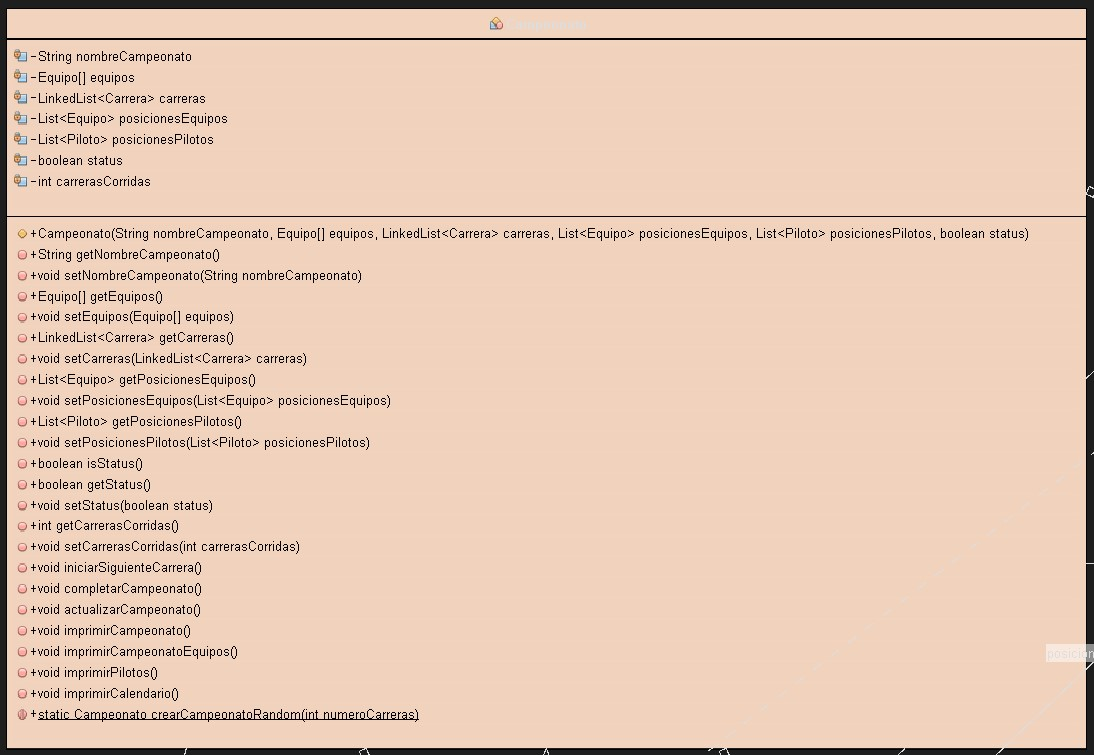
El diagrama UML del proyecto es el siguiente:

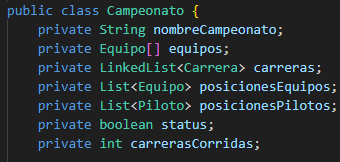


Las clases ocupadas en el proyecto son: Campeonato, Equipo, Piloto, Carrera, Pista, Fecha, Utilerías y Menú.

**Campeonato**



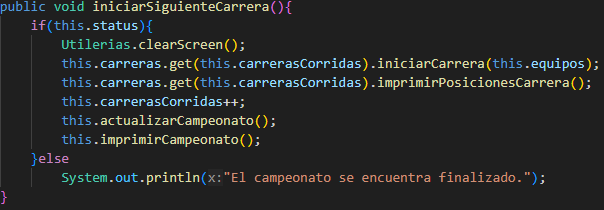
Campeonato es el objeto principal del programa, sus atributos son los siguientes:



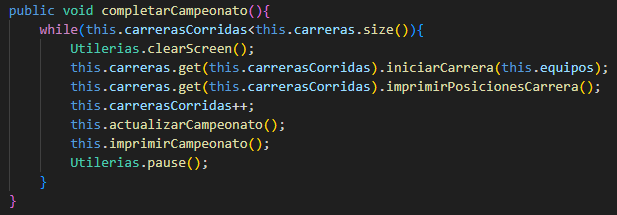
Los equipos se guardan en un arreglo, las carreras en una lista ligada, se usan dos listas para almacenar las posiciones de los equipos y de los pilotos. El atributo *status* se usa para identificar si el campeonato esta en progreso o si ya fue finalizado. La variable **carrerasCorridas** se inicializa siempre en 0 al crear el campeonato.

