

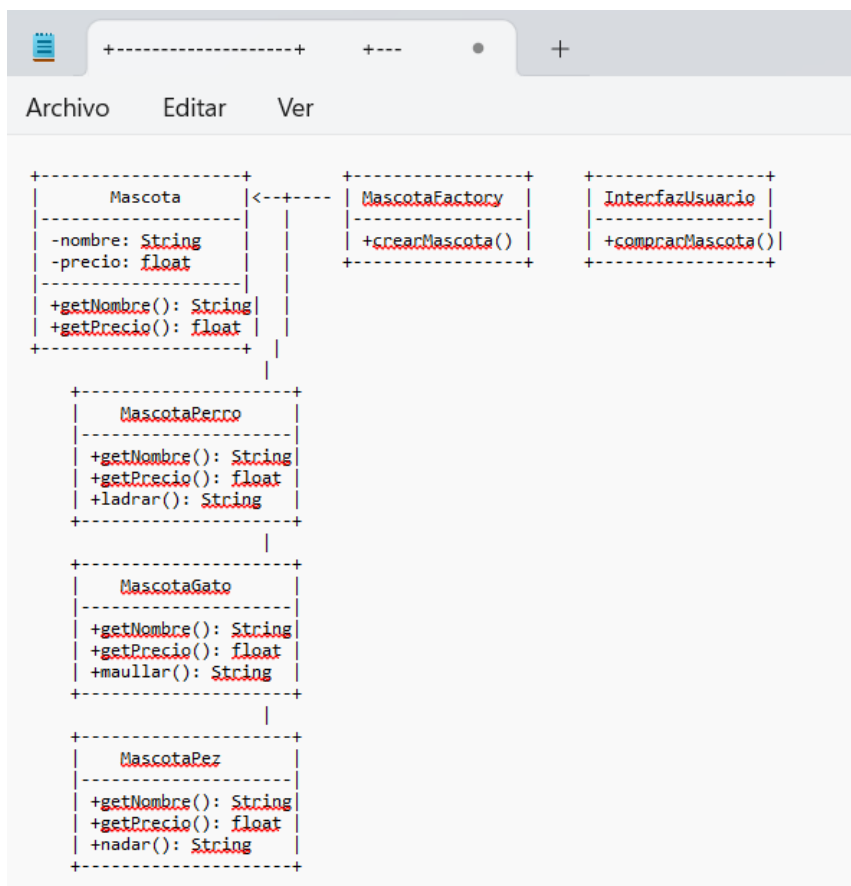
Tarea 1.6: pattern design with javascript

Factory pattern

Es uno de los patrones de diseño más utilizados en Java. Este tipo de patrón de diseño se incluye en el patrón de creación, ya que este patrón proporciona una de las mejores formas de crear un objeto.

En el patrón Factory, creamos un objeto sin exponer la lógica de creación al cliente y nos referimos al objeto recién creado utilizando una interfaz común.

Diagrama de clases



Explicación

La clase **Mascota** es una clase base que define los atributos comunes (nombre y precio) y métodos (`getNombre` y `getPrecio`) para todas las mascotas.

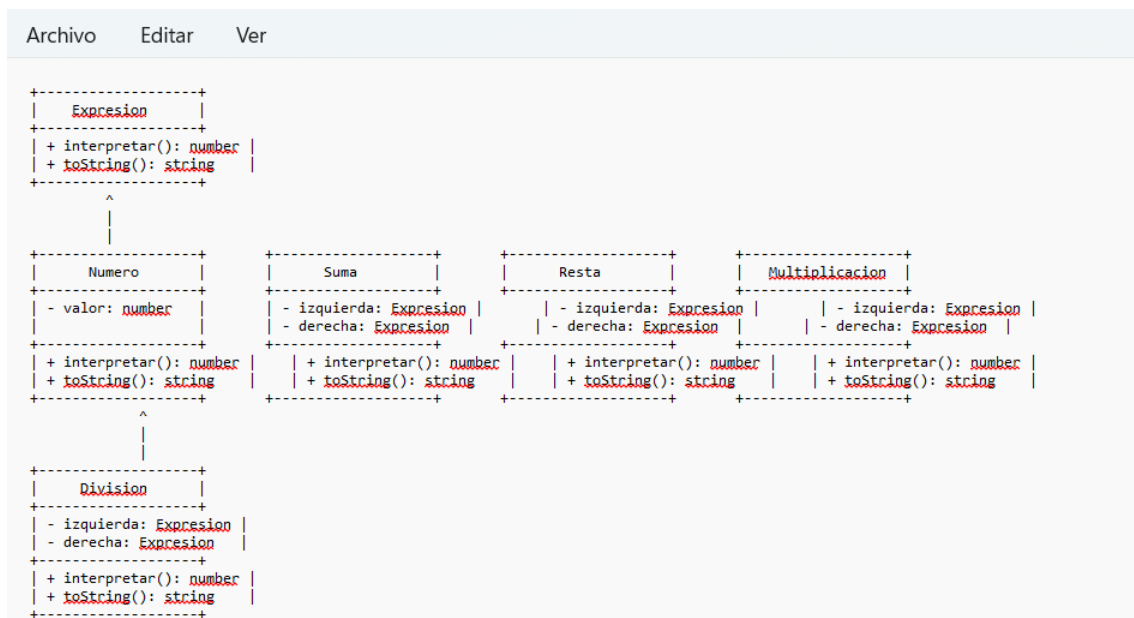
Las clases **MascotaPerro**, **MascotaGato** y **MascotaPez** heredan de **Mascota** y añaden métodos específicos (`ladrar`, `maullar`, `nadar`).

La clase MascotaFactory contiene un método estático crearMascota que, basado en el tipo de mascota seleccionado, retorna una instancia de la clase correspondiente.

Interpreter pattern

El patrón de diseño del intérprete de Java se incluye en los patrones de diseño de comportamiento. Se utiliza para proporcionar una forma de evaluar la gramática del idioma y proporciona un intérprete para manejar esta gramática.

Diagrama de clases



Explicación

Creamos clases como Numero, Suma, Resta, Multiplicacion y Division que representan diferentes partes y operaciones de una expresión matemática.

Utilizamos una función llamada parse para convertir una cadena de entrada que contiene una expresión matemática (como "3 + 5 * 2") en una estructura de árbol de objetos de las clases creadas.

La función interpretar recorre este árbol de objetos creado por parse.