

ORIENTACIÓN A OBJETOS 1

Autoevaluaciones

AUTOEVALUACIÓN 1

En un programa construido con objetos

Seleccione una:

- ☐ a. Se comienza con un código especial que invoca al primer objeto que corresponde
- ☐ b. Es imprescindible especificar siempre un objeto que será la raíz del sistema en todas las ejecuciones
- ☒ c. No hay un objeto más importante que otros. El comienzo de una aplicación depende del flujo de control, de decisiones del desarrollador, del tipo de interacción, etc ✓

La respuesta correcta es: No hay un objeto más importante que otros. El comienzo de una aplicación depende del flujo de control, de decisiones del desarrollador, del tipo de interacción, etc

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

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

Para poder funcionar los objetos conocen

Seleccione una:

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

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

Pregunta 4

Incorrecta

Puntúa 0,00
sobre 1,00

Cuando en el objeto **a** de la clase **A** ejecutamos el método **m** que tiene el siguiente código:

`e = new Estudiante ()`

Seleccione una:

- ☒ a. Indicamos que la variable **a** puede enviarle mensajes a los estudiantes ✗
- ☐ b. Todos los objetos de la clase **A** pueden enviarle mensajes a ese estudiante
- ☐ c. Creamos un objeto de la clase Estudiante, que será conocido por **a** mediante la variable **e**

La respuesta correcta es: Creamos un objeto de la clase Estudiante, que será conocido por **a** mediante la variable **e**

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 metodos cuando se les pide
- ☒ c. Un conjunto de objetos que colaboran entre si enviandose mensajes ✓

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

AUTOEVALUACIÓN 2

Seleccione la afirmación correcta

Seleccione una:

- ☐ a. En un lenguaje de programación orientado a objetos, un tipo es lo mismo que una clase
- ☐ b. Las interfaces en Java son el mecanismo para dar tipo a las clases
- ☐ c. En un lenguaje de programación orientado a objetos las clases no constituyen tipos, solo las interfaces lo hacen
- ☐ 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 ✓

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

Supongamos una jerarquía con clase raíz **A**, subclases **B**, **C** y **D**.

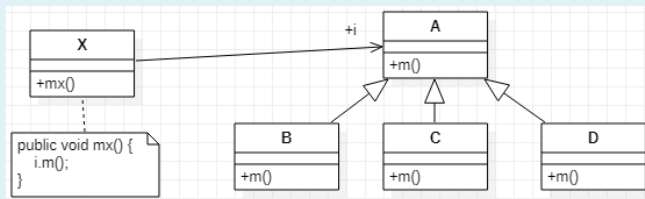
Supongamos que **B** tiene subclases **B1** y **B2**, **C** tiene **C1** y **C2** y **D** tiene **D1** y **D2**

Cuando un objeto **o** de la clase **B1** recibe un mensaje **m()**. Si no encuentra el método **m** en su clase...

Seleccione una:

- ☒ a. busca en la jerarquía de clases primero en **B**. Si lo encuentra lo ejecuta como si fuera propio. Si no lo encuentra busca en **A**. Si lo encuentra lo ejecuta como si fuera propio ✓
- ☐ b. busca en la jerarquía de clases en **B** y en **A**. Ejecuta el que considera más apropiado
- ☐ c. busca en la jerarquía de clases primero en **B**. Si lo encuentra lo ejecuta como si fuera un objeto de **B**. Si no lo encuentra busca en **A**. Si lo encuentra lo ejecuta como si fuera un objeto de **A**

La respuesta correcta es: busca en la jerarquía de clases primero en **B**. Si lo encuentra lo ejecuta como si fuera propio. Si no lo encuentra busca en **A**. Si lo encuentra lo ejecuta como si fuera propio



Supongamos que tenemos una clase **A**, con sub-clases **B**, **C** y **D**. En todas ellas tenemos una implementación del método **m()**.

Supongamos que tenemos también la clase **X** con una variable de instancia **i** de tipo **A**.

Supongamos que como respuesta a un mensaje enviado a una instancia de **X** se ejecuta el método de **mx()**, en el que se envía a **i** el mensaje **m()**.

El binding dinámico nos permite:

Seleccione una:

- ☐ a. que el compilador pueda chequear si la expresión es válida o invalida
- ☐ b. Aprovechar la herencia en la jerarquía de clases de **A**
- ☒ c. Evitar el chequeo explícito de el tipo al que pertenece el objeto apuntado por la variable **i** dejando que dicho objeto se encargue de decidir que método se ejecuta (usando el algoritmo usual de "lookup") ✓

La respuesta correcta es: Evitar el chequeo explícito de el tipo al que pertenece el objeto apuntado por la variable **i** dejando que dicho objeto se encargue de decidir que método se ejecuta (usando el algoritmo usual de "lookup")

Cuando un objeto **o** recibe un mensaje **m()**

Seleccione una:

- ☐ a. En función del tipo del objeto **o** se decide que hacer
- ☐ b. Si encuentra el método **m()** correspondiente en su clase, lo ejecuta
- ☒ c. Decide que método ejecutar chequeando los que tienen nombres iguales en la jerarquía ✖

La respuesta correcta es: Si encuentra el método **m()** correspondiente en su clase, lo ejecuta

Decimos que en un lenguaje de programación orientado a objetos existe polimorfismo cuando

Seleccione una:

- ☐ a. el mensaje **m()** puede ser recibido por objetos de clases diferentes.
- ☐ b. un objeto **o** puede mandar mensajes diferentes a otros objetos
- ☒ c. puedo elegir diferentes implementaciones del mismo método **m()** según me interese ✖

La respuesta correcta es: el mensaje **m()** puede ser recibido por objetos de clases diferentes.