Además del constructor y los métodos de accesos *getters* y *setters* la clase contiene los siguientes métodos:

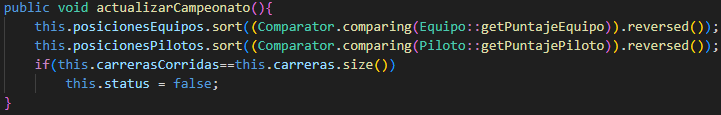
**iniciarSiguienteCarrera:** Se encarga de iniciar la siguiente carrera del torneo, esto lo hace accediendo a la lista de carreras en la posición de la variable “**carrerasCorridas**”, que se inicializa en 0 al crear un Campeonato y se va incrementado con cada carrera finalizada, y aplicando el método “**iniciarCarrera**”, una vez terminada: imprime las posiciones de dicha carrera, llama al método **actualizarCampeonato** e imprime el estado actual del campeonato. Si el campeonato se encuentra finalizado el método simplemente imprimirá un mensaje informándolo.



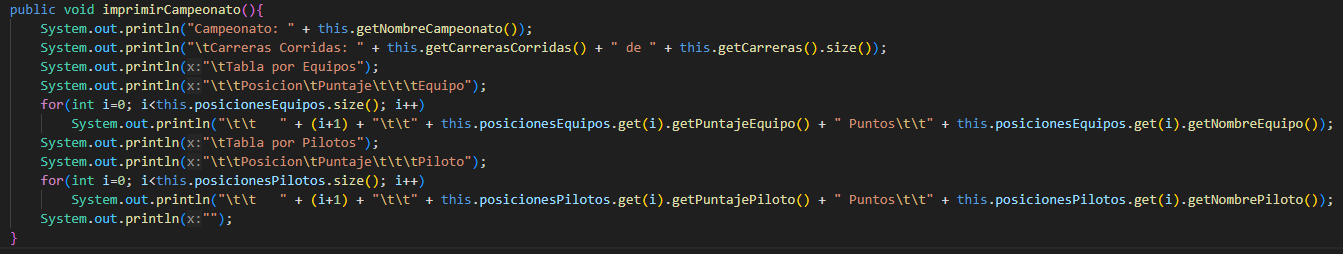
**completarCampeonato:** Similar al método anterior, entra en un ciclo *while* siempre que el numero de carreras corridas sea menor al numero de carreras especificadas en el campeonato. En cada ciclo va iniciando una carrera, mostrando los resultados al usuario y actualizando el estado del campeonato.



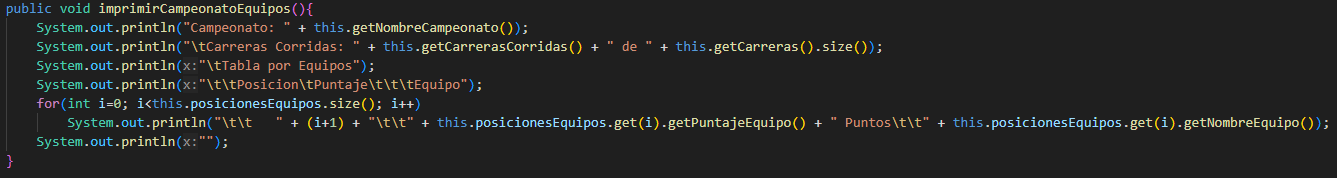
**actualizarCampeonato:** Este método reordena las listas **posicionesEquipos** y **posicionesPilotos**, de manera inversa al orden natural, finalmente revisa si el número de carreras corridas es igual al numero de carreras totales del campeonato, de ser así, cambia el estado del campeonato a finalizado.



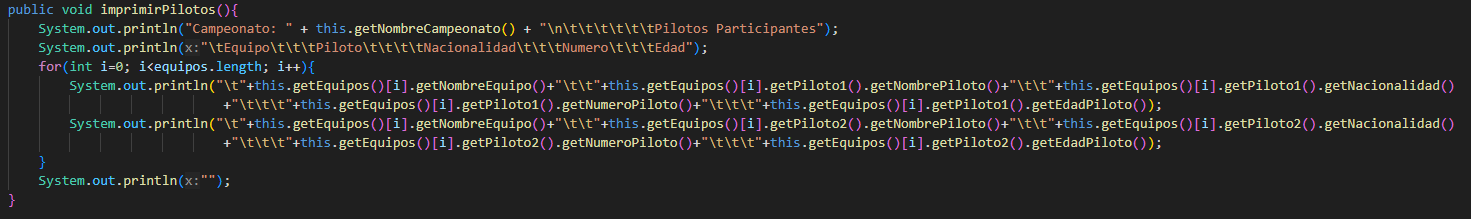
**imprimirCampeonato:** Imprime en pantalla la información completa del torneo: nombre del campeonato, numero de carreras corridas del total, la tabla de posiciones por equipos y por pilotos.



