

Aprende a programar resumen

Last updated by | Christian Ibarra < cic.innsy@gmail.com > | Wed, 03 Oct 2018 00:21:34 GMT

Contents

- [Resumen](#)
 - [Aprende a programar](#)
- [Teoria](#)
- [Conceptos](#)
- [Herramientas](#)
 - [Visual Studio 2017](#)

Resumen

Aprende a programar

- Presentación e instalación del software necesario 5 clases
- Introducción a C# 10 clases
- Clases y Objetos 12 clases
- Assemblies y Referencias. 7 clases
- Variables. Tipo Valor y Tipo Referencia 6 clases
- Métodos, campos y propiedades 3 clases
- Estructuras de control

Teoria

Funcion computable SSI

El teorema del programa estructurado es un resultado en la teoría de lenguajes de programación. Establece que toda función computable puede ser implementada en un lenguaje de programación que combine sólo tres estructuras lógicas. Esas tres formas (también llamadas estructuras de control) específicamente son:

Secuencia: ejecución de una instrucción tras otra.

Selección: ejecución de una de dos instrucciones (o conjuntos), según el valor de una variable booleana.

Iteración: ejecución de una instrucción (o conjunto) mientras una variable booleana sea 'verdadera'. Esta estructura lógica también se conoce como ciclo o bucle.

Conceptos

Desarrollo de software

Programación vs Desarrollo de Software: ¿alguna diferencia?

Programadores

Un programador usa diferentes lenguajes de programación (el proceso es conocido como "escribir un código") para

desarrollar estos programas. Estos lenguajes comúnmente son, pero no siempre, autónomos de algún propietario y/o fabricante. Un buen programador tiende a ser paciente, altamente orientado a los detalles, y excelente en la solución de problemas.

Desarrolladores de Software

Los ingenieros de software usualmente poseen las mismas habilidades que un programador, y técnicamente también "programan". Sin embargo, en la industria AV, su rol se enfoca en el desarrollo de soluciones, productos o creaciones personalizadas y de un solo uso que solucionen problemas, a diferencia de los programadores, quienes crean programas más fundamentales para apoyar la interacción del usuario.

Los desarrolladores de software son, a menudo, generalistas en una amplia variedad de áreas relacionadas con software.

<https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/programaci%C3%B3n-vs-desarrollo-de-software-alguna-diferencia>

Depuración

La depuración de programas es el proceso de identificar y corregir errores de programación. En inglés se conoce como debugging, porque se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación.

Pruebas

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder. Es una actividad más en el proceso de control de calidad.

Colaboración

Paradigma de programación

Un paradigma de programación es un marco conceptual, un conjunto de ideas que describe una forma de entender la construcción de programa, como tal define

Las herramientas conceptuales que se pueden utilizar para construir un programa (objetos, relaciones, funciones, instrucciones). Las formas válidas de combinarlas.

Abstracción

La abstracción, acto mental en el que se aísla conceptualmente una propiedad o función concreta de un objeto, y se piensa qué es, ignorando otras propiedades del objeto en cuestión.

La abstracción; en psicología, proceso que implica reducir los componentes fundamentales de información de un fenómeno para conservar sus rasgos más relevantes con el objetivo de formar categorías o conceptos.

La abstracción; en computación, aislamiento de un elemento de su contexto o del resto de los elementos que lo acompañan.

La capa de abstracción; en computación, manera de ocultar los detalles de implementación de ciertas funcionalidades.

La inversión de abstracción; en computación, antipatrón que tiene lugar cuando una interfaz no expone las funcionalidades requeridas por los usuarios.

Software

Esto incluye software de aplicación como un procesador de textos, que permite a un usuario realizar una tarea, y software del sistema como un sistema operativo, que permite que otro software se ejecute correctamente, al interactuar

con el hardware y con otro software.

Los sistemas informáticos prácticos dividen el software en tres clases principales: software de sistema, software de programación y software de aplicación.

IDE (Integrated Development Environment)

Los desarrolladores utilizan numerosas herramientas a lo largo de la creación, creación y prueba de códigos de software. Las herramientas de desarrollo a menudo incluyen editores de texto, bibliotecas de código, compiladores y plataformas de prueba. Sin un IDE, un desarrollador debe seleccionar, implementar, integrar y administrar todas estas herramientas por separado.

Un IDE reúne muchas de esas herramientas relacionadas con el desarrollo como un marco, aplicación o servicio único. El conjunto de herramientas integrado está diseñado para simplificar el desarrollo de software y puede identificar y minimizar errores de codificación y errores tipográficos.

Un IDE puede mejorar la productividad de los desarrolladores de software gracias a la rápida configuración y estandarización de las herramientas.

Control de versiones (**git**)

¿Qué es el "control de versiones" y por qué debería importarte? El control de versiones es un sistema que registra los cambios en un archivo o conjunto de archivos a lo largo del tiempo para que pueda recuperar versiones específicas más adelante.

Herramientas

Visual Studio 2017

Funcionalidad

IDE de Visual Studio IDE para Windows y Mac

Desarrollar aplicaciones para Android, iOS, Mac, Windows, la Web y la nube

Escribir código con rapidez

Depurar y emitir diagnósticos fácilmente

Realizar pruebas periódicas y publicar con confianza

Extender y personalizar según sus preferencias

Colaborar de manera eficiente