

EAF2 Preguntas test:

Las preguntas de tipo test han sido respondidas mediante la ayuda del temario y conocimientos previos adquiridos.

Pregunta 1 (colección de código predefinido):

Respuesta: **C) Biblioteca** → Es una colección de código predefinido que facilita el trabajo al desarrollador.

Pregunta 2 (resultado de compilar código fuente a través del compilador):

Respuesta: **A) Código objeto** → Es el código fuente traducido por el compilador a código máquina. Este no se puede ejecutar por el ordenador.

Pregunta 3 (El código fuente puede ser ejecutado directamente por):

Respuesta: **D) Ninguna de las anteriores** → El código fuente tiene que ser “traducido” a código máquina para poder ser ejecutado por el ordenador.

Pregunta 4 (La máquina virtual de Java):

Respuesta: **B) Interpreta código de bytes** → Actúa como un puente entre el código de bytes a ejecutar y el sistema.

Pregunta 5 (Traduce código de alto nivel a un lenguaje máquina en tiempo de ejecución):

Respuesta: **C) El intérprete** → Lo hace en tiempo de ejecución a diferencia del compilador. Al hacerlo de esta manera es notablemente más lento que el compilador pero también es más portable.

Pregunta 6 (Afirmaciones NO CIERTAS sobre lenguajes que compilan a código objeto):

Respuesta: **B) En un lenguaje compilado, el mismo código, una vez compilado, puede ejecutarse en diferentes sistemas operativos** → Cuando se cambia de S.O. hay que re-compilar o volver a compilar.

Pregunta 7 (Programación orientada a objetos característica que permite la ordenación de la abstracción):

Respuesta: **D) Jerarquía** → Por ejemplo, la Herencia y la agregación.

Pregunta 8 (Imperativo, declarativo y orientado a objetos son):

Respuesta: **C) Paradigmas de programación**

Pregunta 9 (Método que sobrecarga “f(int x, int y)”):

Respuesta: **C) f(String x, int y)** → La sobrecarga se realiza con método de la misma clase para darle funcionalidades diferentes. Tiene que tener parámetros diferentes.

Pregunta 10 (Método que sobrescribe “f(int x, y)”):

Respuesta: **B) f(int x, y) a la clase B.** → La sobrescritura de un método se realiza en la herencia, es decir, se sobrescribe un método de la clase padre, para darle una funcionalidad diferente o adicional. Tiene que tener el mismo nombre, parámetros y retorno que el de la clase padre.

Pregunta 11 (Lenguajes C++ es un lenguaje de):

Respuesta: **B) Tercera generación** → Son lenguajes de alto nivel. Otros lenguajes de tercera generación con C, Java, Pascal...

Pregunta 12 (Roles de la metodología Métrica 3 NO están identificados):

Respuesta: **D) Todas las opciones se refieren a un rol válido** → Jefe de proyecto y consultores, Analistas, Partes interesadas y programadores.

Pregunta 13 (Métrica 3 → En qué fase se genera la mayor parte del código):

Respuesta: **B) Construcción del sistema de información** → Tiene como objetivo final la construcción y la prueba de los diferentes componentes del sistema de información a partir de especificaciones lógicas y físicas obtenidas de la fase de diseño.

Pregunta 14 (Métrica 3 → Fase donde se generan las especificaciones necesarias para la posterior construcción) :

Respuesta: **A) Diseño del sistema de información** → El propósito es obtener la definición de la arquitectura del sistema y el entorno tecnológico que le dará soporte, junto con la especificación detallada de los componentes del sistema de información.

Pregunta 15 (De que se trata de un programa compuesto de un conjunto de herramientas que utilizan los programadores para desarrollar código):

Respuesta: **B) Entorno de desarrollo integrado (IDE)** → Puede estar pensado para un único lenguaje de programación o para varios.

Ejercicio 16 (Pressupost Tècnic Informàtic):

Para realizar el ejercicio 16 primero tenemos que tener descargadas e instaladas las herramientas JDK (incluye JVM y JRE), el entorno o software Elipse (versión para desarrolladores), configurar el JDK con el entorno Eclipse (esto básicamente es indicar la ruta de la versión Java instalada) y por último instalar el conector “WindowsBuilder”. Para instalar este último lo hacemos directamente desde Eclipse. Para ellos nos dirigimos al menú superior *Help* → *Install New Software*. En la pantalla que aparece clicamos en “add”, en el campo “Name”, indicamos el nombre (en mi caso “WindowsBuilder”) y en el campo “Location” la ruta de descargar o el repositorio para WindowsBuilder y clicamos “OK”. Clicamos en “Select all” y en “Next”. Tendremos que reiniciar Eclipse para que los cambios tengan efecto.

Iniciamos el ejercicio:

Lo primero es crear el proyecto, el paquete y la clase. Para ello seguimos los siguientes pasos:

File → New → Other → SWT Designer (carpeta) → SWT/JFace Java Project

Indicamos el nombre del proyecto, en mi caso “Exercici16”. Después creamos la clase y el paquete. Para seguimos los siguientes pasos (nos situamos en el proyecto y con el botón derecho):

New → other → SWT (carpeta) → Application Windows

En el campo “Source folder” lo dejamos como viene por defecto. En el campo “Package” indicamos el nombre del paquete (en mi caso “eaf2_ex16”) y por último en el campo “Name”, indicamos el nombre de la clase (en mi caso “Exercici16”). Clicamos en “Finish”. Podemos observar como el conector WindowsBuilder ha añadido los paquetes necesarios.

