Principios de Lenguajes de Programación

Tipo de Datos: Registro

Facultad de Informática
Universidad Nacional del Comahue

Primer Cuatrimestre

Índice

- Registros
  - Definición
  - Lenguajes ejemplos
  - Ejemplos

Registros

- Un registro es un tipo de dato estructurado como agregación heterogénea de elementos de datos identificados por nombre
  - Introducidos por COBOL
  - Utilizados prácticamente en todos los lenguajes (excepto FORTRAN hasta los 90)

Registros

Consideraciones de Diseño

- Cómo se realizan las referencias
- Qué operaciones se definen para la unidad

Registros

- Sintaxis
  - C, C++, C# usan *struct*
  - En C++, *structs* es una variante menor de clase
  - En C# las estructuras son iguales a las clases pero de almacenamiento en pila
    - Las clases se almacenan en el *heap*

Registros

- Sintaxis:
  - COBOL usa números de nivel para mostrar la anidación de los registros.

```
01 EMPLOYEE-RECORD.
  02  EMPLOYEE-NAME.
      05 FIRST PICTURE IS X(20).
      05 MIDDLE      PICTURE IS X(10).
      05 LAST  PICTURE IS X(20).
  02  HOURLY-RATE    PICTURE IS 99V99
```

Registros

- Sintaxis:
  - ADA usa definición recursiva para mostrar la anidación de los registros.

```
Type Employee_Name_Type is record
      FIRST    String (1..20);
      MIDDLE   String (1..10);
      LAST     String (1..20);
end record;
Type Employee_Record_Type is record
      Employee_Name  Employee_Name_Type;
      Hourly_Rate:   Float;
      LAST     String (1..20);
end record;
```

Registros

- Referencia a los campos del registro
  - COBOL
    - field_name OF record_name_1 OF ... OF record_name_n
    - de menor a mayor

      ```
      MIDDLE OF EMPLOYEE-NAME OF EMPLOYEE-RECORD
      ```
  - Otros (notación punto)

    record_name_1.record_name_2. ... record_name_n.field_name

    ```
    Employee_Record.Employee_Name.Middle
    ```

Registros

- Referencias completas, requieren todos los nombres de los registros. (Ej Cobol, Ada)
- Referencias elípticas permiten ignorar referencias no ambiguas for example in COBOL
  - Siempre brindan el nombre del campo
  - Requiere que el compilador “comprenda” las estructuras de datos
  - Pascal incorpora la cláusula ***with***

Registros

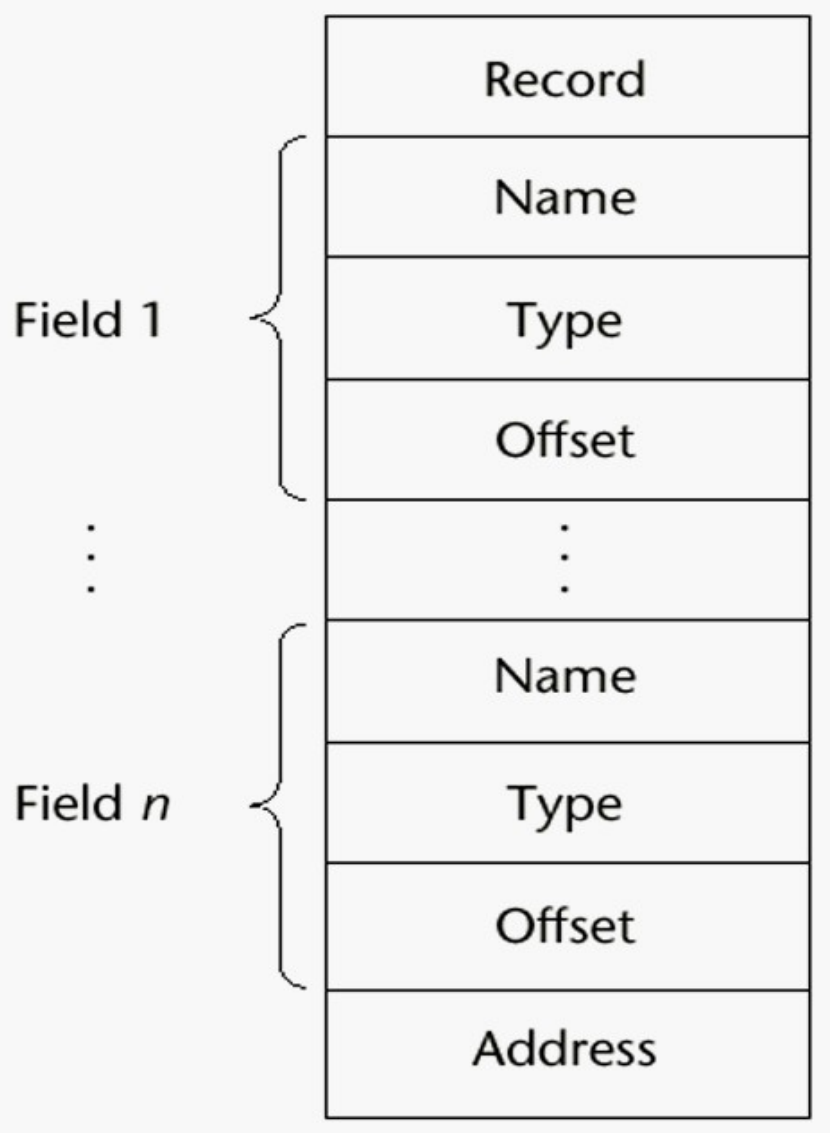
- Pascal, Modula y Algol

```
persona: RECORD
        nombre : string[40];
        edad  : integer;
        domicilio: string[100];
End;
...
   persona.nombre = “Ana Laura”;
   persona.edad = 22;
   with persona do
      write(nombre,‘ tiene’, edad, ‘ años’);
```

