|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOMBRE DE LA PRÁCTICA** | **Prácticas de C ++** | | | **No.** | | **1** | |
| **ASIGNATURA:** | **Métodos númericos** | **CARRERA:** | **INGENIERIA EN SISTEMAS COMPUTACIONALES** | | **PLAN:** | | **ISIC-2010-224** |

**I. COMPETENCIA(S) ESPECÍFICA(S): Aprender c++**

**II. MATERIAL EMPLEADO:**

* Laptop con Linix Lite
* Editor de texto Visual Studio Code
* Terminal

**III.**

**DESARROLLO DE LA PRÁCTICA:**

**Ejercicio 1:**

**Modifica la estructura para que se pueda almacenar también.**

**num\_canciones**

**anio**

**precio**

**struct CD{**

**char titulo[100];**

**char artista[50]**

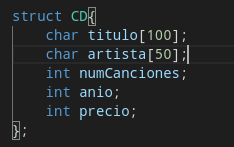
**};**

**Nota: El usuario debe ingresar todos los datos por el teclado**

* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



* Después la estructura llamada CD que tendrá como elementos titulo, artista de tipo cadena de caracteres y numCanciones, anio y precio como tipo entero



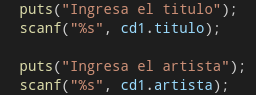
* Declarar el método principal de tipo entero



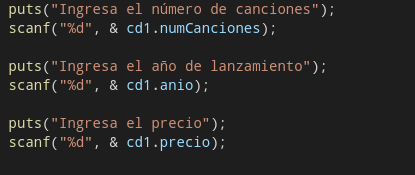
* Hacemos la instancia de la estructura CD llamándola cd1



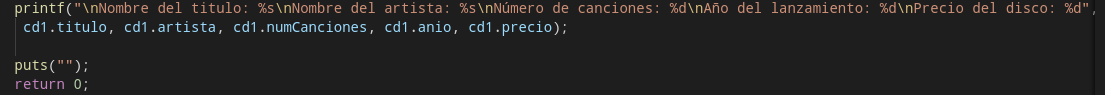
* Después pedir al usuario por teclado las cadenas y almacenarlas donde corresponden



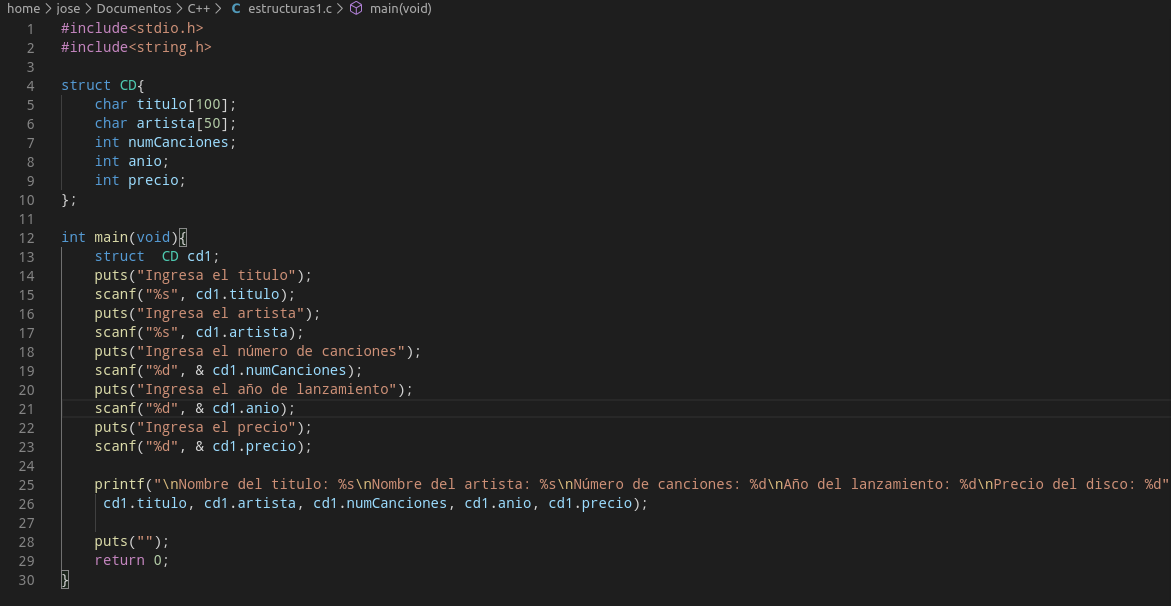
* Lo mismo ocurrirá con los de tipo entero, se pedirán por teclado y se almacenaran donde corresponden



