1a GUIA DE POO

(Roberto Tecla)

Nombre: **Ricardo Luis Lara Hernández**  Grupo: **2CM11**

1.-Diga a que concepto corresponde la definición siguiente: denota las **características esenciales** **que distinguen a un objeto de otra clase de objetos** y provee así una frontera conceptual definida, relativa a la perspectiva del observador.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) Identidad de un objeto | b) Encapsulamiento | c) Operación | d) Abstracción |  |

2.- Diga a que concepto corresponde la definición siguiente: **engloba todas las** (usualmente estáticas) **propiedades del objeto** mas los valores actuales (usualmente dinámicos) de cada una de sus propiedades.

|  |  |  |
| --- | --- | --- |
| A) Estado de un objeto | B) Comportamiento de un objeto |  |
| C) Identidad de un objeto | D) Interfaz de un objeto |  |

3.-Diga a que concepto corresponde la definición siguiente: Es como **actúa** y reacciona **un objeto en términos** de **cambio de estado**.

|  |  |  |
| --- | --- | --- |
| A) Encapsulamiento | B) Comportamiento de un objeto |  |
| C) Identidad de un objeto | D) Interfaz de un objeto |  |

4.-Diga a que concepto corresponde la definición siguiente: es la **propiedad** de un objeto que lo **distingue de todos los otros objetos**

|  |  |  |
| --- | --- | --- |
| A) Estado de un objeto | B) Comportamiento de un objeto |  |
| C) Identidad de un objeto | D) Interfaz de un objeto |  |

5.- Es una **colección de objetos que comparten estructura y comportamiento**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) herencia | b) Clase | c) paquete | d) ) interfaz |  |

6.- Diga a que concepto corresponde la definición siguiente: es una **función** o transformación q**ue se puede aplicar o que puede ser aplicada** por los objetos de una clase.

|  |  |  |
| --- | --- | --- |
| A) Identidad de un objeto | B) Encapsulamiento |  |
| C) Operación | C) Abstracción |  |

7.- Diga a que concepto corresponde la definición siguiente: Es una **relación entre clases** en la que **una clase comparte la estructura y/o el comportamiento definido en una o más clases**.

|  |  |  |
| --- | --- | --- |
| A) Herencia | B) Interfaz |  |
| C) Una clase abstracta | C) Encapsulamiento |  |

8.-De acuerdo con su **ámbito** que **variables** se pueden usar en mas de un método de la misma clase

|  |  |  |  |
| --- | --- | --- | --- |
| a) Las de instancia | b) Las locales | c) Ninguna de las anteriores |  |

9.-De acuerdo con su **ámbito** que **variables** se declaran dentro de un método

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a) Las de instancia | b) Las de clase | c) Las locales | d) Ninguna de las anteriores |  |

10.-Cual de las siguientes no es una **palabra reservada** en Java?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a) byte | c) catch | c) real | d) implements | e) static |  |

11.-Los tipos básicos en Java son:

|  |
| --- |
| A) String, float, double, integer, short, long, char |
| B) integer, byte, String, float, short, long |
| C) int, byte, double, float, short, long, char, boolean |
| D) int, short, long, char, double, float |

12.-Cual de las siguientes afirmaciones no es cierta para un constructor ?

|  |  |  |
| --- | --- | --- |
| A) Su tipo de retorno es void | B) Esta sobrecargado |  |
| C) Sirve para inicializar los objetos de la clase | D) Se llama como la clase |  |

13.-En Java\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**solo contiene constantes y métodos abstractos**

|  |  |  |
| --- | --- | --- |
| A) Una clase abstracta | B) Una abstracción |  |
| C) Una interfaz | D) Una operación |  |

Diga si las afirmaciones siguientes son **verdaderas(**V) **o falsas** (F):

|  |  |
| --- | --- |
| 1.-El **encapsulamiento** permite ocultar los detalles de implementación de los objetos de una clase | ( V ) |
| 2.- Si un programa en Java tiene un método **main** podemos asegurar que es una **aplicación** | ( V ) |
| 3.-Si un programa en Java tiene un método **main** podemos asegurar que es un **applet** | ( F ) |
| 4.-Si un programa en Java tiene como superclase a la clase **Applet** podemos asegurar que es un **applet** | ( V ) |
| 5.-El **estado** es lo que permite distinguir a un objeto de otro | ( F ) |
| 6.-La **herencia** en Java es múltiple | ( V ) |
| 7.-La **herencia** en Java es simple | ( V ) |
| 8.-Es la  **herencia** la que permite usar métodos y atributos definidos en una superclase | ( V ) |
| 9.-El nombre del **constructor** de los objetos de una clase no tiene que ser el mismo que el nombre de dicha clase | ( F ) |
| 10.-El **constructor** de los objetos de una clase tiene como tipo de retorno **void** | ( F ) |
| 11.-Si 2 o más métodos tienen el mismo nombre pero distinta implementación entonces el método está **sobrecargado** | ( V ) |
| 12.-Si hay más de un constructor en una clase entonces el **constructor** está **sobrecargado** | ( V ) |
| 13.-Una **clase abstracta** no puede tener instancias directas | ( V ) |
| 14.-Una **clase abstracta**, en Java, no puede tener variables de instancia | ( F ) |
| 15.-En Java si una clase tiene un **método abstracto** entonces es una clase abstracta | ( V ) |
| 16.-Un método **abstracto** no tiene implementación | ( V ) |
| 17.-Un método **abstracto** no se define es decir solo se declara | ( V ) |
| 18.-Una **interfaz**, en Java, solo contiene **constantes y métodos abstractos** | ( V ) |
| 19.-Todo lo que está dentro de una **interfaz** es **publico** | ( V ) |
| 20.-Todo lo que está dentro de una **interfaz** es **privado** | ( F ) |
| 21.-Una **interfaz** puede contener métodos abstractos y variables de instancia | ( F ) |
| 22.-Una clase no puede **implementar** más de una **interfaz** | ( F ) |
| 23.-La clase que **implementa** una interfaz no tiene que implementar todos los métodos contenidos en dicha **interfaz** | ( F ) |
| 24.-Una **variable de clase** es compartida por todos los objetos de la clase | ( V ) |
| 25.-En Java se pueden **concatenar cadenas** usando el operador + | ( V ) |
| 26.-Se usa nombreObjeto.metodo(listaParametrosReales) para llamar un **método de instancia** | ( V ) |
| 27.-Se usa nombreClase.metodo(listaParametrosReales) para llamar un **método de instancia** | ( F ) |