- Salida: Ana Laura tiene 22 años

Registros: implementación

- Descriptor en compilación



Registros: operaciones

- Asignación:
  - Pascal, Ada, y C: si los tipos son idénticos
- Comparación
  - Ada: = y /=; un operando puede ser constante
- Movimiento calificado (MOVE CORRESPONDING)
  - COBOL: mueve a todos los campos del registro origen a los campos del registro destino con el mismo nombre

Registros: componentes estructurados

- Pascal

```
persona: RECORD
   nombre: string[40];
   edad: integer;
   domicilio: RECORD
      calle:string[20];
      numero: integer;
      colonia: string[15];
      ciudad: string[15];
   End;
End;
arr_person : ARRAY [1..50] OF persona;
```

Registros: componentes estructurados

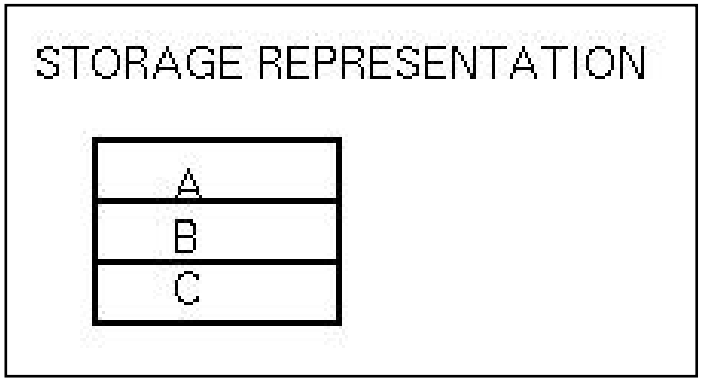
- C

```
struct persona{
   char nombre[40];
   int  edad;
   char domicilio[50];
   float pagos;
} arrPerson[25];
```

Registros: implementación

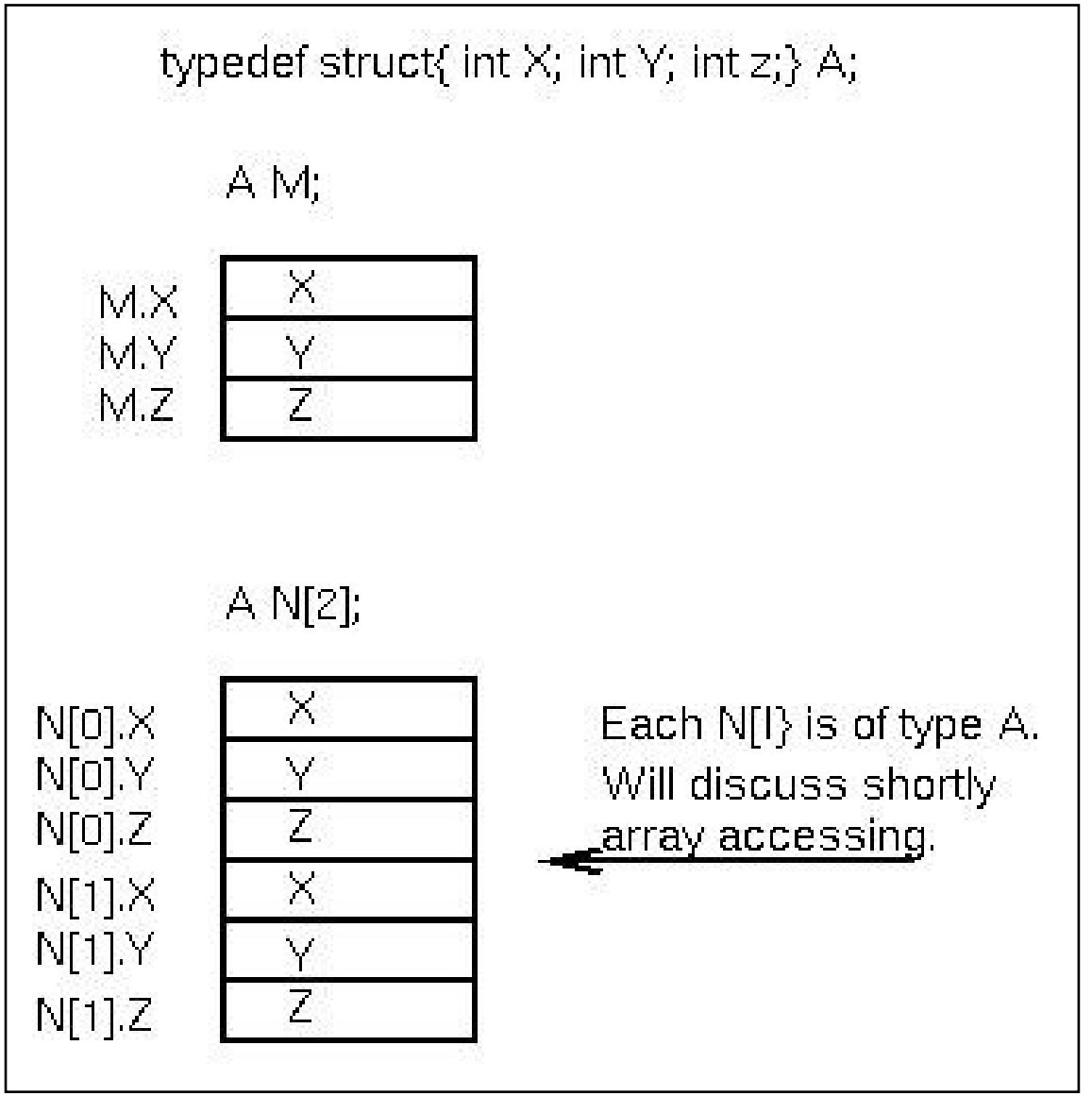
- Representación
  - Secuencial para cada elemento componente
- Ejemplo

```
record { A: object;
         B: object;
         C: object }
```