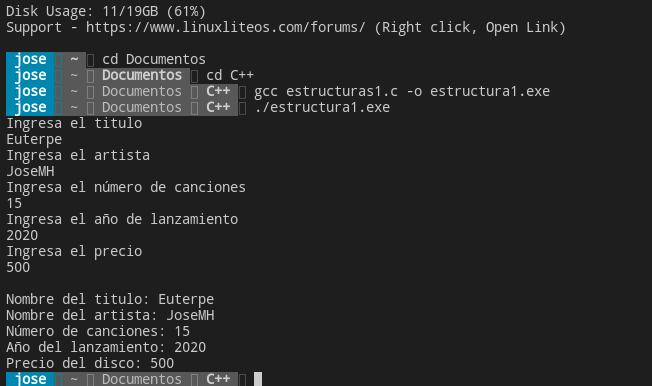
* Después solo sera imprimir los datos recibidos en la pantalla, dando salto de linea y retornar 0



Código completo:



* Una vez compilado nuestro archivo llamado estructuras1.c haremos el .exe para ejecutarlo y este es el resultado



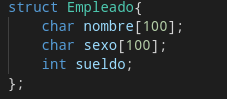
**Ejercicio 2**

**Realiza un programa en C que almacene y muestre la información de un empleado de la empresa DATACIC en una estructura llamada “empleado”. La información con que se cuenta del empleado es: nombre, sexo y sueldo.**

* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



* Después será declarar la estructura llamada Empleado que tiene como elementos el nombre y sexo de tipo cadena de caracteres y el sueldo como entero



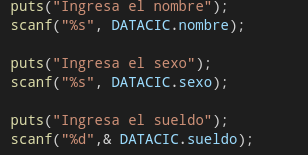
* Luego declarar nuestro método principal de tipo entero



* En seguida llamar a nuestra estructura llamado Empleado y la instancia llamarla DATACIC como el nombre de la empresa



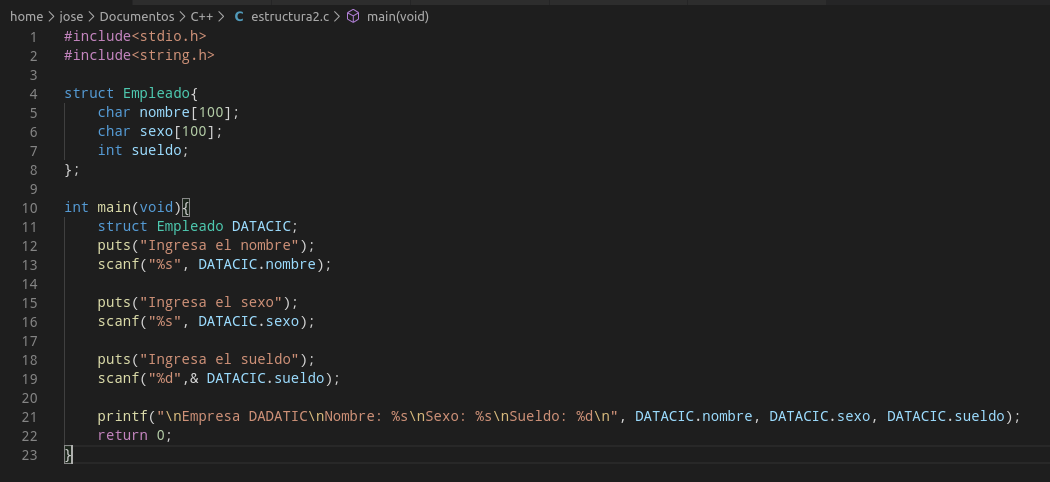
* Pedir datos por teclado y almacenarlos en el lugar correspondiente



