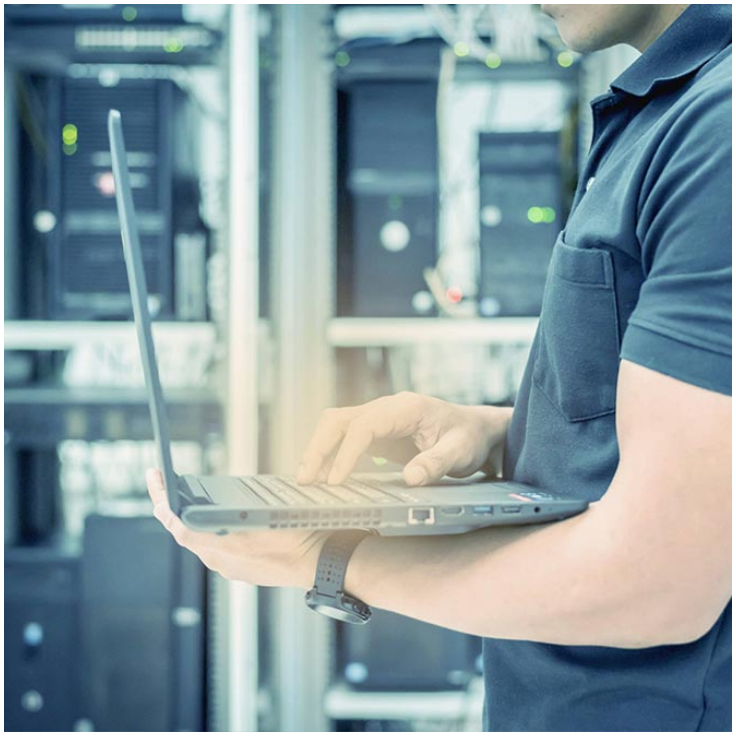
Curso de Programación Orientado a Componentes (180 horas)



Categoría: Informática, Programación y Comunicaciones
Página del curso: https://www.ensenanzadual.com/cursos/curso-de-programacion-orientado-a-componentes-180-horas/

Objetivo

Con el Curso de **180 horas** de **Curso de Programación Orientado a Componentes** adquirirás los conocimientos necesarios para cambiar de carrera o actualizar tus competencias profesionales, no lo dudes más y solicita información sobre este curso.

Descripción

UNIDAD FORMATIVA 1. DISEÑO DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES UNIDAD DIDÁCTICA 1. LA ORIENTACIÓN A OBJETOS.

Principios de la orientación a objetos. Comparación con la programación estructurada:

- - Ocultación de información (information hiding). - - El tipo abstracto de datos (ADT). Encapsulado de datos. - - Paso de mensajes.

Conceptos básicos de orientación a objetos:

- Clases:
  - Atributos, variables de estado y variables de clase. - Métodos. Requisitos e invariantes. - Gestión de excepciones. - Agregación de clases.

- Objetos:
  - Creación y destrucción de objetos. - Llamada a métodos de un objeto. - Visibilidad y uso de las variables de estado. - Referencias a objetos. - Persistencia de objetos. - Optimización de memoria y recolección de basura (garbage collection).

- Herencia:
  - Concepto de herencia. Superclases y subclases. - Herencia múltiple. - Clases abstractas. - Tipos de herencia: herencia de implementación, herencia de interfaces y de tipos y otros tipos de herencia. - Polimorfismo y enlace dinámico (dynamic binding). - Directrices para el uso correcto de la herencia.

- Modularidad:
  - Librerías de clases. Ámbito de utilización de nombres. - Ventajas de la utilización de modulos o paquetes.

- Genericidad y sobrecarga:

- Concepto de genericidad. - Concepto de Sobrecarga. Tipos de sobrecarga. - Comparación entre genericidad y sobrecarga.

Desarrollo orientado a objetos:

- - Lenguajes de desarrollo orientado a objetos de uso común. - - Herramientas de desarrollo.

Lenguajes de modelización en el desarrollo orientado a objetos:

- - El lenguaje unificado de modelado (UML). - - Diagramas para la modelización de sistemas orientados a objetos. **UNIDAD DIDÁCTICA 2. LA ORIENTACIÓN A COMPONENTES.**

Fundamentos conceptuales:

- - Definición de componente. - - Comparación entre componentes y objetos. - - Módulos.

- Interfaces:

- Tipos de interfaces. - Versionado de interfaces. - Interfaces como contratos. - - Escalado de componentes. - - Estado de componentes.

