Veremos cómo crear una clase global para poder utilizar la misma en cualquiera de nuestros programas, para ello utilizaremos la transacción SE24 en OBJECT TYPE introduciremos el nombre de la nueva clase que será Z\_CL\_VEHICULO y presionamos el botón CREAR. Luego seleccionamos la opción CLASS y continuamos OK.

En la siguiente ventana de diálogo ingresamos la descripción de la clase [CLASE GLOBAL], si presionamos el botón a la derecha de la clase, podemos definir una relación de herencia especificando el nombre de la superclase, en INSTANTIATION [PUBLIC] definimos la visibilidad de la clase que puede ser pública, privada o protegida, por último guardamos OK

Vamos a **definir los atributos** de la clase EN LA SOLAPA ATRIBUTES:

Definimos el atributo SPEED que será un atributo de instancia [INSTANCE ATRIBUTE].

Definimos la visibilidad del atributo como publica, el tipo de atributos será I ,e introducimos una descripción para el mismo [VELOCIDAD]

También creamos los **atributos:**

**[MAX\_SPEED]** [INSTANCE ATRIBUTE] [PUBLIC] [I] [VELOCIDAD MAXIMA]

**[STOP]** [INSTANCE ATRIBUTE] [PUBLIC] [I] [STOP]

**[WRITE]** [INSTANCE ATRIBUTE] [PUBLIC] [I] [MOSTRAR LISTADO]

Ahora **definimos los métodos**, en la solapa METHOD:

Creamos el método INCREASE\_SPEED el cual definimos como método de instancia [INSTANCE] con visibilidad pública [PUBLIC] con la descripción [AUMENTO DE VELOCIDAD].

También creamos el método CALCULATE\_STOP el cual definimos como método de instancia [INSTANCE] con visibilidad pública [PUBLIC] con la descripción, cuyo propósito sería [RESETEAR LA VELOCIDAD].

Y el método DISPLAY\_WRITE el cual definimos como método de instancia [INSTANCE] con visibilidad pública [PUBLIC] con la descripción, que [MOSTRAR EL LISTADO DE VELOCIDADES].

Para este método **definimos parámetro**s, EN LA SOLAPA PARAMETERS:

Creamos el parámetro P\_PARAMETRO Type [IMPORTIN] de entrada del tipo P con la descripción [PARAMETRO]

También creamos el parametro P\_RESULTADO Type [IMPORTIN] de entrada del tipo P con la descripción [RESULTADO]

Ahora creamos una **excepción** para el método DISPLAY\_WRITE, en el botón Exception:

Exception [VEHICULO\_NO\_EXISTE] Description [EL VEHICULO NO EXISTE]

También creamos el **evento,** en la solapa EVENTS,

Event [CAMBIO\_ESTADO\_VEHICULO] como de Type [INSTANCE], con visibilidad [PUBLICA] y una descripción acorde de su propósito [CAMBIO DE ESTADO DE UN VEHICULO].

Por último podemos crear los **tipos**, en la solapa TYPES, para utilizar dentro de nuestra clase global.

Type [MARCA\_VEHICULO] visibilidad [PUBLIC] Typing [TYPE] Associated [CHAR30] Description [MARCA DEL VEHICULO]

Finalmente grabamos, y activamos la clase.

Tal como mencionamos anteriormente para las interfaces, cuando se define una clase en forma global a través del repositorio correspondiente a la transacción SE24. Se genera un programa ABAP denominado Pool de clases el cual es del tipo K. El pool de clases tiene una parte de definiciones y otra parte de implementaciones de la clase, este programa tiene mucha diferencias con respecto a un programa ABAP común y silvestre las principales diferencias son las siguientes:

Los programas ABAP tales como los reportes, module pool, o los módulos de funciones, usualmente tienen una parte de declaración en donde se definen los datos globales, estos son visibles en todos los bloques de procesamiento de programa. Los pool de clases por otro lado tienen una parte de definición en donde se pueden definir datos y tipos de objetos, pero no se pueden definir objetos de datos ni field symbols.

Los tipos que se definen en el pool de clases son solo visibles en la parte de implementación de la clase global. Por otro lado los únicos bloques de procesamiento que se pueden utilizar en un pool de clases son la parte de declaración e implementación de una clase global. La parte de implementación solo podría implementar los métodos declarados en la clase global. No se puede utilizar cualquiera de los otros bloques de procesamiento ABAP tales como los módulos de diálogo, los bloques de eventos, subrutinas y módulos de función.

Los bloques de procesamiento del pool de clases no son controlados por el entorno de ejecución ABAP, no ocurren eventos y no se puede llamar a cualquiera de los eventos o procedimientos de diálogo los pool de clases sirven exclusivamente para la programación de las clases. Solo se puede acceder a los datos y funciones de una clase utilizando su interface.

Por otro lado debido a que los eventos y módulos de diálogo no están permitidos en las clases entonces no se puede procesar las pantallas en las clases, no se pueden crear listados y pantallas de selección en las clases debido a que a estas no pueden reaccionar a los eventos adecuado si el objetivo es hacer que las pantallas están disponibles en las clases, en lugar de módulos de diálogo podremos llamar a los métodos de la clase de la lógica de proceso de la pantalla