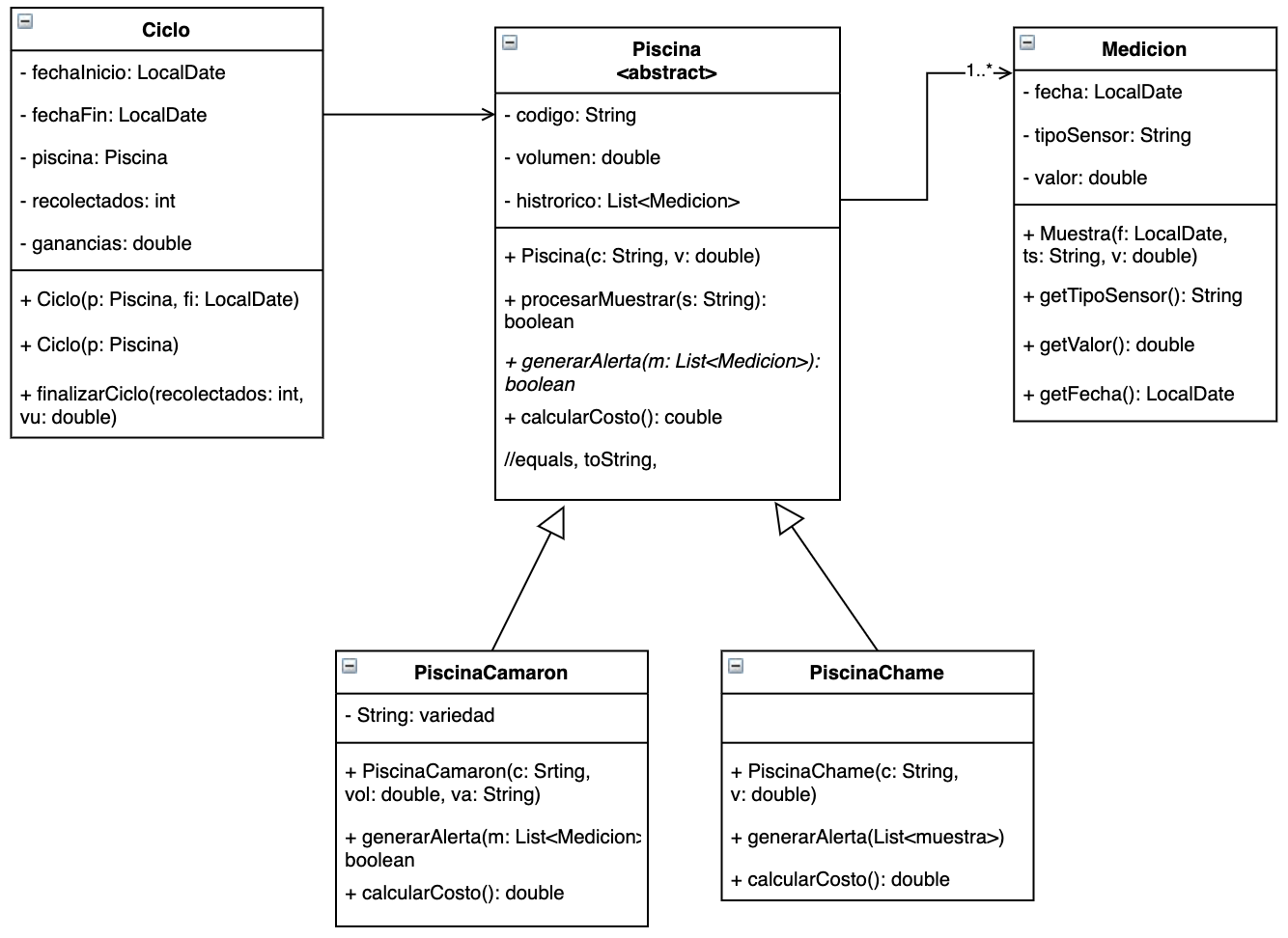
Una empresa se dedica a la crianza de camarones y peces chames en piscinas. Con el propósito de monitorear el estado de los animales en las piscinas hay sensores que miden algunos aspectos ambientales de las mismas. Los periodos de crianzas se denominan ciclos. Un ciclo va desde que colocan las semillas de los animales de las piscinas hasta que se recolectan los animales de las misma.  A usted se le da el diagrama de clases del sistema



En base al diagrama de clases anterior realice los siguiente:

**Clase Piscina -  23 puntos**

* **(1 puntos)** Modifique la declaración de la clase Piscina para que sea abstracta.
* **(2 puntos)** Declare el método **generarAlerta(medicionesdia: List<Medidion>): boolean - este método es abstracto.** El método retorna verdadero cuando los datos recolectados de los sensores de un día estén fuera de los rangos óptimos.
* **(12 puntos)** Implemente el método **procesarMuestra(muestra: String): boolean** - este método recibe una cadena de caracteres con los datos recolectados de los sensores en la piscina en un día.

La cadena muestra tiene el siguiente formato:

¨tiposensor1|valor1 **,** tiposensor2|valor2,...¨

Genere una medición por cada dato recolectado y agregue a la lista de histórico. Finalmente llame a la función **generarAlerta** con las mediciones del día recibidas y retorne el valor devuelto por esta función.

* **(5 puntos)** Implemente el método **equals** - dos piscinas son iguales si tienen el mismo código y el mismo volumen.
* **(3 puntos)** Implemente el método **calcularCosto(): double -** este metodo retorna el costo de mantenimiento de la piscina. El costo del mantenimiento es el volumen de la piscina por 0.10

**PiscinaCamaron - 21 puntos**

* **(2 puntos)** Modifique la declaración de la clase para que herede de Piscina.
* **(2 puntos)** Variables de instancia de la clase
* **(4 puntos)** Constructor de la clase.
* **(2 puntos)** Implemente el método get de la variable de instancia variedad.
* **(8 puntos)** Implemente el método **generarAlerta(medicionesdia: List<Medicion>): boolean -** El método recibe una lista con las mediciones de un día y retorna verdadero cuando los valores de temperatura y ph están fuera del rango óptimo

Los rangos óptimos son:

**Temperatura** entre 26 y 30

**PH** entre 6 y 8

Asuma que en la lista **medicionesdia** están las dos mediciones requeridas.

* **(3 puntos)** Implemente el método **calcularCosto(): double -** este metodo retorna el costo de mantenimiento de la piscina mas 100 dolares de aditivos especiales (Debe llamar **calcularCosto()** del padre)

**Clase Ciclo -  7 puntos**

* **(3 puntos)** Constructor que reciba la piscina y fije la fecha de inicio a la actual. **Debe llamar al constructor que reciba todos los parámetros.**
* **(4 puntos)** Método **finalizarCiclo(int numerorecolectado, double valorunidad) -** Este método fija la fecha de fin de ciclo a la fecha actual, el número de especímenes recolectados al valor recibo y calcula la ganancia obtenida y la fije.

ganancias = ingresos - costos ingresos

ingresos = valorunidad \* recolectados

**Clase Main (7 puntos)**

Pida al usuario una variedad de camaron y muestre el total de gananacias obtenidas por los piscinas de esa variedad de camaron.