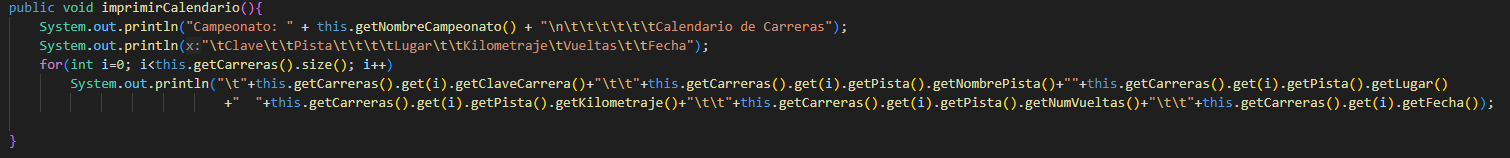
**imprimirCampeonatosEquipos:** Imprime únicamente el nombre del campeonato, las carreras corridas y la tabla de posiciones por equipos.



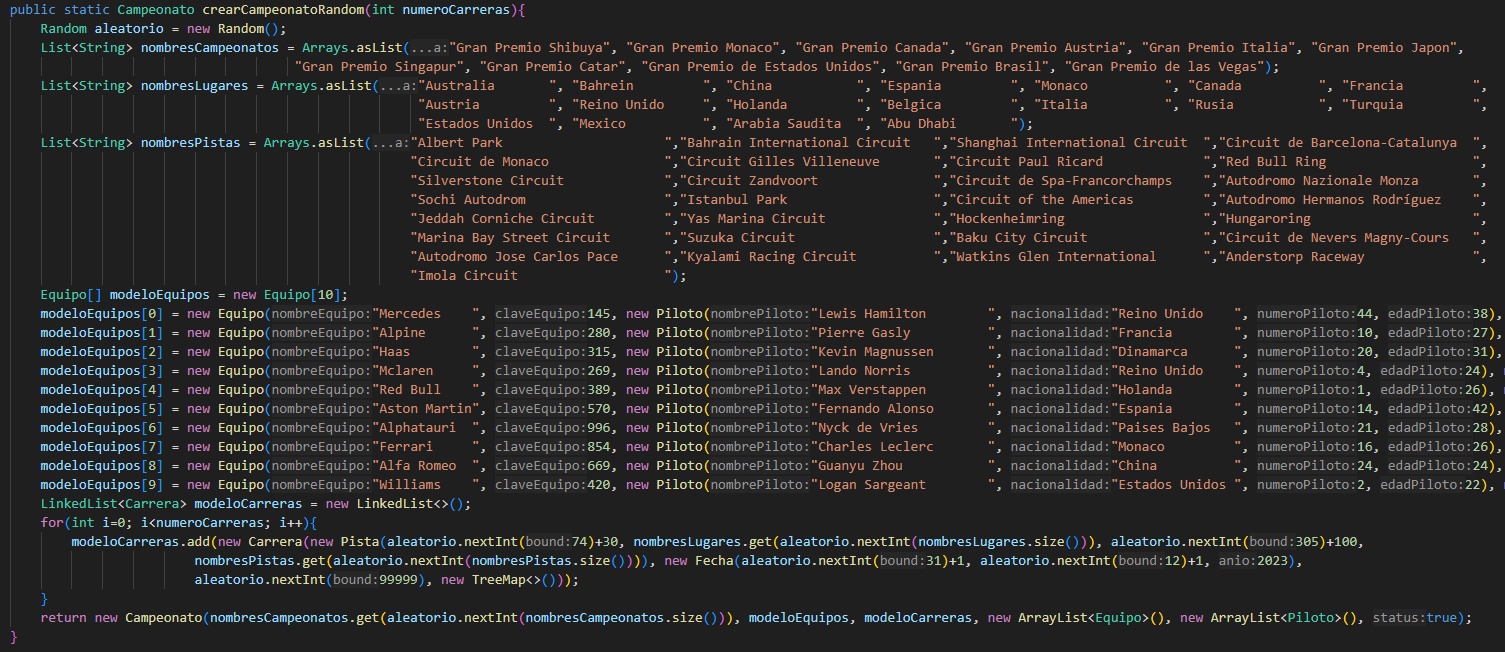
**imprimirPilotos:** Imprime en pantalla los pilotos que participan en el torneo y su información.



