



UNIVERSIDAD DEL SURESTE: DE LA FRONTERA COMALAPA.

DOCENTE: Ing. Icel Bernardo Lepe Arriaga.

ASIGNATURA: Programación lógica.

ALUMNO: Ramiro Gerardo Resendíz Valdéz.

CUATRIMESTRE: Octavo (8^{vo}).

CARRERA: Ingeniería en sistemas computacionales.

GRUPO: ISC13SDC0220-A.

UNIDAD: Primera (1^{ra}).

TRABAJO: Mapa conceptual de la unidad uno de la antología.



FECHA DE ENTREGA: 22/Enero/2023.

Conceptos fundamentales.

Estilos de programación.

Para asegurar la portabilidad del código, se recomienda usar sólo el estándar de Fortran 77. La única excepción que se ha hecho en este manual es usar letras minúsculas.

Se debe siempre usar el sangrado apropiado para bloques de ciclos y sentencias como se mostró en el tutorial.

La estructura total del programa deberá ser modular. Cada subprograma deberá resolver una tarea bien definida. Mucha gente prefiere escribir cada subprograma en un archivo por separado.

Se repite lo que se había indicado previamente: Escriba código legible, pero también agregue comentarios al código fuente para explicar lo que se está haciendo.

Evaluación de expresiones.

Toda expresión regresa un valor. Si hay más de un operador, se evalúan primero operadores mayor precedencia, en caso de empate, se aplica regla asociatividad.

Las sentencias de control de flujo determinan el orden en que se ejecutarán las otras sentencias dentro del programa, el lenguaje Java soporta varias sentencias de control de flujo.

Toda variable utilizada en una expresión debe tener un valor almacenado para que la expresión, al ser evaluada, dé como resultado un valor.

El orden general de evaluación de los operadores de una expresión va de izquierda a derecha, con la excepción de las asignaciones que lo hacen de derecha a izquierda.

Definición de funciones.

Si no tienes ninguna preferencia personal de estilo, recomendamos el estilo de programación del núcleo de Linux o el estilo de programación de GNU.

Cuando escribas un nuevo programa o biblioteca, sigue un estilo consistente de ubicación de llaves y de indentación.

Los tabuladores de 8 espacios para indentación también ayudan al diseño de funciones que encajen bien en la pantalla, lo cual significa que las personas puedan entender el código sin tener que desplazarse atrás y adelante para entenderlo.

tabuladores de 8 espacios para indentación proporciona un número de beneficios.

Disciplina de tipos.

Haskell y otros lenguajes funcionales utilizan el sistema de tipos de Milner, que tiene dos características fundamentales; disciplina estática de tipos y Polimorfismo.

En los lenguajes de programación con disciplina de tipos, cada tipo representa una colección de valores o datos similares.

Pascal, cercano a tener disciplina de tipos pero no realiza comprobación de tipos en los registros variantes incluso puede omitirse la etiqueta discriminadora en dichos registros.

Ada, resuelve el problema de los registros variantes realizando comprobación dinámica de tipos (sólo en este caso).

Tipos de datos.

Un tipo de datos define un conjunto de valores y las operaciones sobre estos valores, casi todos los lenguajes de programación explícitamente incluyen la notación del tipo de datos.

Tipo de dato carácter (Char) Es cualquier signo tipográfico, puede ser una letra, un número, un signo de puntuación o un espacio, este término se usa mucho en computación.

Un tipo de dato entero en computación es un tipo de dato que puede representar un subconjunto finito de los números enteros.

En lenguajes de programación un tipo de dato es un atributo de una parte de los datos que indica al ordenador o al programador, algo sobre la clase de datos sobre los que se va a procesar.