INSTITUTO BARRANQUERAS

Paradigmas de Programación

Programar es mucho más complejo que definir una serie de instrucciones para ser ejecutadas por una computadora. La programación generalmente es tan diversa como uno se lo pueda imaginar. A esta diversidad en el mundo de la programación se le conoce como paradigmas o estilos de programación. Las características de un lenguaje de programación dictaminan el paradigma al cual pertenece.

Programación Imperativa y Programación Declarativa

DECLARATIVA

La programación declarativa un programa se describe en términos de proposiciones y afirmaciones que son declaradas para describir el problema, sin especificar los pasos para resolverlo; en este tipo de programas, el estado no puede ser modificado ya que todos los tipos de datos son inmutables.

Consiste en decirle a un programa lo ***qué*** tiene que hacer en lugar de decirle ***cómo*** debería hacerlo. En la práctica, este enfoque implica proporcionar un lenguaje específico de dominio (DSL, del inglés *Domain-specific lenguage*) para expresar lo ***qué*** el usuario quiere y resguardarlos de las arquitecturas o construcciones de bajo nivel (bucles, condicionales, asignaciones) que materializan el estado final deseado.

La programación declarativa es un término que agrupa los siguientes paradigmas de programación:

- Programación lógica: Los problemas se representan por medio de lógica matemática.
- Programación funcional: Todo se resuelve por medio de la evaluación de funciones matemáticas. Es un estilo de programación cuyo método básico es la aplicación de funciones a sus argumentos.
- Lenguajes de dominio específico (DSLs). Lenguajes descriptivos para un propósito específico, tales como HTML, CSS («Hojas de estilo en cascada», es un lenguaje de diseño gráfico) y SQL (el lenguaje SQL sirve para el acceso a la información almacenada en las bases de datos. Es un lenguaje sencillo de consulta, que permite realizar operaciones de selección, inserción, actualización y borrado de datos, así como operaciones administrativas sobre las bases de datos).

U.E.G.P N° 161

INSTITUTO BARRANQUERAS

IMPERATIVA

Se basa en dar instrucciones a la computadora de cómo hacer las cosas en forma de algoritmos. En este paradigma se expresa **como debe solucionarse** un problema especificando una **secuencia de acciones** a realizar a través de uno o más procedimientos denominados subrutinas o funciones.

Los distintos lenguajes de programación imperativa pueden clasificarse a su vez en tres estilos distintos de programación subordinados: el procedimental, el modular y el estructurado.

Programación Procedimental

Divide las tareas de las que se debe ocupar un programa en tareas parciales más pequeñas que se describen en el código por separado. Estos se denominan como procedimientos, dependiendo del lenguaje de programación, o también como subprogramas, rutinas o funciones. El sentido y el propósito de esta distribución es hacer que el **código de programa sea más claro** y evitar las **repeticiones innecesarias de código**. De esta forma, se crean principios básicos de programación que también se pueden reutilizar en otros programas.

Programación Modular

Cada uno de los componentes de programa se diseña, desarrollan y prueban con total independencia los unos de los otros. El código fuente se divide específicamente en bloques parciales lógicos independientes los unos de los otros para proporcionar más **transparencia** y facilitar el **proceso de *debugging*** (resolución de errores). Los bloques parciales individuales, denominados **módulos**, se pueden probar por separado antes de vincularlos posteriormente a una aplicación conjunta.

Programación Estructurada

A medida que un programa aumenta de tamaño, aumenta a su vez la complejidad para leerlo, su tiempo de desarrollo, de mantenimiento y disminuye su calidad si no se tiene un correcto orden y estructura del programa.

Es por esto que surgió un paradigma de programación llamado programación estructurada, que consiste en mejorar la claridad, calidad y acelerar el tiempo de

desarrollo, utilizando únicamente subrutinas y 3 estructuras de control: Secuencial, de Selección (IF y SWITCH) y de Iteración (ciclos FOR y WHILE).

La programación estructurada es una teoría de programación que consiste en construir programas de fácil comprensión, es especialmente útil, cuando se necesitan realizar correcciones o modificaciones después de haber concluido un programa o aplicación. Al utilizar la programación estructurada, es mucho más sencillo entender la codificación del programa, que se habrá hecho en diferentes secciones.

Se basa en una metodología de desarrollo de programas llamada refinamiento sucesivos: Se plantea una operación como un todo y se divide en segmentos más sencillos o de menor complejidad, una vez terminado todos los segmentos del programa, se procede a unificar las aplicaciones realizadas por el grupo de programadores. Si se ha utilizado adecuadamente la programación estructurada, esta integración debe ser sencilla y no presentar problemas al integrar la misma, y de presentar algún problema, será rápidamente detectable para su corrección.

La representación gráfica de la programación estructurada se realiza a través de diagramas de flujo, el cual representa el programa con sus entradas, procesos y salidas.