**imprimirCalendario:** Imprime la información sobre las carreras del campeonato, detalles sobre las pistas y la fecha en la que se llevaran a cabo las carreras.

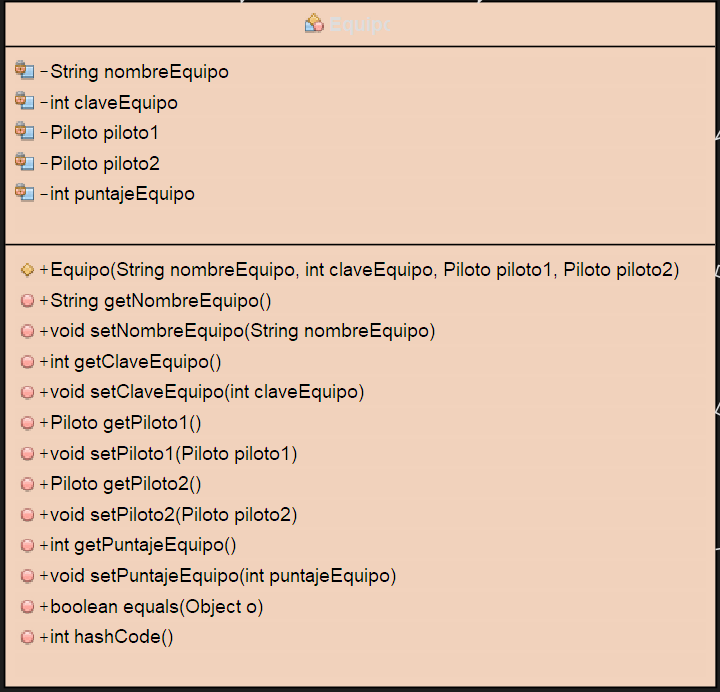


**crearCampeonatoRandom:** Durante las pruebas y la ejecución final del programa, nos dimos cuenta de lo tedioso que es llenar toda la información necesaria para un campeonato, pero solo tener un campeonato previamente cargado en código en el método *main* no nos parecía suficiente, por lo que decidimos crear un método que genere un campeonato de manera semi aleatoria. Es semi aleatoria porque los equipos y pilotos siempre se crean de la misma forma. Para el resto de la información se crearon listas con varios elementos posibles para dichos atributos, y por medio de un *random* se elige uno de manera aleatoria, de esta forma el campeonato se genera con una combinación de los valores posibles.

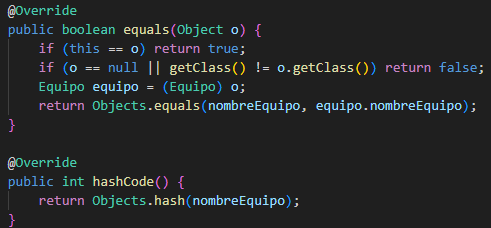


**Equipo**

Cada equipo cuenta con un nombre, clave, dos pilotos y un puntaje.

