

Características del tipo de datos Pila

- ☐ Aumentar, disminuir, retardar
- ☒ Ampliar, desampliar, cima, esVacio
- ☐ Disminuir, vaciar, consultar

¿Qué es una interfaz?

- ☐ Es una colección de patrones conceptuales que juntos modelan el proceso de diseño y que determinan la estructura de los programas realizados.
- ☒ Es una clase abstracta pura, es decir, una clase donde se indican los métodos, pero no se implementa ninguno.
- ☐ Se refieren a posiciones de memoria
- ☐ Se oculta la representación y solo se puede acceder a la información a través de métodos públicos

¿Cuáles son las características de los paradigmas orientados a objetos?

- ☐ Herencia, Polimorfismo, Definición de clases y Formularios
- ☐ Polimorfismo, Secuencia de datos, Duplicación y Herencia
- ☒ Encapsulación, Herencia, Envío de mensajes y Polimorfismo
- ☐ Polimorfismo, Encapsulación, Herencia y Secuencia de datos

¿Qué es un paradigma de programación?

- ☒ Colección de patrones conceptuales que juntos modelan el proceso de diseño
- ☐ Secuencia por la que un objeto es definido
- ☐ Modelo por el que se define el entorno en el que el programa va a ser procesado
- ☐ Una incógnita en el programa

¿Cuáles son los paradigmas del conocimiento operacional?

- ☒ Imperativo y Orientado a Objetos
- ☐ Funcional puro y Orientado a Objetos
- ☐ Funcional lógico
- ☐ Imperativo y Funcional puro

A los elementos que dependen de su multiplicidad con otros elementos se les conoce como

- ☐ Asociados
- ☐ Compuestos
- ☐ Derivados
- ☒ Agregados

¿Cuáles son los 3 tipos de colecciones en Java?

- ☐ Conjunto, Diccionario (Matriz asociativa) y array
- ☐ Diccionario, array y objetos
- ☐ Listas, Diccionario (Matriz asociativa) y array
- ☒ Conjunto, Listas y Diccionario (Matriz asociativa)

Herencia se puede definir como:

- ☐ De una clase pueden formarse más clases
- ☒ Una clase obtiene propiedades de otra clase
- ☐ Una clase puede dividirse en varias clases
- ☐ Todas son correctas

¿Qué método se utiliza para eliminar un elemento específico de un conjunto en Java?

☒ remove(Object objeto)

☐ delete()

☐ delete(Object objeto)

☐ eliminar(Object objeto)

¿Cuál es el concepto de "polimorfismo" en POO?

☐ La capacidad de una clase para ocultar su complejidad

☒ La capacidad de una clase para tener múltiples formas.

☐ La capacidad de una clase para compartir atributos y métodos con una clase padre

¿Qué no se puede hacer con de una clase abstracta?

☐ Declarar una variable

☐ Crear una función

☒ Instanciar un objeto

¿A qué conlleva la duplicación de información?

☐ Todas son correctas

☐ Evitar realizar demasiadas peticiones dentro del programa.

☐ Mantener la misma estructura para distintas piezas de código

☒ Incremento en la dificultad de evolución en el código.

Una clase es aquella que no implementa alguno de sus métodos.

- abstracta
- hija
- padre
- heredada

Una clase abstracta sólo se utiliza para definir?

- objetos
- subclases
- métodos
- variables

Que no se puede hacer dentro de una clase abstracta

- declarar una variable
- crear una función
- instanciar un objeto

¿Qué es una interfaz?

- Es una colección de patrones conceptuales que juntos modelan el proceso de diseño y que determinan la estructura de los programas realizados.
- Es una clase abstracta pura, es decir, una clase donde se indican los métodos, pero no se implementa ninguno.
- Se refieren a posiciones de memoria
- se oculta la representación y sólo se puede acceder a la información a través de métodos públicos

¿Cómo se podría definir una herencia múltiple y que lenguajes pueden soportarlo?

