

Cuestionario 1:

Pregunta 1

Creamos clases para:

Seleccione una:

- a. Para poder encapsular la estructura de los objetos de forma que no sea accesible
- b. Representar la estructura y el comportamiento de todos los objetos que son instancias de la clase
- c. Para indicar el código que se ejecutara cada vez que una clase es invocada

Pregunta 2

Cuando un objeto recibe un mensaje:

Seleccione una:

- a. Decide que variables puede modificar para cumplir con la acción que le requieren
- b. Ejecuta el método que corresponde al mensaje recibido
- c. Invoca un procedimiento que decide cual es la acción a realizarse

Pregunta 3

A los objetos de nuestra aplicación los crean...

Seleccione una:

- a. Un conjunto de procedimientos especiales que se ejecutan al comenzar la aplicación
- b. Otros objetos de acuerdo a la lógica de la aplicación
- c. Los objetos que pertenecen a una clase especial que controla el flujo del código

Pregunta 4

Cuando desarrollamos software con el paradigma de objetos, nuestros programas se pueden ver como:

Seleccione una:

- a. Un conjunto de acciones (llamadas métodos) que se invocan desde los objetos
- b. Un conjunto de clases que ejecutan métodos cuando se les pide
- c. Un conjunto de objetos que colaboran entre sí enviándose mensajes

Pregunta 5

Para poder funcionar los objetos conocen:

Seleccione una:

- a. Los nombres de los procedimientos de otros objetos para poder invocarlos
- b. Las variables más importantes del programa para poder modificarlas
- c. A otros objetos a los que pueden enviarle mensajes usando el protocolo que dichos objetos exhiben

Respuestas cuestionario 1:

Pregunta 1

La respuesta correcta es: Representar la estructura y el comportamiento de todos los objetos que son instancias de la clase

Pregunta 2

La respuesta correcta es: Ejecuta el método que corresponde al mensaje recibido

Pregunta 3

La respuesta correcta es: Otros objetos de acuerdo a la lógica de la aplicación

Pregunta 4

La respuesta correcta es: Un conjunto de objetos que colaboran entre si enviandose mensajes

Pregunta 5

La respuesta correcta es: A otros objetos a los que pueden enviarle mensajes usando el protocolo que dichos objetos exhiben

Cuestionario 2:

Pregunta 1

Seleccione la afirmación correcta

Seleccione una:

- a. Si una clase es abstracta, todos sus métodos deben corresponderse con alguna interfaz
- b. En un lenguaje fuertemente tipado, Interfaz y clase abstracta son sinónimos
- c. Solo puedo aplicar polimorfismo si utilizo interfaces como tipos para las variables
- d. Las clases abstractas no deben implementar interfaces
- e. Las interfaces nos permiten definir tipos desacoplándolos de sus implementaciones

Pregunta 2

Seleccione la afirmación correcta

Seleccione una:

- a. Para filtrar una colección en Java, le envío en mensaje `stream()` con una expresión lambda como parámetro.
- b. Para filtrar una colección en Java, es recomendable utilizar un iterador y una colección adicional en la que se acumula el resultado.
- c. Para filtrar una colección en Java, utilizo la librería de Pipes and Filters que ofrecen los iteradores.
- d. Para filtrar una colección en Java, es recomendable utilizar el protocolo de streams.

Pregunta 3

Seleccione la afirmación correcta

Seleccione una:

- a. El mensaje `iterator()` recorre la colección que lo recibe.
- b. El mensaje `iterator()` que entienden las colecciones, recibe como parámetro una expresión lambda y la ejecuta con todos sus elementos.
- c. El mensaje `iterator()` transforma la colección que lo recibe en un objeto `Iterator` que se puede recorrer.
- d. El mensaje `iterator()` que entienden las colecciones retorna un objeto que abstrae la forma en la que se recorre la colección.