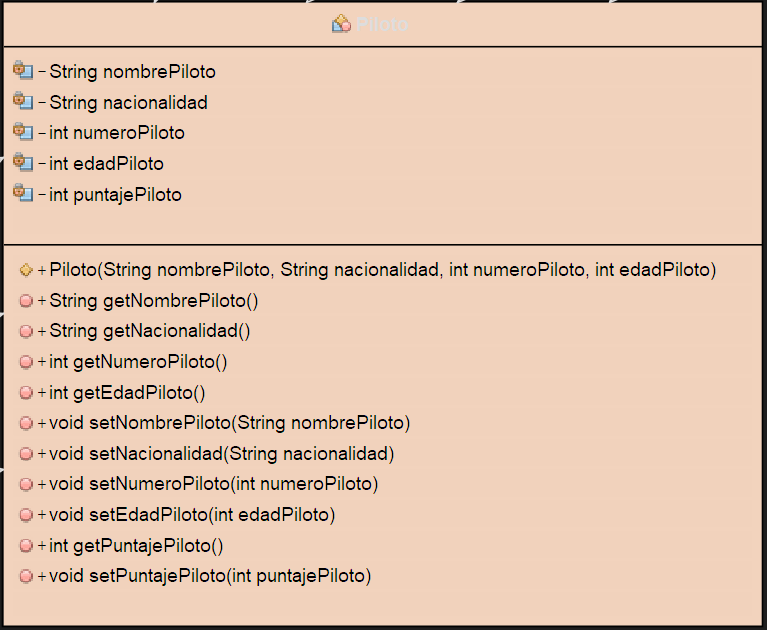


Para esta clase además del constructor y métodos de acceso, solo implementamos dos métodos más: *equals* y *hashCode*. Estos métodos sobrescriben a los que están por defecto, esto lo hicimos para que solo se considerara el nombre del equipo en estos métodos, para evitar que se pudieran crear dos equipos con el mismo nombre.



**Piloto**

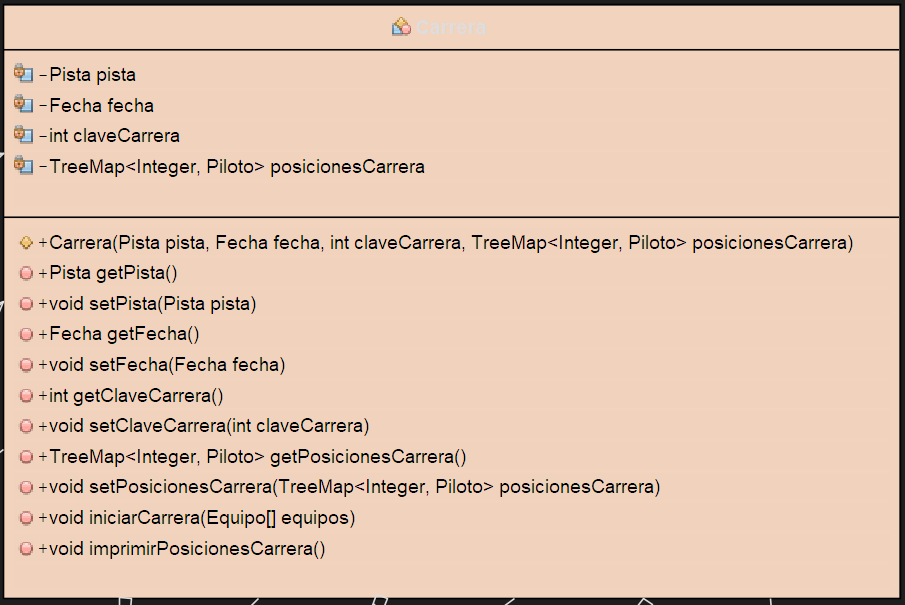
Cada piloto tiene nombre, nacionalidad, numero, edad y un puntaje.



Esta clase no tiene métodos además del constructor y los de acceso.

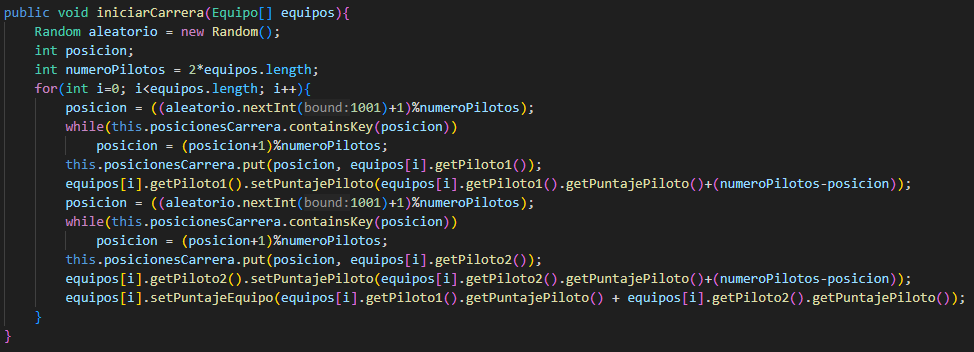
**Carrera**

Cada carrera tiene una pista, fecha, clave y un *TreeMap* que tiene como clave un entero que representa la posición en la carrera y un Piloto como valor.

