palabra reservada, OBJECT, para identificar un tipo de variable estructurado que tiene, como el RECORD, cero o más campos, pudiendo cada uno ser de diferente tipo, pero con la posibilidad de incluir también *procedimientos* y *funciones* denominados **métodos**; por ejemplo, la declaración de una variable tipo objeto será:

```
Empleado = OBJECT
  nombre,
  apellido:       str32
  salarioHora,
  horasPago:      integer
  PROCEDURE       Empleado.Vinc(nombre1,
                  nombre2: str32;
                  s, horasTrab: integer);
  FUNCTION        PagoRegular(horasTrab:
                  integer): integer;
  PROCEDURE       HagaCheque(horasTrab:
                  integer);
END
```

Una variable tipo objeto puede heredar campos y métodos de otro tipo objeto, agregar campos y métodos propios, y derogar ("override") para sí la acción de un método, ej:

```
Ejecutivo = OBJECT (Empleado)
  FUNCTION        PagoRegular (horasTrab:
                  integer): integer;
                  OVERRIDE
  PROCEDURE       Bonific (rendimiento:
                  integer);
END
```

GUIAS PARA LA ADOPCION

La orientación por objetos responde adecuadamente a prioridades actuales de la ingeniería de software, como:

- Reutilización de código.
- Simplificación del diseño.
- Interfaces bien definidas.
- Compatibilidad entre códigos.
- Portabilidad.
- Autonomía de los datos.
- Extensibilidad de los programas.

Se reconoce que algunos lenguajes como SIMULA, Smalltalk y Eiffel llevan el concepto de O.O. de una manera innata, pura y consistente, mientras que otros lenguajes tradicionales han sido *adaptados* o *extendidos* a este enfoque objetual.

La tecnología de objetos aplicada al ciclo completo de la ingeniería de software contribuye a resolver las preocupaciones típicas:

- Los proyectos no están cumpliendo con las expectativas.
- Los usuarios o clientes no participan lo suficiente en el desarrollo de los sistemas.
- La programación se codifica *desde cero* una y otra vez.
- Cada vez es más difícil hacer mantenimiento al software existente.
- El incremento de la productividad es imperativo ante la creciente competencia.

Al comenzar a adoptar la tecnología de objetos, es importante hacerlo con determinación pero al mismo tiempo con cautela; o sea, decididos a hacerlo, pero sin arriesgarnos indebidamente. He aquí unos puntos importantes a considerar:

- Inicialmente se debe escoger un proyecto que no esté en la ruta crítica de la compañía, mientras se adquiere la experiencia, para abordar esos procesos críticos cuando esté afianzada la "cultura" de los objetos entre analistas y programadores.
- El primer proyecto debe comprometer a un grupo de personas pequeño pero bien calificado y, sobre todo, con buena disposición ante el proyecto y ante el aprendizaje requerido.
- El grupo de trabajo se debe dedicar con exclusividad al proyecto.
- El grupo debe comenzar por capacitarse en los conceptos del análisis y diseño O.O. y, si es posible, tener un asesor con experiencia. La capacitación debe ser permanente.