* Imprimir los datos ingresados en un mensaje final al usuario



* Código completo



* Una vez compilado nuestro archivo llamado estructura2.c haremos el .exe para ejecutarlo y este es el resultado



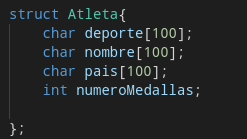
**Ejercicio 3**

**Realiza una programa en C que almacene y muestre la información de un atleta (deporte, nombre, pais, n\_medallas).**

* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



* Después declarar la estructura llamada Atleta con elementos deporte, nombre y país de tipo cadena y numeroMedallas como entero



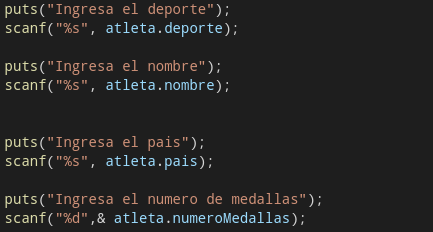
* Declarar el método principal de tipo entero



* Luego llamar la estructura Atleta e instanciarlo como atleta



* Pedir datos al usuario por teclado y guardarlos donde corresponde



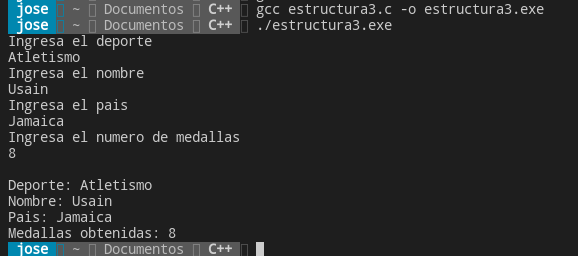
* Luego mandar un mensaje final al usuario donde se recopilaran los datos ingresados y dar salto de linea



* Código completo



* Una vez compilado nuestro archivo llamado estructura3.c haremos el .exe para ejecutarlo y este es el resultado



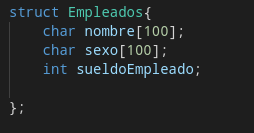
**Ejercicio 4**

**Realiza un programa en C que almacene y muestre la información de 5 empleado de la empresa DATACIC en una estructura llamada “empleado”. La información con que se cuenta del empleado es: nombre, sexo y sueldo. Muestra el empleado de menos sueldo.**

* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



* Declarar la estructura Empleados que tiene como elementos nombre y sexo como cadenas y sueldoEmpleado como entero



* Declarar el método principal



* Luego llamara la estructura Empleados y llamar la instancia como DATACIC que es el nombre de la empresa, aquí tiene algo especial ya que tendrá 5 espacios en la instancia, funcionara casi como una matriz



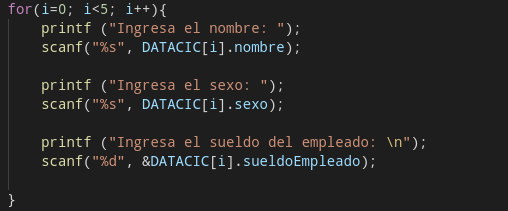
* Las siguientes cadenas solo tendrán como función un respaldo de la cadena con su mismo nombre pero sin “R”



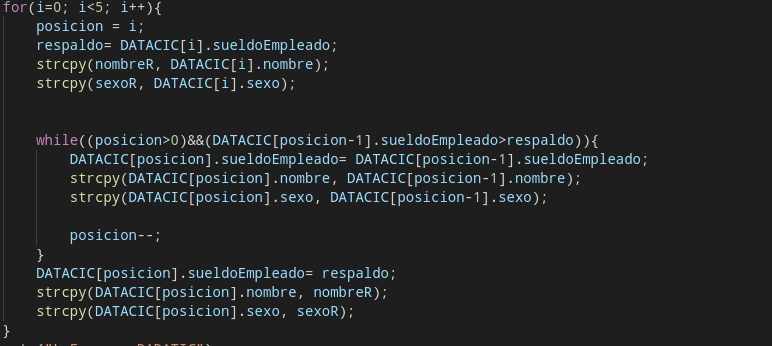
* Después declarar variables de tipo entero, la variable i será utilizada para contador y posición para movernos entre posiciones y el respaldo que será donde se almacenara el sueldo del empleado