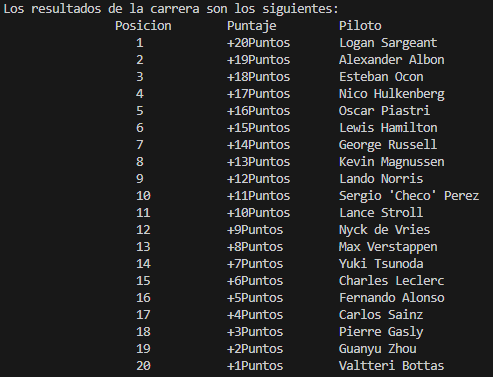


Además del constructor y los métodos de acceso esta clase cuenta con los siguientes métodos:

**iniciarCarrera:** Este método recibe el arreglo de equipos y lo utiliza para obtener la cantidad de pilotos que hay en la carrera. Después entra en un ciclo *for* en el cual asigna una posición aleatoria a cada piloto (desde 0 hasta el numero de pilotos menos uno), esta será la posición en la que termino dicho piloto en la carrera. Si la posición obtenida ya esta ocupada le suma uno y vuelve a comprobar, repitiendo esto con un *while* hasta encontrar una posición libre. Con dicha posición agrega al piloto al *TreeMap*, finalmente accede al puntaje de dicho piloto y le agrega un valor dependiendo del lugar en el que termino la carrera. Dado que cada equipo tiene dos pilotos, este proceso se hace dos veces (una para cada piloto) por ciclo. El valor de **numeroPilotos** nos permite utilizarlo como modulo para que las posiciones siempre estén dentro de las posibles en base al número de participantes en la carrera.

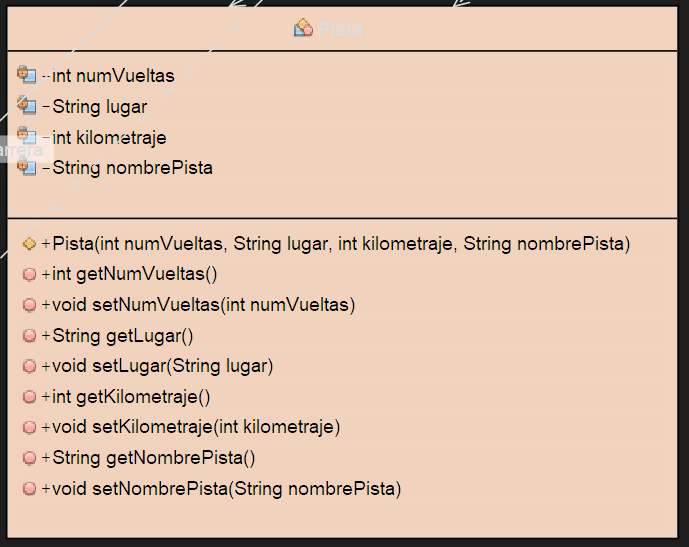


**imprimirPosicionesCarrera:** Imprime una tabla con los resultados de la carrera: posiciones, puntaje y piloto.



**Pista**

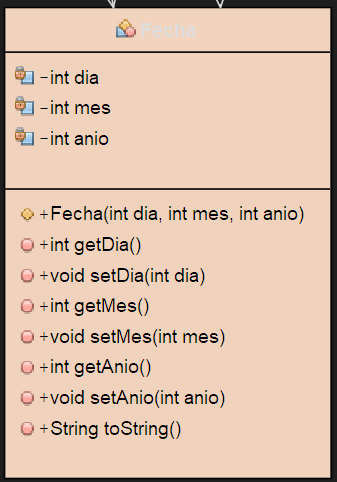
Cada pista cuenta con numero de vueltas, lugar, kilometraje y nombre de la pista.



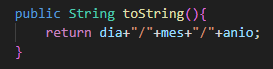
Esta clase no tiene métodos además del constructor y los de acceso.

**Fecha**

Cada fecha tiene día, mes y año.

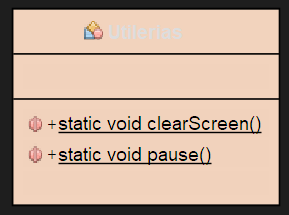
****

Además del constructor y los métodos de acceso, se agrego un método para sobrescribir el toString y poder definir como se mostraría la fecha:

****

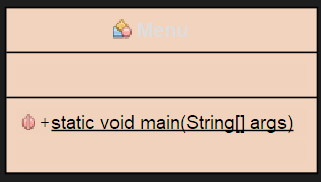
**Utilerías**

Esta clase no tiene atributos, únicamente tiene dos métodos de clase que se utilizan para limpiar la pantalla (*clearScreen*) y para pausar la ejecución del programa (*pause*).