Pregunta 4

Seleccione la afirmación correcta

Seleccione una:

- a. El mensaje `iterator()` espera como parámetro la clase de iterador a utilizar, que dependen del tipo de colección que estamos iterando.
- b. Para iterar una colección con un iterador, le envío el mensaje `iterator()` sin importar la clase de colección con la que trato.
- c. Para iterar una colección con un iterador, si es indexada (`List`) le envío el mensaje `listIterator()` y si es un conjunto (`Set`) le envío el mensaje `setIterator()`
- d. Solo las colecciones indexadas (`ArrayList`, `Vector`, etc.) entienden el mensaje `iterator()` que permite recorrerlas.

Pregunta 5

Seleccione la afirmación correcta

Seleccione una:

- a. En un lenguaje de programación orientado a objetos las clases no constituyen tipos, solo las interfaces lo hacen
- b. En un lenguaje de programación orientado a objetos, un tipo es lo mismo que una clase
- c. Las interfaces en Java son el mecanismo para dar tipo a las clases
- d. Java es un lenguaje orientado a objetos sin tipos porque solo tiene clases
- e. Un tipo en un lenguaje orientado a objetos es un conjunto de firmas de métodos

Respuestas cuestionario 2:

Pregunta 1

La respuesta correcta es: Las interfaces nos permiten definir tipos desacoplándolos de sus implementaciones

Pregunta 2

La respuesta correcta es: Para filtrar una colección en Java, es recomendable utilizar el protocolo de streams.

Pregunta 3

La respuesta correcta es: El mensaje `iterator()` que entienden las colecciones retorna un objeto que abstrae la forma en la que se recorre la colección.

Pregunta 4

La respuesta correcta es: Para iterar una colección con un iterador, le envío el mensaje `iterator()` sin importar la clase de colección con la que trato.

Pregunta 5

La respuesta correcta es: Un tipo en un lenguaje orientado a objetos es un conjunto de firmas de métodos

Cuestionario 3:

Pregunta 1

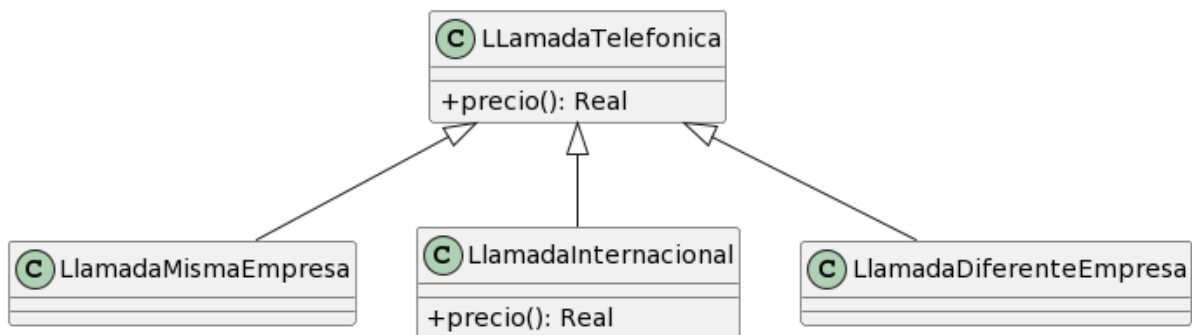
Indique cuál de estas afirmaciones es correcta:

Seleccione una:

- a. La notación para indicar que una clase es abstracta es escribiendo su nombre en cursiva, y la notación para indicar que una operación es abstracta es con el prefijo #.
- b. La notación para indicar que una clase es abstracta es poniendo su nombre en cursiva o con el estereotipo <<abstract>>. La notación para indicar que una operación es abstracta es similar (en cursiva o con el estereotipo <<abstract>>).
- c. La notación para indicar que una clase es abstracta es únicamente con el estereotipo <<abstract>>

Pregunta 2

Dado el siguiente modelo UML



¿Cuál de las siguientes afirmaciones es correcta?