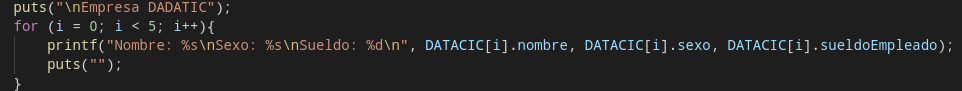
* Meter en un ciclo for los mensajes para que el usuario ingrese los empleados, lo que cambia es en que rubro se guardara, ya que tiene la función parecida como arreglo se hira cambiando para llenar toda la matriz



* A qui viene la parte más importante del programa ya que debido a que son 5 empleados para tener quien es el que gana menos seria demasiadas condiciones así que opte utilizar el método de ordenamiento **inserción** que consiste si el número menor es mayor al actual cambiara de posición. Este método lo decidí ya que es mejor que el burbuja y en sintaxis es fácil de digerir así que lo primero que se hace es respaldar la posición actual ya que cambiara y si no se hace esto la información actual se perderá, después va un ciclo while que tendrá como condición si la posición es mayor que 0 ya que si no generara un error llamando una posición negativa y lo dicho de el método **inserción** que si la posición anterior es mayor a la actual entonces se cambiara la actual con la anterior y la posición se decremento en uno. Después solo cambar la posición actual con la del principio.



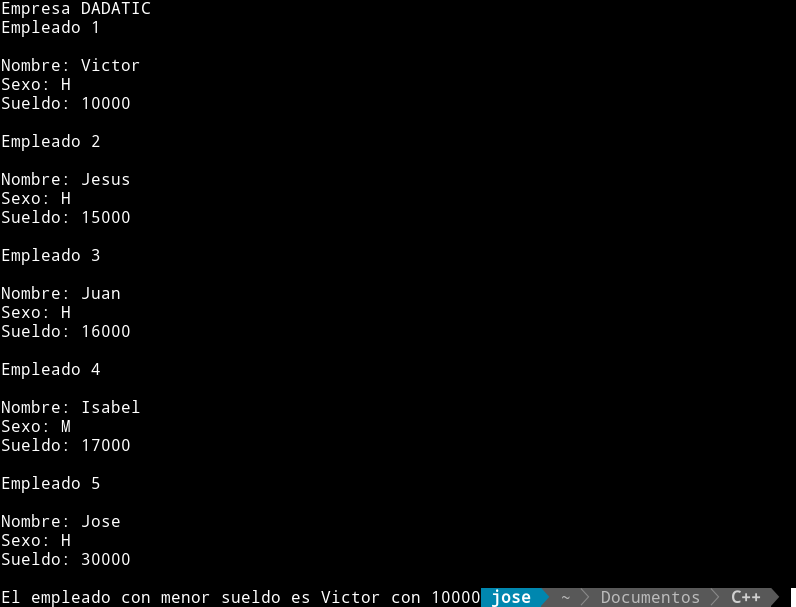
* Luego imprimir los 5 empleados, pero ahora no será como se ingresaron si no que primero aparecerá el que gana menos debido al método de ordenación



* Un mensaje que dirá quien es el empleado que gana menos con su nombre



* Una vez compilado nuestro archivo llamado estructura4.c haremos el .exe para ejecutarlo y este es el resultado



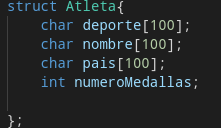
**Ejercicio 5**

**Realiza una programa en C que almacene y muestre la información de 10 atleta (deporte, nombre, pais, nmedallas) y te diga que atleta tiene mayor número de medallas.**

* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



* Declarar la estructura Atleta que tiene como elementos nombre, país y deporte cadenas y numeroMedallas como entero



* Declarar el método principal



* Luego llamara la estructura Atleta y llamar la instancia como atleta, aquí tiene algo especial ya que tendrá 10 espacios en la instancia, funcionara casi como una matriz



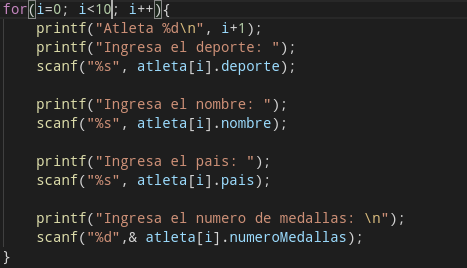
* Las siguientes cadenas solo tendrán como función un respaldo de la cadena con su mismo nombre pero sin “R”