**Menú**

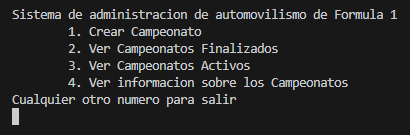
Esta es la clase principal del proyecto, contiene únicamente un método que es el *main*.



Lo primero que se hace es declarar una lista de campeonatos, después se crea un campeonato desde código para que el usuario pueda utilizar las opciones del menú desde el inicio. Luego de crearse se añade a la lista.

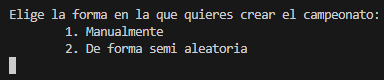
Lo siguiente es entrar al *while* para mostrar el menú, se limpia la pantalla utilizando el método *clearScreen()* y siempre que se termine de ejecutar alguna de las opciones del menú se regresara a esta pantalla.

Las opciones que se presentan al usuario son las siguientes:



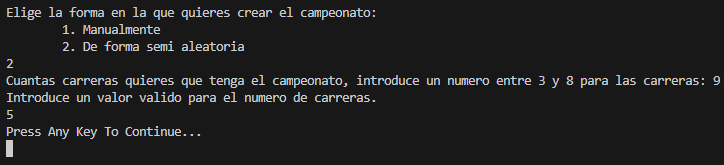
1. Crear Campeonato

Una vez seleccionada esta opción se nos presentan otras dos:



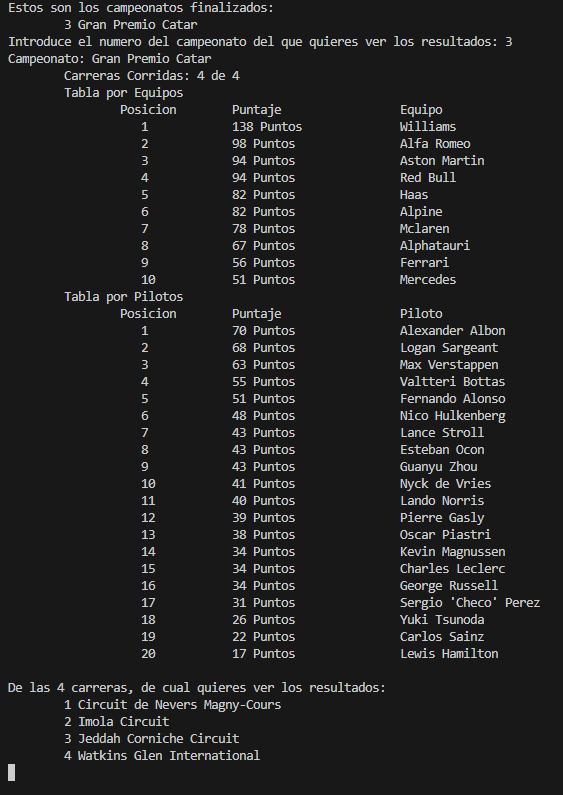
Si elegimos manualmente se nos irán pidiendo uno por uno los datos necesarios para crear el campeonato: número de equipos, nombres de los equipos, nombres de los pilotos, numero de carreras.

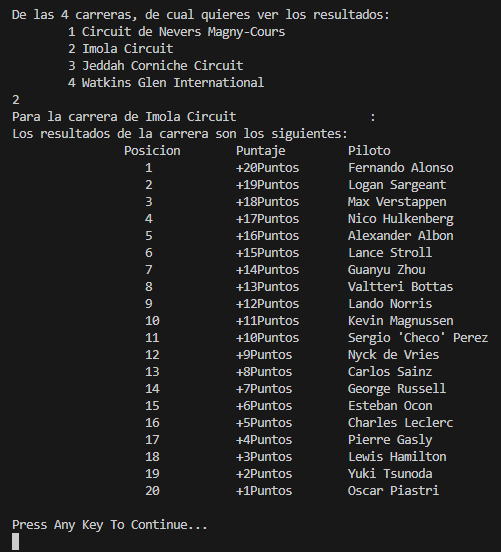
Si elegimos la opción de “De forma semi aleatoria”, únicamente se nos pedirá el numero de carreras que tendrá el campeonato, el cual debe ser de 3 a 8, elegir un numero fuera de ese rango nos indicara que es invalido y nos pedirá nuevamente un valor.



