Clase 3

Entidades con las que trabajan los lenguajes

Cada entidad posee uno o más atributos

- Variables Atributos: nombre tipo, área de memoria, valor, etc.
- Rutinas (Funciones o procedimiento) Atributo: nombre, parámetros
- Sentencias Atributo: acción que realiza la sentencias.

Descriptor: Lugar donde se almacenan los atributos y las entidades.

¿Momento? Si es compilado, en la compilación. Algunas cosas se completan en ejecución como el valor de las variables. Sino se hace dinámicamente.

Ligadura o binding: Momento en que a un atributo de una entidad se le asocia un valor determinado.

Estabilidad: una vez que ese valor es establecido. ¿Queda fijo o se puede modificar?

Estabilidad del binding

- **Estática**: Se establece antes de la ejecución (definición del lenguaje, implementación de lenguaje o compilación) y no se puede cambiar. Estático por el momento del binding y su estabilidad.
- **Dinámica**: Se establece en el momento de ejecución y puede cambiarse de acuerdo a alguna regla especifica del lenguaje. (Excepción: constantes)
  - **Ejecución**: Momento en que se le da un valor a esa variable.

Momento de ligadura

- En definición: Cuando se establecen en nombre o cuales van a ser las sentencias, cuales van a ser y que nombre se le van a dar los tipos predefinidos.
- En implementación: Representación de esos valores enteros. Por ejemplo = tamaño
- En compilación: Asignación del tipo a las variables.

Variable

- Celda de memoria que ocupa una dirección, la cual posee una sentencia de asignación (la más importante) que me permite cambiar su valor, dicho valor posiblemente cambie, salvo que sea una constante.
- Hablando de forma abstracta, una variable es una celda de memoria, su nombre sería una dirección y la sentencia de asignación es la modificación destructiva del valor.

Atributos

Nombre: Identificación de la variable.

- Aspectos de diseño = Si posee una longitud máxima, que caracteres son aceptados y que sin sensibles a la mayúscula/minúscula.

Alcance: Rango de instrucción en que se conoce el nombre. Relacionada con la visibilidad.

- **Reglas de alcance**: Sirven para saber a que variable hago referencia cuando no es local tengo las siguientes opciones. Depende del lenguaje cual elija.
    - Alcance estático (llamado alcance léxico): Se fija en la estructura del programa. Puede ligarse estáticamente a una declaración (implícita o explicita) examinando el texto del programa, sin necesidad de ejecutarlo. En simples palabras el alcance depende de donde este declarada la variable. Usada por la mayoría de los lenguajes.
    - Alcance dinámico: Cada declaración de variable extiende su efecto sobre todas las instrucciones ejecutadas posteriormente, hasta que una nueva declaración para una variable con el mismo nombre es encontrada durante la ejecución. En vez de ver la estructura se va fijando las instrucciones anteriores hasta encontrar la declaración de esa variable. Por ejemplo: API, PERL. Desventaja: Menos legible, porque el seguimiento para saber de dónde viene. Mas fáciles de implementar.

Tipo de dato: Conjunto de valores y operaciones permitidas.

Chequeo de tipos: verifica el uso correcto de las variables.

Tenemos 3 tipos de datos

- Predefinido: Los que vienen con el lenguaje (tipo base)
- Definidos por el programador a partir de los predefinidos y los constructores.
- Tipos de datos abstractos TAD: El programador debe especificar la representación y las operaciones

Momento de ligadura estática: Durante la compilación

- Explícito: Se establece mediante una declaración. Ej.: int z;
- Inferido: El tipo de una expresión se deduce de los tipos de su componente. Ej.: Lenguajes funciones. double x = 2 * x
- Implícito: La ligadura se deduce por regla. Ej.: Fortran.

Momento de ligadura dinámico: Durante la ejecución. Mas flexible, pero más costoso en ejecución. Los cheques son dinámicos y hay menor legibilidad.

L-valor (izquierdo): Lugar de la memoria asociado a la variable

Tiempo de vida: Periodo de tiempo que existe la ligadura. Periodo de tiempo en que la variable este alocada en memoria.

Alocación: Momento que se reserva la memoria. Según el momento:

- Estática: La variable tiene un lugar de memoria cuando el programa se carga.
- Dinámica: Cuando arranca el programa se sabe donde va a estar alocada esa variable.
    - Automática: Aparece la declaración
    - Explicitas: a través de algún constructor. Ej.: Punteros
- Persistentes: su tiempo de vida no depende de la ejecución. Ej.: Archivos, base de datos.

R-Valor (derecho): Valor que se le da a la variable

Valor almacenado en el l.-valor de la variable. Se interpreta de acuerdo al tipo de la variable.

Momento

- Dinámico: por naturaleza
- Contantes: se congela ahí.

¿Cuál es el r-valor luego de crear la variable? Dos alternativas

- Se ignora el problema: se pone lo que este en la memoria.
- Estrategia de inicialización: se inicializan por defecto en un valor o la se inicializa en la declaración

Cosas a tener en cuenta

- Algunos lenguajes permiten que el r-valor de una variable sea una referencia al l-valor(dirección) de otra variable
- Alias: Dos variables son alias si comparten el mismo objeto de dato en el mismo ambiente de referencia. El uso de alias puede llevar a programas de difícil lectura y errores (si cambia uno cambia los dos, tener cuidado). punteros
- Sobrecarga : Es la capacidad de un lenguaje de programación, que permite nombrar con el mismo identificador diferentes variables u operaciones

Clasificación de variables de acuerdo al alcance

Local: Referencias que se han creado dentro del programa o subprograma.

No Local: Referencias que se utilizan dentro del subprograma, pero no han sido creadas en él.

Global: Referencias creadas en el programa principal.