Seleccione una:

- a. El diagrama es correcto. Como solo LlamadaInternacional define precio, significa que solo esa clase redefine el mensaje. LlamadaMismaEmpresa y LlamadaDiferenteEmpresa heredan la definición de LlamadaTelefonica
- b. El diagrama es incorrecto. LlamadaInternacional no puede tener un método llamado precio() porque LlamadaTelefónica ya lo define.
- c. El diagrama es incorrecto, falta el método precio() en LlamadaMismaEmpresa y en LlamadaDiferenteEmpresa
- d. El diagrama es correcto, pero ni las instancias de LlamadaMismaEmpresa ni de LlamadaDiferenteEmpresa entienden el mensaje precio()

Pregunta 3

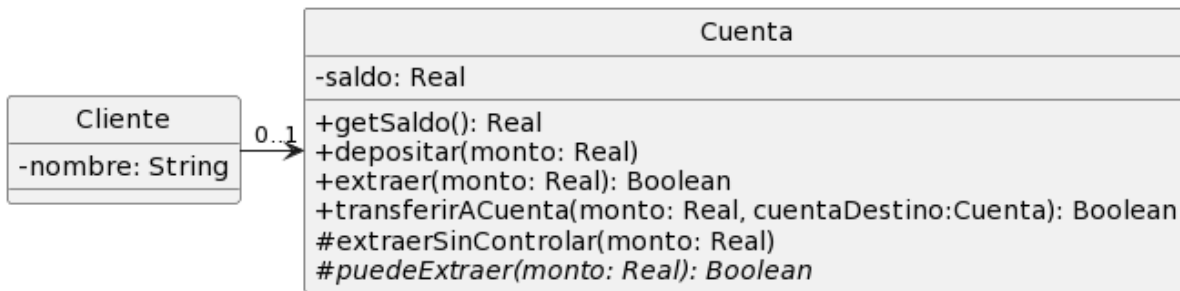
En UML, la relación de conocimiento entre objetos o instancias de clases:

Seleccione una:

- a. se modela con una asociación hacia el/los objetos que se conocen, agregando en el final de la asociación nombre (rol) y multiplicidad.
- b. se representa como un atributo que identifica al/los objetos que se conocen
- c. debe ser siempre con dirección, es decir, navegable únicamente para uno de los extremos
- d. debe ser siempre bidireccional, es decir, navegable hacia las dos clases

Pregunta 4

Según el siguiente modelo UML. ¿Cuál de las siguientes afirmaciones es INCORRECTA?

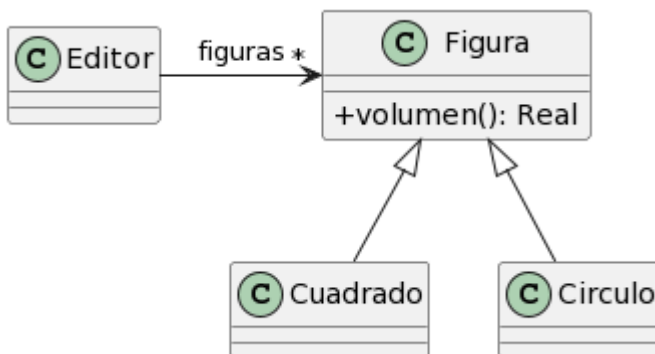


Seleccione una:

- a. Una instancia de Cliente puede enviarle el mensaje getSaldo() a una instancia de Cuenta
- b. Una instancia de Cliente puede enviarle el mensaje depositar(monto:Real) a una instancia de Cuenta
- c. Una instancia de Cliente puede enviarle el mensaje extraerSinControlar(monto: Real) a una instancia de Cuenta
- d. Una instancia de Cliente puede enviarle el mensaje extraer(monto:Real) a una instancia de Cuenta
- e. Una instancia de Cliente puede enviarle el mensaje transferirACuenta(monto:Real, cuentaDestino:Cuenta) a una instancia de Cuenta

Pregunta 5

Dado el siguiente modelo, se quiere agregar un método en la clase Editor que permita retornar las figuras que tiene.



¿Cuál de las siguientes puede ser una signatura (firma) correcta en UML para ese método?

Seleccione una:

- a. + getFiguras(): List<Figura>
- b. + getFiguras(): List
- c. + getFiguras(): Figuras
- d. + getFiguras(): Figura[*]

Respuestas cuestionario 3:

Pregunta 1

La respuesta correcta es: La notación para indicar que una clase es abstracta es poniendo su nombre en cursiva o con el estereotipo <<abstract>>. La notación para indicar que una operación es abstracta es similar (en cursiva o con el estereotipo <<abstract>>).

