

Capítulo 1.- Introducción a la Programación Orientada a Objetos

Cuestionario

- 1.1. El diseño descendente también es conocido como técnica *Top-Down*
 - a) Verdadero
 - b) Falso
- 1.2. Las especificaciones de entrada se definen en la etapa de análisis del problema
 - a) Verdadero
 - b) Falso
- 1.3. La codificación del programa consiste en realizar el Pseudocódigo
 - a) Verdadero
 - b) Falso
- 1.4. Cuando se declara un objeto, se crea una instancia de la clase a la que pertenece
 - a) Verdadero
 - b) Falso
- 1.5. Los constructores siempre se declaran de un tipo de dato específico en una clase
 - a) Verdadero
 - b) Falso
- 1.6. Los constructores pueden devolver valor
 - a) Verdadero
 - b) Falso
- 1.7. Una clase puede tener más de un constructor
 - a) Verdadero
 - b) Falso
- 1.8. Se puede crear una instancia de una clase abstracta
 - a) Verdadero
 - b) Falso
- 1.9. Una interfaz puede tener atributos
 - a) Verdadero
 - b) Falso
- 1.10. El nombre de una clase abstracta se escribe con itálicas en los diagramas de UML
 - a) Verdadero
 - b) Falso
- 1.11. Las tres etapas a seguir para la resolución de un problema son:
 - a) Análisis, Diseño del algoritmo y Especificaciones
 - b) Análisis, Diseño de algoritmo y Desarrollo e Implementación
 - c) Definición del problema, Especificaciones de entrada y Especificaciones de salida
 - d) Diseño descendente, Refinamiento y Herramientas
- 1.12. La etapa de Análisis del Problema se subdivide en...
 - a) Análisis, Diseño del algoritmo y Especificaciones
 - b) Análisis, Diseño de Algoritmo y Resolución en la computadora
 - c) Definición del problema, Especificaciones de entrada y Especificaciones de salida
 - d) Diseño descendente, Refinamiento y Herramientas
- 1.13. La descomposición del problema original en subproblemas más simples y a su vez dividir estos subproblemas en otros más simples se realiza en la etapa de ...
 - a) Análisis
 - b) Diseño del algoritmo

- c) Resolución del Problema en la Computadora
 - d) Ninguna de las anteriores
- 1.14. Característica de la POO que pone especial atención en los aspectos importantes de una entidad, ignorando aspectos irrelevantes
- a) Abstracción
 - b) Encapsulamiento
 - c) Modularidad
 - d) Herencia
- 1.15. Restricción de un miembro de una clase que no lo hace visible desde fuera de la clase sino solamente por las clases heredadas
- a) Público
 - b) Privado
 - c) Protegido
 - d) Interno
- 1.16. Símbolo utilizado para representar la restricción de acceso público en los diagramas de UML
- a) +
 - b) -
 - c) #
 - d) ~
- 1.17. Tipo de relación entre clases en donde una clase se puede formar de otras clases, donde se modela una relación entre un "todo" y sus "partes", pero los componentes no dependen de la existencia del objeto de la clase constituida
- a) Asociación
 - b) Agregación
 - c) Composición
 - d) Herencia
- 1.18. Tipo de relación entre clases que se representa con un triángulo blanco indicando una punta de flecha en los diagramas de UML
- a) Asociación
 - b) Agregación
 - c) Composición
 - d) Herencia
- 1.19. Tipo de relación entre clases que se representa con un rombo negro en los diagramas de UML
- a) Asociación
 - b) Agregación
 - c) Composición
 - d) Herencia
- 1.20. Símbolo utilizado para representar la restricción de acceso privado en los diagramas de UML
- a) +
 - b) -
 - c) #
 - d) ~