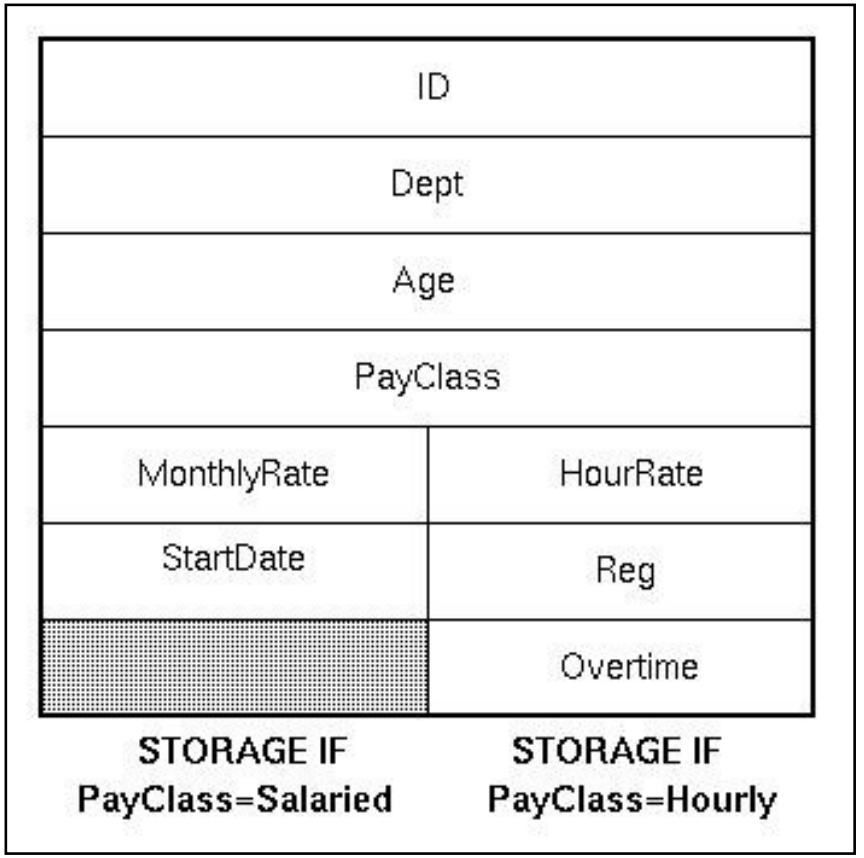
Registros: implementación

- Ejemplo



Registros Variantes: ejemplo

```
type PayType=(Salaried, Hourly);
var Employee:record
    ID: integer;
    Dept: array[1..3] of char;
    Age: integer;
    case PayClass: PayType of
       Salaried:(MonthlyRate:real;
    StartDate:integer);
       Hourly:(HourRate:real;
    Reg:integer;
    Overtime:integer)
    end
```



Bibliografía

- Pratt, Terrance W., Programming Languages: Design and Implementation, Cap. 6 (4 ed)
- Sebesta, Robert, Concepts of Programming Languages, 9th Edition. 2009. Cap. 6
- Material de apoyo en pedco.