Pregunta 2

La respuesta correcta es: El diagrama es correcto. Como solo LlamadaInternacional define precio, significa que solo esa clase redefine el mensaje. LlamadaMismaEmpresa y LlamadaDiferenteEmpresa heredan la definición de LlamadaTelefonica

Pregunta 3

La respuesta correcta es: se modela con una asociación hacia el/los objetos que se conocen, agregando en el final de la asociación nombre (rol) y multiplicidad.

Pregunta 4

La respuesta correcta es: Una instancia de Cliente puede enviarle el mensaje extraerSinControlar(monto: Real) a una instancia de Cuenta

Pregunta 5

La respuesta correcta es: + getFiguras(): Figura[*]

Cuestionario 4:

Pregunta 1

En Reuso de código (Herencia Vs. Composición):

Seleccione una:

- a. El uso de herencia entre objetos genera más bajo acoplamiento que el uso de composición
- b. El uso de composición entre objetos genera más alto acoplamiento que el uso de herencia
- c. El reuso por composición, permite usar al objeto a través de su protocolo, sin necesidad de tener que conocer su implementación
- d. Cuando la estructura de la nueva subclase es como la de la superclase, se aconseja usar herencia, no composición. El comportamiento heredado que no sirva, puede anularse o redefinirse.

Pregunta 2

Las Heurísticas para Asignación de Responsabilidades (HAR) son útiles para:

Seleccione una:

- a. Mejorar el Diagrama de Clases del Diseño
- b. Decidir que clases candidatas forman el Modelo Conceptual o del Dominio
- c. Decidir que objeto dentro del Sistema es responsable o receptor de una operación

Retroalimentación

Pregunta 3

Seleccione la afirmación correcta

Seleccione una:

- a. Es importante testear temprano, y tanto como sea el riesgo del artefacto a testear
- b. Es importante diseñar bien el programa y chequearlo entre pares para evitar escribir tests de unidad.
- c. Es importante testear todos los métodos hasta alcanzar una cobertura mayor al 80% del código.
- d. Es importante testear ni bien se termina de escribir el programa, y con tanta cobertura como sea posible.

Pregunta 4

Seleccione la opción correcta

Seleccione una:

- a. Hacer testeo de unidad en objetos es asegurarse de no romper el encapsulamiento de los objetos.
- b. Hacer testeo de unidad en objetos es asegurarse de que el diseño hace una adecuada asignación de responsabilidades.
- c. Hacer testeo de unidad en objetos es asegurarse de que el programa no tiene malos olores de diseño (por ejemplo, métodos largos)
- d. Hacer testeo de unidad en objetos es asegurarse de que el programa hace lo que se espera, lo hace como se espera, y no falla.

Pregunta 5

El Modelo Conceptual o del Dominio es útil para:

Seleccione una:

- a. mostrar el comportamiento (métodos) de las clases candidatas.
- b. representar el dominio del problema en la etapa de Análisis
- c. representar la solución del Sistema OO, mostrando la vista estática del Diseño final.
- d. representar las interacciones entre los objetos candidatos del Sistema

Respuestas cuestionario 4:

Pregunta 1

La respuesta correcta es: El reuso por composición, permite usar al objeto a través de su protocolo, sin necesidad de tener que conocer su implementación

Pregunta 2

La respuesta correcta es: Decidir que objeto dentro del Sistema es responsable o receptor de una operación

Pregunta 3

La respuesta correcta es: Es importante testear temprano, y tanto como sea el riesgo del artefacto a testear

Pregunta 4

La respuesta correcta es: Hacer testeo de unidad en objetos es asegurarse de que el programa hace lo que se espera, lo hace como se espera, y no falla.

