**Концепция переменной**

Это центральное понятие императивных языков программирования (переменная – это абстракция области памяти)

Переменная характеризуется атрибутами, которые обычно включают:

* имя
* тип переменной (область допустимых значений)
* множество операций, допустимых на данной переменной
* размер
* тип данных
* адрес
* область видимости
* время жизни
* значение
* время связывания
* область связывания (область видимости)

Связывание – это ассоциирование какого-либо атрибута имени.

Примеры, когда связывание может произойти:

1. При проектировании языка (константа с именем True, связанная булевским значением «Истина»)
2. При реализации языка (при написании компилятора различные виды связываются с их внутренним представлением)
3. Во время компиляции (тип переменной, в статически типизированных языках, определяется в это время, как и ссылки на перегружаемые функции)
4. Во время загрузки (выполняемый образ программы загружается в память для выполнения, например, в это время определяются адреса глобальных переменных)
5. Во время вызова процедур или функций, это происходит во время выполнения программы (фактические параметры привязываются к формальным параметрам; локальные переменные привязываются к их адресам во время вызова)
6. Во время выполнения программы (новое значение может быть привязано к переменной; в языках с динамической типизацией ссылки (адреса) на перегружаемые функции определяются во время выполнения; тип переменной определяется во время выполнения)

**Область связывания (область видимости)**

Правила определения областей связывания тесно объединены с концепцией времени связывания.

Правила бывают статические и динамические.

Статические – работают на этапе компиляции и основаны на лексической структуре языка.

Динамические – смотри ООП

|  |  |
| --- | --- |
| Вид переменной | Время жизни переменной |
| Глобальные | От начала до конца работы программы |
| Локальные | От создания её внутри блока до завершения работы блока |
| Динамические  (ссылочного типа) | От вызова команды распределения переменной в куче до момента освобождения этой переменной |

**Время жизни**

**Тип данных**

Типы – это абстракции, которые представляют множество допустимых значений, а также операции и взаимосвязи, применимые к ним (это доменное определение типа, знать наизусть)

Для ЭВМ память – однородная совокупность разрядов без какой-либо структуры.

Тип – шаблон, накладываемый на области памяти для того, чтобы интерпретировать эту совокупность разрядов каким-либо образом (тоже знать)

Тип определяет область допустимых значений переменной; а также множество значений, к которому принадлежит константа, которая принимает выражение или формирует функция.

Тип значения, задаваемого константой, переменной, функцией или выражением можно определить по их виду или описания без необходимости выполнять какие-либо вычисления.

Каждая операция или функция требует аргументы определённого типа и даёт результат так же фиксированного типа. Если операция допускает элементы нескольких типов (операция + допускает аргументы целого и вещественного типа), то тип результата определяется по специальным правилам языка.

Транслятор может использовать информацию о типах для проверки вычислимости и правильности различных конструкций.