1. Ver Campeonatos Finalizados

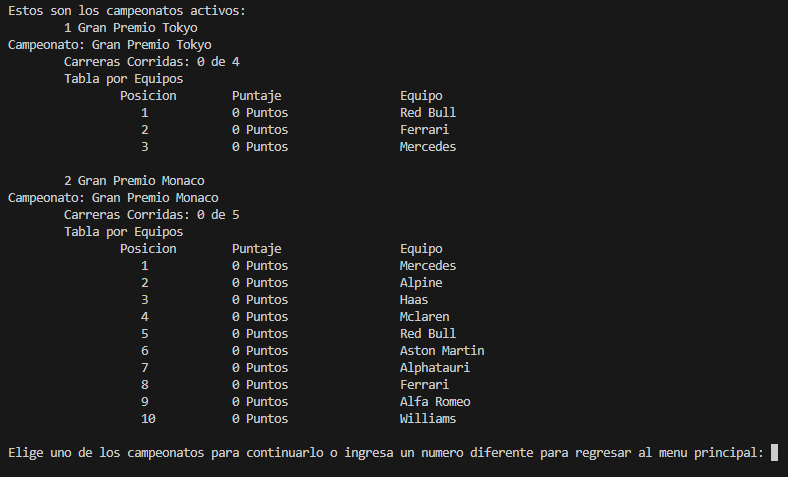
Esta opción muestra los campeonatos que ya han sido completados, al elegir alguno se muestra la información de como finalizo el campeonato y se puede elegir alguna de las carreras para ver los resultados específicos de la misma.



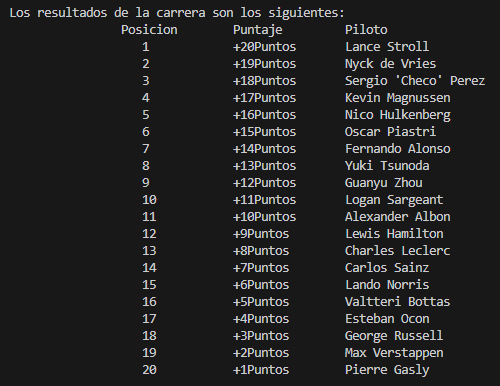


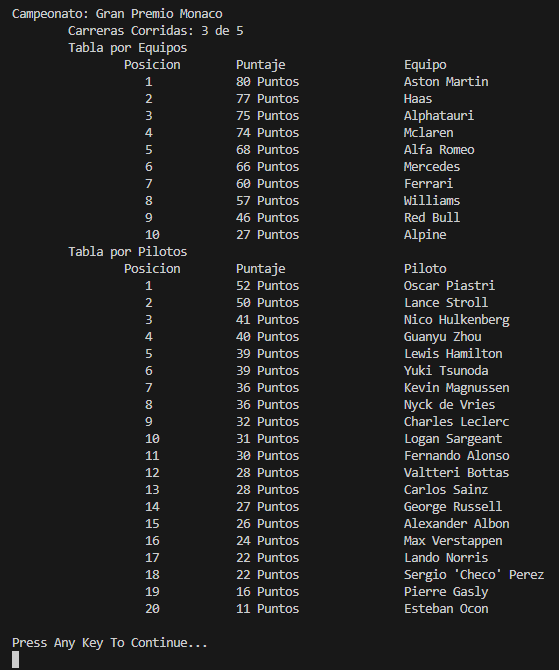
1. Ver Campeonatos Activos

Esta opción muestra los campeonatos que aun están en progreso, muestra la tabla de posiciones por equipos actualizada hasta ese punto y permite elegir alguno de los campeonatos para continuarlo.



Al elegir uno de los campeonatos se nos presentaran las opciones de “Iniciar una carrera” o “Completar el campeonato”. Iniciar una carrera, como su nombre lo indica únicamente ejecutara una de las carreras del campeonato, mostrara los resultados de dicha carrera y la tabla actualizada del campeonato.

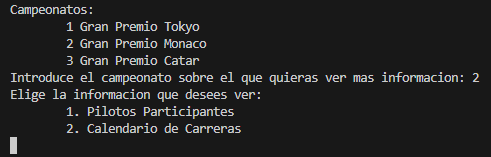




“Completar el campeonato” ira ejecutando una por una todas las carreras que hagan falta, mostrando el resultado de dicha carrera en cada ejecución.

1. Ver información sobre los Campeonatos

Este método nos pide seleccionar uno de los campeonatos.



Después se mostrarán las opciones para desplegar la información de los pilotos participantes o del calendario de carreras.