- pueden extender simultáneamente más de una clase y algunos lenguajes que soportan son: C++, Perl o Python
- pueden extender simultáneamente más de una clase y algunos lenguajes que soportan son: Java, C#, Delphi o Objective-C
- pueden haber varias clases padres con el mismo nombre pero con funciones diferentes y algunos lenguajes que soportan son: Java, C#, Delphi o Objective-C
- pueden haber varias clases padres con el mismo nombre pero con funciones diferentes y algunos lenguajes que soportan son: C++, Perl o Python

¿Cuáles son los 3 tipos de colecciones en Java?

- Conjunto, Diccionario (Matriz asociativa) y array
- Diccionario, array y objetos
- Listas, Diccionario (Matriz asociativa) y array
- Conjunto, Listas y Diccionario (Matriz asociativa)

¿Cómo podrías reconocer los diccionarios?

- Son estructuras de datos donde cada elemento tiene asociado una clave que usaremos para recuperarlo.
- Los elementos no tienen una posición asignada.
- Los elementos están ordenados según su posición. Por lo tanto, podemos insertar, consultar, eliminar o reemplazar elementos de una posición determinada.

¿Cómo podrías reconocer las listas?

- Son estructuras de datos donde cada elemento tiene asociado una clave que usaremos para recuperarlo.
- Los elementos no tienen una posición asignada.
- Los elementos están ordenados según su posición. Por lo tanto, podemos insertar, consultar, eliminar o reemplazar elementos de una posición determinada.

¿Cómo podrías reconocer los conjuntos?

- Son estructuras de datos donde cada elemento tiene asociado una clave que usaremos para recuperarlo.
- Estructuras de datos donde los elementos no tienen un orden y no se permiten duplicados.
- Los elementos están ordenados según su posición. Por lo tanto, podemos insertar, consultar, eliminar o reemplazar elementos de una posición determinada.

¿Qué método se utiliza para eliminar un elemento específico de un conjunto en Java?

- remove(Object objeto)
- delete(Object objeto)
- delete()
- eliminar(Object objeto)

PREGUNTAS PROGRA II

1. ¿Qué es un paradigma de programación?
 1. Colección de patrones conceptuales que juntos modelan el proceso de diseño
 2. Secuencia por la que un objeto es definido
 3. Modelo por el que se define el entorno en el que el programa va a ser procesado
 4. Una incógnita en el programa
2. ¿Cuáles son los paradigmas del conocimiento operacional?
 1. Imperativo y Orientado a Objetos
 2. Funcional puro y Orientado a Objetos
 3. Funcional lógico
 4. Imperativo y Funcional puro
3. ¿Cuáles son las características de los paradigmas orientados a objetos?
 1. Encapsulación, Herencia, Envío de mensajes y Polimorfismo
 2. Herencia, Polimorfismo, Definición de clases y Formularios
 3. Polimorfismo, Secuencia de datos, Duplicación y Herencia
 4. Polimorfismo, Encapsulación, Herencia y Secuencia de datos
4. La programación orientada a objetos es:
 1. Un método de implementación en que los programas son organizado como colecciones cooperativas de objetos
 2. Una sentencia de implementación en que las clases son organizadas como colecciones cooperativas de objetos
 3. Un método de implementación en que las clases son organizadas como colecciones cooperativas de objetos
 4. Ninguna de las anteriores
5. Complete la frase: "En respuesta a un mensaje el destinatario deberá realizar algún/a _____ para satisfacer la petición"
 1. Implementación
 2. Método
 3. Programa
 4. Petición
6. Complete la frase: "Un programa orientado a objetos está estructurado como una comunidad de agentes que interactúan, llamados/as _____"
 1. Clases
 2. Ambientes
 3. Objetos
 4. Programas

7. ¿A qué se define la herencia?

1. De una clase pueden formarse más clases
2. Una clase obtiene propiedades de otra clase
3. Una clase puede dividirse en varias clases
4. Todas son correctas

Cuál de las siguientes oraciones es incorrecta sobre un constructor:

1. Siempre debe recibir parámetros para funcionar.
2. Da un estado inicial al objeto.
3. Tiene el mismo nombre de la clase.
4. Contiene los parámetros mínimos para que el objeto pueda vivir.

La herencia de clases consiste en:

1. Que la clase particular adopta el nombre y métodos de la clase general.
2. Que sólo los métodos de la clase son heredados por la clase particular.
3. Que los métodos y atributos de la clase son heredados por la clase particular.
4. Ninguno de los anteriores.