Arquitecturas de componentes:

- - Basadas en objetos. Composición y uso de objetos. - - Multicapa. - - Basadas en middleware. - - Basadas en objetos distribuidos.

Diseño de componentes:

- Principios de diseño de componentes:

- Dependencias no cíclicas. - Principio □open/closed□.Reusabilidad. - Configurabilidad. - Abstracción. - Dependencias.

- Técnicas de reusabilidad:

- Patrones. - Librerías. - Interfaces. - Protocolos y esquemas de mensajes. - Uso de lenguajes de programación. - Estructuras y jerarquías de estructuras. -

Arquitecturas de sistemas.

- Modelo de componente:
  - Especificación de servicios: transacciones, seguridad, persistencia y acceso remoto. - Especificación de Interface. - Especificación de la implementación. - Especificación de las unidades de despliegue (modulos).

- Modelos de integración de componentes:
  - Referencias e identidad de objetos, componentes e interfaces. - Servicios de localización. - Modelos de intercambio: objetos distribuidos, capa intermedia (Middleware) e interacción e integración mediante servicios web. - Comparación entre métodos de intercambio en las principales infraestructuras de componentes: OMG: CORBA, OMA, Java: JavaBeans, EJBs y Microsoft: COM, OLE/ActiveX, .NET

- Diagramación y documentación de componentes:
  - Modelo de información: diagramas conceptuales, diagramas de arquitectura de componentes y diagramas de despliegue. - Modelo dinámico: diagramas de interacción y de actividad, diagramas de casos de uso y diagramas de estado.

UNIDAD FORMATIVA 2. IMPLEMENTACIÓN E INTEGRACIÓN DE ELEMENTOS SOFTWARE CON TECNOLOGÍAS BASADAS EN COMPONENTES UNIDAD DIDÁCTICA 1. DESARROLLO DE COMPONENTES.

Lenguajes de desarrollo de componentes.:
  - - Comparativa con lenguajes orientados a objetos.
- Lenguajes orientados a componentes:
  - Descripción de intefaces. - Ensamblado. - Descripción de arquitectura.

Requisitos principales del desarrollo orientado a componentes:
  - - Modularidad - - Despliegue independiente. - - Reemplazabilidad. - - Seguridad. - - Separación entre interfaz e implementación.

Infraestructuras (frameworks) de componentes:
- Modelos de infraestruturas de componentes:
    - Orientados a conexión. - Orientados a contexto. - Orientados a aspectos.

- Descripción de las intraestructuras de componentes de uso común:
    - OMG: CORBA, OMA. - Java: JavaBeans, EJBs. - Microsoft: COM, OLE/ActiveX, .NET

Métodos de desarrollo de componentes:
    - - Uso de lenguajes orientados a objetos. - - Selección de infraestructuras de componentes.

Construcción de software mediante componentes:
    - - Definición de interfaces. Lenguajes de descripción de interfaces. - - Reutilización de componentes. - - Técnicas de ensamblado en infraestructuras de uso común.

Técnicas específicas de desarrollo:
    - - Componentes en la capa de servidor web. Páginas dinámicas. - - Componentes en la capa de servidor de aplicaciones.
- Componentes en la capa de aplicación cliente:
    - Componentes de interfaz gráfico. - Componentes orientados a documento. - - Componentes en la capa de servicios web. - - Componentes para dispositivos móviles.

Herramientas para el desarrollo de componentes:
    - - Entornos integrados de desarrollo de componentes.
- Configuración e instalación de herramientas de uso común:
    - Entorno Java. - Entorno .NET

- Gestión del ciclo de vida en el desarrolo de componentes mediante herramientas de uso común:

- Uso de repositorios de componentes. Registro de componentes. - Reutilización de componentes para la construcción de sistemas software. - Definición de metadatos de componente. Descriptores de interfaces. - Modelo de seguridad. - Instalación de componentes. - Depuración y prueba de componentes. **UNIDAD DIDÁCTICA 2. COMPONENTES DISTRIBUIDOS.**

Programación distribuida en infraestructuras de uso común:

- - Programación multihilo (multithreading). - - Comunicaciones síncronas y asíncronas.

Modelos de intercambio:

- - Llamadas a procedimientos remotos. - - Orientados a mensajes. - - Orientados a recursos.

Información adicional

- Online: Si
- Tipo: Profesiones
- Horas: 180
- Unidades: 4