Pregunta 5

La respuesta correcta es: representar el dominio del problema en la etapa de Análisis

Cuestionario 5:

Pregunta 1

Seleccione la afirmación correcta

Seleccione una:

- a. Al no ser basado en clases sino en prototipos, ECMAScript no implementa el concepto de herencia.
- b. En ECMAScript, cada objeto hereda comportamiento y estado de su prototipo.
- c. En ECMAScript, cada objeto hereda estado de su prototipo. El comportamiento no se hereda.
- d. En ECMAScript, cada objeto hereda comportamiento de su prototipo. El estado no se hereda.

Pregunta 2

Seleccione la afirmación correcta

Seleccione una:

- a. En Smalltalk, no existen las clases. El lenguaje se basa en prototipos que se clonan y especializan.
- b. En Smalltalk las clases también son objetos, tienen sus propios atributos y métodos. El new, por ejemplo, es un mensaje de clase.
- c. Las clases de Smalltalk son objetos especiales, que solo entienden mensajes predefinidos para crear instancias. No es posible extender una clase para que entienda otros mensajes.
- d. En Smalltalk, al igual que en java, tenemos métodos estáticos que se definen en las clases y se invocan directamente.

Pregunta 3

Seleccione la afirmación correcta

Seleccione una:

- a. ECMAScript es un lenguaje dinámico, en el que no se indica explícitamente el tipo de las variables.
- b. ECMAScript es un lenguaje fuertemente tipado, en el que se declara el tipo de cada variable y parámetro, utilizando clases.
- c. ECMAScript puede utilizarse tanto como un lenguaje dinámico, en el que no es necesario indicar explícitamente el tipo de las variables o como un lenguaje fuertemente tipado.
- d. ECMAScript es un lenguaje fuertemente tipado, en el que se declara el tipo de cada variable y parámetro, utilizando interfaces.

Pregunta 4

Seleccione la afirmación correcta

Seleccione una:

- a. En Smalltalk todo se implementa con objetos. Sin embargo, la implementación de los objetos de las librerías base (como Boleanos, Strings, colecciones, etc.) no puede modificarse.
- b. En Smalltalk casi todo se implementa con objetos. Los tipos primitivos (Integer, String, etc.) y las estructuras de control no.
- c. Smalltalk hace una clara diferencia entre las partes de nuestra aplicación implementadas en objetos, por nosotros, y los elementos base del lenguaje (por ejemplo, librerías) que han sido implementados combinando paradigmas (objetos, funcional, procedural).
- d. En Smalltalk todo se implementa con objetos y está abierto a modificación. Incluso lo que comúnmente conocemos como estructuras de control (como el if, while, etc.) se implementa como envíos de mensajes a objetos.

Pregunta 5

Seleccione la afirmación correcta:

Seleccione una:

- a. Smalltalk es un lenguaje fuertemente tipado, en el que se declara el tipo de cada variable y parámetro, utilizando interfaces.
- b. Smalltalk puede utilizarse tanto como un lenguaje dinámico, en el que no es necesario indicar explícitamente el tipo de las variables o como un lenguaje fuertemente tipado.
- c. Smalltalk es un lenguaje fuertemente tipado, en el que se declara el tipo de cada variable y parámetro, utilizando clases.
- d. Smalltalk es un lenguaje dinámico, en el que no se indica explícitamente el tipo de las variables.

Respuestas cuestionario 5:

Pregunta 1

La respuesta correcta es: En ECMAScript, cada objeto hereda comportamiento y estado de su prototipo.

Pregunta 2

La respuesta correcta es: En Smalltalk las clases también son objetos, tienen sus propios atributos y métodos. El new, por ejemplo, es un mensaje de clase.

Pregunta 3

La respuesta correcta es: ECMAScript es un lenguaje dinámico, en el que no se indica explícitamente el tipo de las variables.

Pregunta 4

La respuesta correcta es: En Smalltalk todo se implementa con objetos y está abierto a modificación. Incluso lo que comúnmente conocemos como estructuras de control (como el if, while, etc.) se implementa como envíos de mensajes a objetos.

Pregunta 5

La respuesta correcta es: Smalltalk es un lenguaje dinámico, en el que no se indica explícitamente el tipo de las variables.