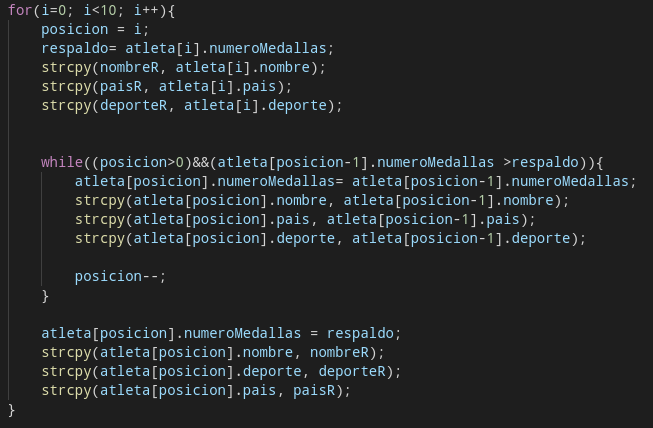
* Después declarar variables de tipo entero, la variable i será utilizada para contador y posición para movernos entre posiciones y el respaldo que será donde se almacenara el sueldo del empleado



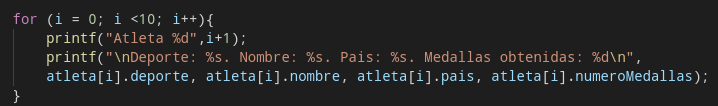
* Meter en un ciclo for los mensajes para que el usuario ingrese los atletas, lo que cambia es en que rubro se guardara, ya que tiene la función parecida como arreglo se hira cambiando para llenar toda la matriz



* A qui viene la parte más importante del programa ya que debido a que son 10 atletas para tener quien es el que gana más medallas seria demasiadas condiciones así que opte utilizar el método de ordenamiento **inserción** que consiste si el número menor anterior es mayor al actual cambiara de posición. Este método lo decidí ya que es mejor que el burbuja y en sintaxis es fácil de digerir así que lo primero que se hace es respaldar la posición actual ya que cambiara y si no se hace esto la información actual se perderá, después va un ciclo while que tendrá como condición si la posición es mayor que 0 ya que si no generara un error llamando una posición negativa y lo dicho de el método **inserción** que si la posición anterior es mayor a la actual entonces se cambiara la actual con la anterior y la posición se decremento en uno. Después solo cambar la posición actual con la del principio.



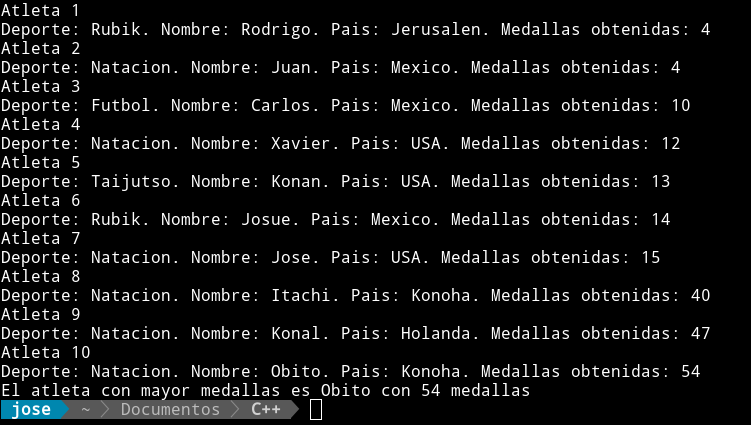
* Luego imprimir los 10 atletas, pero ahora no será como se ingresaron si no que primero aparecerá el que gana menos medallas debido al método de ordenación



* Un mensaje que dirá quien es el atleta que gana más medallas con su nombre



* Una vez compilado nuestro archivo llamado estructura5.c haremos el .exe para ejecutarlo y este es el resultado