¿Qué implica que una clase herede un atributo privado?

1. Los atributos son heredados, pero la clase particular no posee los atributos.
2. Se necesitan métodos públicos para acceder a los atributos.
3. No es posible acceder a esos atributos y es necesario declararlos de nuevo.
4. La clase particular puede acceder a los atributos indefinidamente.

¿En cuál de las siguientes situaciones utilizaría un atributo `protected`?

1. Una clase externa quiere acceder a dicho atributo.
2. El acceso al atributo sólo se permita en la misma clase donde se declara.
3. Únicamente las clases hijas tengan acceso al atributo.
4. Tanto las clases externas como particulares tienen acceso al atributo.

¿A qué conlleva la duplicación de información?

1. Evitar realizar demasiadas peticiones dentro del programa.
2. Incremento en la dificultad de evolución en el código.
3. Mantener la misma estructura para distintas piezas de código.
4. Ninguna de las anteriores.

Si una clase A está compuesta de una clase B ¿cuál de las siguientes frases es correcta?

1. La clase A puede vivir sin existir la clase B.
2. La clase B puede vivir sin existir la clase A.
3. Ambas clases pueden existir independientemente.
4. Ambas clases no pueden existir sin la otra.

A los elementos que dependen de su multiplicidad con otros elementos se les conoce como:

1. Compuestos.
2. Agregados.
3. Asociados.
4. Derivados.

Preguntas Progra 2

1. ¿Quién es el receptor?

1. En la programación orientada a objetos, el receptor se refiere al objeto que recibe un mensaje en una interacción entre objetos
2. El receptor es el que recibe la información
3. EL receptor es el interceptor de la información

2. Qué es POO?

1. Programación orientada a objetos

2. Programación on the go
3. ¿Cómo se crea un objeto?
 1. `mi_silla = new Silla()`
 2. `mi_silla = new Silla();`
 3. `new silla = mi_silla`
 4. `silla = mi silla;`
4. Características del tipo de datos Pila,
 1. Aumentar, disminuir, retardar
 2. Ampliar, desampliar, cima, esVacio
 3. Disminuir, vaciar, consultar
5. Para que es "this".
 1. La referencia "this" (éste) permite nombrar al objeto que recibe el mensaje que se está realizando.
 2. This se refiere a un objeto
 3. This llama a una función
 4. El método this equivale a la llamada referente a "este"
6. ¿Qué es la herencia en POO?
 1. Una técnica para ocultar la complejidad del código.
 2. Un mecanismo para reutilizar código y definir nuevas clases basadas en clases existentes.
 3. Una forma de definir una variable que puede contener diferentes tipos de datos.
 4. Un patrón de diseño para encapsular la funcionalidad relacionada.
7. ¿Qué es la encapsulación en POO?
 - a. Un mecanismo para ocultar la complejidad del código.
 - b. Un patrón de diseño para encapsular la funcionalidad relacionada.
 - c. Un mecanismo para reutilizar código y definir nuevas clases basadas en clases existentes.
 - d. Un mecanismo para proteger los datos y el comportamiento de una clase.
8. ¿Qué es la herencia en POO?
 - a. Un mecanismo para ocultar la complejidad del código.
 - b. Un patrón de diseño para encapsular la funcionalidad relacionada.
 - c. Un mecanismo para proteger los datos y el comportamiento de una clase.
 - d. Un mecanismo para reutilizar código y definir nuevas clases basadas en clases existentes.
9. ¿Qué es la abstracción en POO?
 - a. Un mecanismo para ocultar la complejidad del código.
 - b. Un patrón de diseño para encapsular la funcionalidad relacionada.
 - c. Un mecanismo para proteger los datos y el comportamiento de una clase.
 - d. Un mecanismo para definir una interfaz común para un conjunto de clases.
10. ¿Cuál es el concepto de "polimorfismo" en POO?
 - a. La capacidad de una clase para ocultar su complejidad.
 - b. La capacidad de una clase para tener múltiples formas.
 - c. La capacidad de una clase para compartir atributos y métodos con una clase padre.