Realización del ejercicio:

Empezamos por el diseño → Para diseñar al programa hay que hacer clic en la pestaña “Design”. En este apartado vamos añadiendo los botones (Button), campos de texto (Text) y texto (Label) tal y como aparece en la imagen del enunciado. Después de añadir botones, campos, etc.. le damos funcionalidad, esto los hacemos en el campo “Componentes”, clicando sobre el botón, campo, etc.. con el botón derecho del ratón “Add event handler → Selection → widgetSelected”. Esto creará las clases anónimas que mantendrán a la escucha dichos componentes.

NOTA: Es importante cambiar el tipo de variable de local a variable de clase o campo de clase de los botones y los campos de texto para acceder a ellos desde cualquier parte. Esto lo hacemos desde “Windows -> Preferences -> WindowBuilder -> Code Generation i seleccionar Field a Default code generation settings” o desde la pestaña “Design” en la paleta, seleccionando el botón o el campo.

Una vez tenemos terminado el diseño, empezamos con la codificación. Los primeros dos campos “Preu hora y Número d’hores” el dato a introducir serán de tipo doublé y tienen la limitación de 0-100 y 0 a 23. Hay que capturar esos datos y comprobar que sean correctos, es decir, que no sean de tipo texto, signos o no estén en rango. Por ese motivo cree la función “compruebaPreuHora” de tipo “void” con 2 parámetros de entrada (serán los valores preuHora y hores) a evaluar. Si al pulsar en el botón “Calcular” uno de los valores es incorrecto, mostrará un mensaje en el campo error (en mi caso es una variable de tipo Text “textMensaje”), se pondrá el foco en el campo con el dato erróneo y seleccionará todo el campo. Si ambos datos son correctos se habilitará el apartado “Import Pressupost” que inicialmente estaba deshabilitado y se deshabilitará el apartado superior “Parámetros Pressupost”.

NOTA: Para habilitar/deshabilitar generé una función (ya que es un proceso que se realiza varias veces) denominada “habilitarDeshabilitar” que recibe 3 parámetros de tipo “boolean”. Cada parámetro hace referencia al menú superior, inferior y a las cajas del texto inferior.

Nos queda la configuración del apartado inferior “Import Pressupost”. Este apartado tiene 2 campos de texto donde introducir valores de tipo entero, “Tipus Llicència → Los valores aceptados son 1, 2, y 3” y “Sortida domicili → Los valores aceptado son 0 y 1 (sin salida o con salida). Al pulsar el botón “Calcular (está a la escucha)” se evalúan ambos datos a través de la función “compruebaLlicenciaSortida” que recibe dos parámetros de tipo entero. Evalúa si los datos introducidos son correctos (están en rango, son enteros), si son incorrectos, se pone el foco en el campo erróneo y se selecciona todo el campo, también muestra un mensaje en el campo “textMensaje” indicando si el valor no es un entero o si no están en rango. Si ambos datos son correctos rellena los campos “Preu asistencia → es el precio de la hora por el número de horas”, “Tipus Llicencia → muestra un texto con el software instalado”, “Import Llicencia → muestra el precio de la licencia según la opción seleccionada (1, 2 o 3)”. “Import domicili → Muestre el precio si se ha salido a domicilio o no”, por último el campo “Import Total → Muestra el precio final, es decir, la suma de Preu asistencia + import Llicencia + import domicili, lo hace a través del método << calcularImportTotal() >>”.

Nos queda el botón “**Restrablir**” (está a la escucha) → Si pulsamos en el botón restablir se resetea todo (variables y campos de texto) y se deshabilita el menú inferior y habilita el menú superior, para que se vuelva a empezar.

Como dato adicional he generado una serie de contantes para facilitar la lectura del código. Entre otras a destacar “PREU_HORA, NUM_HORES, SORTIDA_DOMICILI y TIPUS_LLICENCIA”. Estas constantes hacen uso del método “matches” dentro del bloque “catch” si alguno de los valores no es un valor numérico. Los utilizo para poner el foco una vez salte el catch.

Ejercicio 17

En este ejercicio necesitamos tener instalado y configurado el servidor “Apache Tomcat”, tanto en el sistema operativo como en Eclipse. En eclipse hay que añadirlo a través del “Eclipse Marketplace”.

Este ejercicio indica que debemos realizar una aplicación cliente-servidor haciendo uso de un “servlet”. Para realizarlo seguí las instrucciones del temario paso a paso, pero podríamos destacar los siguiente:

Consta de un fichero **java** (servlet), un fichero **xml** y un fichero **html**. El fichero HTML muestra el contenido web (formulario, título, etc...), el fichero java se encarga de enlazar con el fichero HTML y el fichero XML.

Sólo tuve que hacer el cambio en el nombre del proyecto, del paquete y modificar el fichero XML en la etiqueta “servlet-class” que apunta al paquete.nombreFicheroHtml.

Se puede ver el funcionamiento clicando con el botón derecho del fichero “html → Run as → Run on server” o desplegando la aplicación “war” a través del servidor TomCat por el navegador.