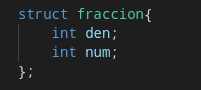


**Ejemplo 2**

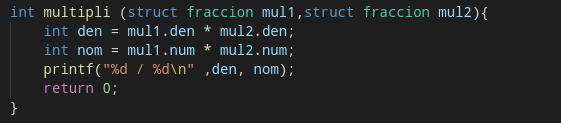
* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



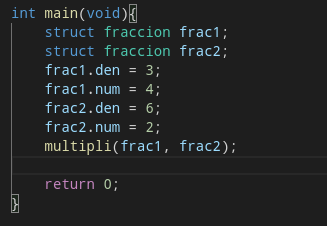
* Después declararemos la estructura fracción que tiene como elementos dos variables de tipo entero llamadas den y num



* Declarar la función que será de tipo entera que recibirá por parámetros elementos de estructuras y se multiplicaran los dos denominadores y denominadores imprimiendo el valor de ambas multiplicaciones



* En el método principal solo se instancia la estructura en 2 teniendo valores distintos y se llamara la función multipli

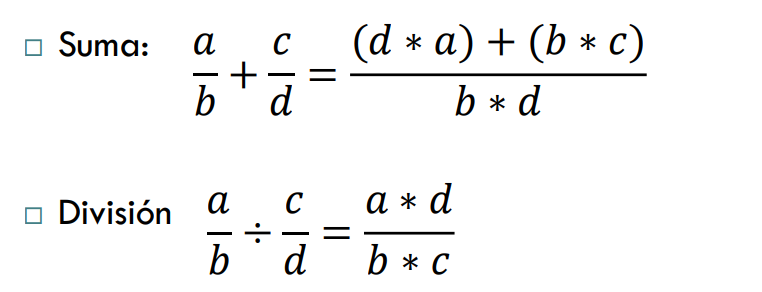


* Una vez compilado nuestro archivo llamado estructura6.c haremos el .exe para ejecutarlo y este es el resultado



**Ejercicio 7**

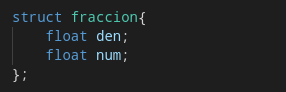
**Implementar funciones para la suma, la resta y la división de fracciones usando la estructura fracción.**

****

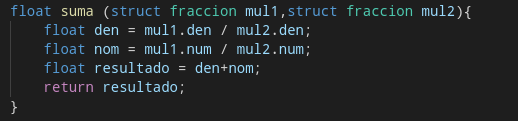
* Primero importamos las bibliotecas necesarias que son la del estándar de entrada y salida, y la necesaria para trabajar con cadenas



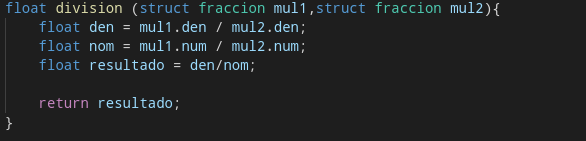
* Después declararemos la estructura fracción que tiene como elementos dos variables de tipo flotante llamadas den y num



* Luego como el ejercicio nos pide crear una función llamada suma que hará una serie de operaciones y retornara el valor, aquí inicia la función suma que recibirá por parámetros elementos de tipo estructura, después dividirá ambos elementos y unirá las dos operaciones en la variable de tipo flotante con una suma



* La otra función se llama división que recibirá dos parámetros de tipo estructura, se dividen ambos valores y se unen en otra variable llamada resultado que será la división de ambas operaciones anteriores



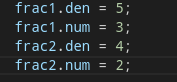
* Declarar el método principal



* Después de la estructura fracción hacer dos instancias con nombre frac1 y frac2



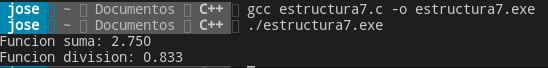
* Luego darles valores a estas instancias



* Imprimir el resultado de ambas funciones



* Una vez compilado nuestro archivo llamado estructura7.c haremos el .exe para ejecutarlo y este es el resultado



**IV Conclusión:**

Una vez resueltos estos problemas planteados llego a la conclusión que esto de las estructuras esta demasiado útil ya que podemos instanciarlo y por así decirlo es como el molde para que puedan hacerse mucho más, yo lo veo como el chasis de un carro que es el molde de todos los carros pero cada marca lo implementa como quiere y le da polimorfismo. Esto resulta muy útil en bastantes casos.