La programación estructurada propone segregar los procesos en estructuras lo más simple posibles, las cuales se conocen como secuencia, selección e interacción, que están disponibles en todos los lenguajes modernos de programación imperativa en forma de sentencias, combinando esquemas sencillos se pueden llegar a construir sistemas amplios y complejos pero de fácil entendimiento.

La programación estructurada es un método disciplinado de escribir programas que sean claros, que se demuestre que sean correctos y fáciles de modificar.

La programación estructurada consiste en dividir los programas en módulos y se basa en el desarrollo de programas que van de lo general a lo particular, es decir, del conjunto al elemento, es decir de un todo a lo específico.

Para la solución de un problema en particular, se inicia considerando las funciones que tiene que cumplir el programa en general y después se va desmembrando estas funciones en subfunciones más pequeñas hasta llegar al caso último o más particular y que ya no se pueda subdividir en casos más pequeños. Una vez que ya se tiene el programa desmembrado de lo general a lo particular, se empieza a programar estas funciones pequeñas, particulares o módulos, de esta manera, siempre podremos construir nuevos

módulos o unidades insertando el nombre del módulo donde corresponda y desarrollándolo a parte.

La modificación de los módulos es más fácil y se pueden referenciar cuantas veces se requiera, con lo que se ahorra tiempo en la programación, un programa tiene un diseño estructurado si cumple las dos siguientes condiciones:

- El teorema de Estructura.
- Está debidamente documentado

El teorema de Estructura dice que "un programa cumple el teorema de estructura si y sólo (ó) si es propio y contiene únicamente las tres estructuras básicas de control" que son la secuencial, la alternativa y la repetitiva, un programa es propio si y sólo si cumple: que tenga un solo punto de entrada y un solo punto de salida y que entre dos puntos de control del programa exista al menos un camino.

La programación estructurada es un estilo con el cual él se busca que el programador elabore programas sencillos y fáciles de entender, la programación estructurada hace uso de tres estructuras básicas de control que son: Estructura Secuencial, Estructura Selectiva y la Estructura Repetitiva (ó Iterativa)

La programación estructurada se basa un teorema fundamental, el cual afirma que cualquier programa, no importa el tipo de trabajo que ejecute, puede ser elaborado utilizando únicamente las tres estructuras básicas.

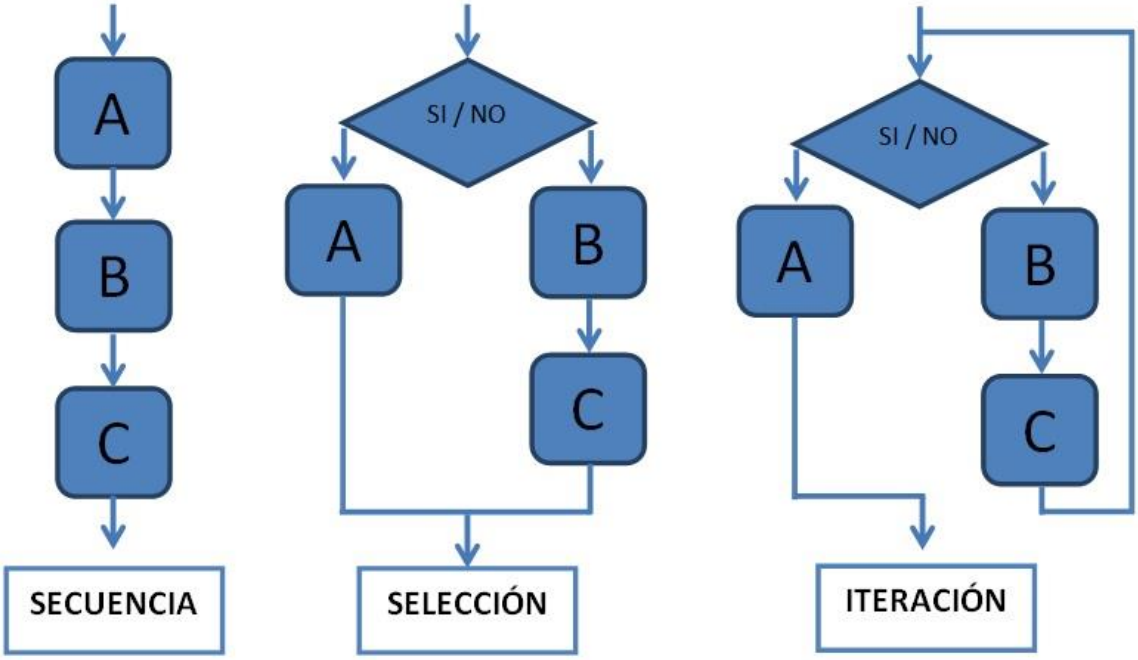
DEFINICIÓN DE LAS 3 ESTRUCTURAS BÁSICAS

1. Estructura Secuencial: Indica que las instrucciones de un programa se ejecutan una después de la otra, en el mismo orden en el cual aparecen en el programa. Se representa gráficamente como una caja después de otra, ambas con una sola entrada y una única salida. (las cajas A y B pueden ser definidas para ejecutar desde una simple instrucción hasta un módulo o programa completo, siempre y cuando éstos también sean programas apropiados)**.**

