persona("V-13458796", "Leonardo", "Caballero", "M")
```

```
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
```

```
TypeError: this constructor takes no arguments
```