2. Estructura Selectiva: También conocida como la estructura: verdadero - falso, plantea la selección entre dos alternativas con base en el resultado de la evaluación de una condición; equivale a la instrucción IF de todos los lenguajes de programación (en el diagrama de flujo anterior, C es una condición que se evalúa; A es la acción que se ejecuta cuando la evaluación de esta condición resulta verdadera y B es la acción ejecutada

cuando el resultado de la evaluación indica falso. La estructura también tiene una sola entrada y una sola salida; y las funciones A y B también pueden ser cualquier estructura básica o conjunto de estructuras).

3. **Estructura Repetitiva (Iterativa):** También llamada la estructura hacer – mientras - que, corresponde a la ejecución repetida de una instrucción mientras que se cumple una determinada condición (aquí el bloque A se ejecuta repetidamente mientras que la condición C se cumpla o sea cierta. También tiene una sola entrada y una sola salida; igualmente A puede ser cualquier estructura básica o conjunto de estructuras).



Ventajas de la Programación Estructurada

- Como la programación estructurada está compuesta por segmentos bien definidos, los programas son más simples, rápidos y fáciles de entender, pueden ser leídos de forma secuencial y se pueden hacer las correcciones o modificaciones después de haber concluido.
- La estructura de los programas es clara debido a que las instrucciones están más relacionadas entre sí.
- Los códigos son reutilizables para futuras aplicaciones.

- Se pueden hacer las modificaciones o correcciones de los programas de manera más sencilla.
- Reducción del esfuerzo en las pruebas y depuración de errores; ya que se puede encontrar y leer las secuencias más fácilmente y así se reducen también los costos durante el mantenimiento.
- Permite escribir los códigos más fácil y rápidamente incrementando así la eficiencia de los programadores.

Desventajas de la Programación Estructurada.

- Complejidad para adaptarse.
- Mayor cantidad de código (aunque a la larga no, por la reutilización).

Conclusión Imperativo-Declarativo

Si al escribir un programa de cómputo lo que hacemos es explicarle a la computadora por medio de instrucciones detalladas de "cómo hay que realizar una tarea", entonces estamos programando en forma imperativa. Es decir, estamos dando los pasos o conjunto de instrucciones necesarias para resolver un problema. Por otro lado, si al escribir un programa describiendo "qué hay que hacer", entonces estamos programando en forma declarativa. Es decir, describimos el problema que queremos solucionar, pero no las instrucciones necesarias para resolverlo.

Ejemplo

Problema: Llegar a casa desde la Escuela

• Imperativo Sal por la puerta, cruza a la acera de enfrente, gira a la izquierda, camina hasta el final de la calle, ...

• Declarativo La dirección es: calle ..., número ....

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

U.E.G.P N° 161

INSTITUTO BARRANQUERAS

¿Qué softwares de programación existen?

Por software de programación entendemos el conjunto de todas las herramientas que le permiten al programador, crear, escribir códigos, depurar, mantener y empaquetar los proyectos.

Algunos de los distintos programas por los que pasará el proyecto para gestionarlo son:

Editores de código o texto

Al escribir los códigos se auto-completan marcando los errores sintácticos y la refactorización.

Compiladores

Como mencionados anteriormente, éstos traducen el código ingresado a lenguaje de máquina generando un código binario ejecutable.

Depuradores

Sirven para optimizar el tiempo de desarrollo mediante el monitoreo de la ejecución de un programa, el seguimiento a los valores de ciertas variables, las referencias a objetos en memoria y por ende, nos ayuda a corregir errores.

Enlazadores

Este programa toma objetos generados en los primeros pasos del proceso de compilación y los recursos necesarios de la biblioteca, quita aquellos procesos y datos que no necesita, y enlaza el código con dicha biblioteca para así aumentar su tamaño y extensión.

Interpretadores o traductores

Como leíste en éste artículo, el traductor (o intérprete) carga el código ingresado y traduce las instrucciones para que el programa pueda ser ejecutado.

IDE

El IDE (Integrated Development Environment) o **Entorno de Desarrollo Integrado**, es una aplicación informática que proporciona una serie de servicios que facilitan la programación de software, tales como:

- funciones de autocompletado;
- un editor de código fuente;
- gestión de conexiones a bases de datos;
- integración con sistemas de control de versiones;
- simuladores de dispositivos;
- un depurador para agilizar el proceso de desarrollo de software, entre otros.