Práctica 12

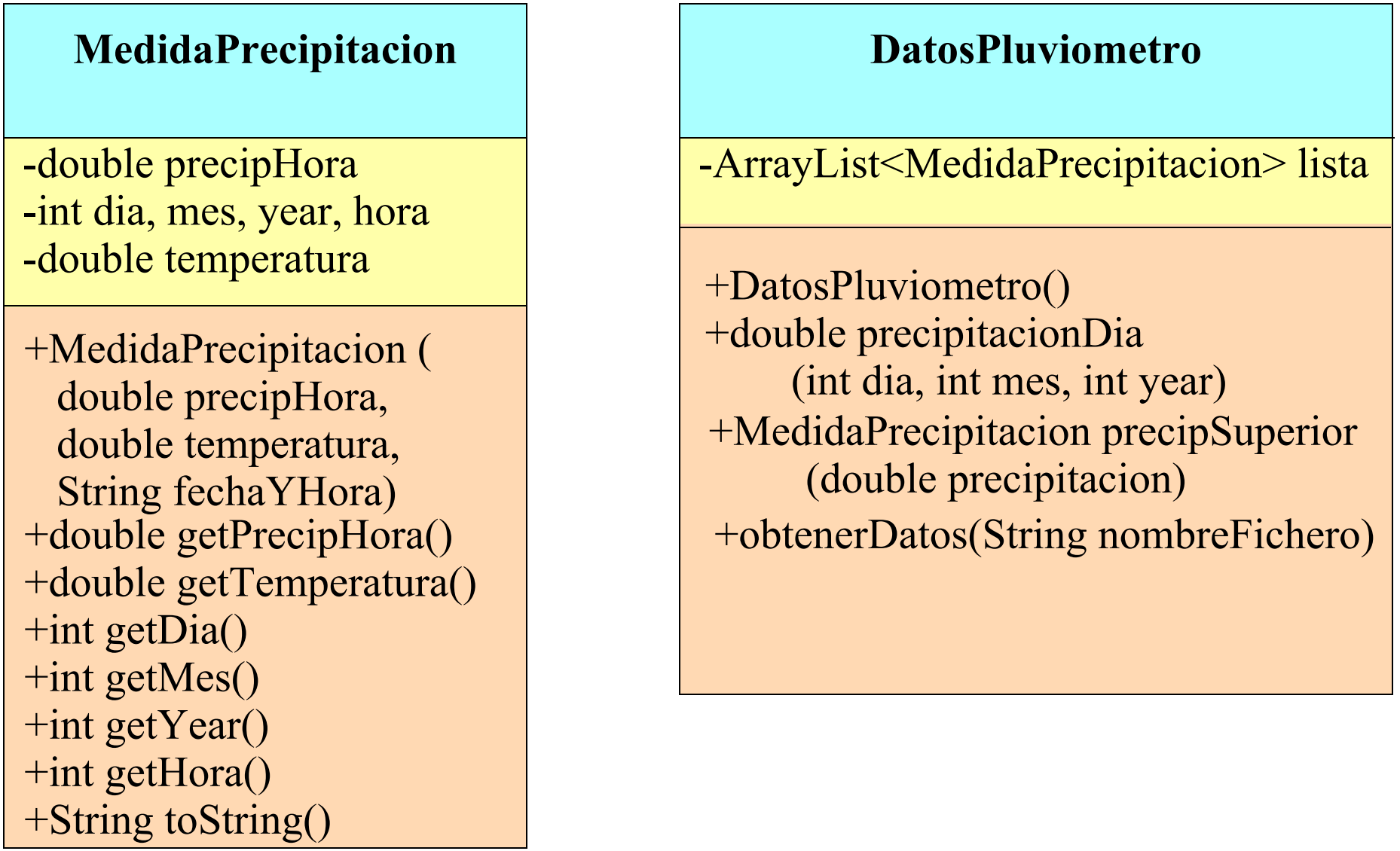
Objetivos: Usar `ArrayList`, excepciones y leer ficheros de texto

Descripción: Se desea escribir parte del software de análisis de las medidas obtenidas por un pluviómetro

Se dispone de una clase llamada `MedidaPrecipitacion` cuyos objetos contienen una medida de precipitación recogida en una hora (en litros/m^2), así como la temperatura (en ºC) y la hora, día, mes y año

- el constructor recoge en sus parámetros los valores iniciales de la precipitación y temperatura y toma la hora, día, mes y año del String `fechaYHora`, con el siguiente formato: `dd/MM/yyyy HH:mm`
  Ejemplo: `10/07/2018 12:32`
- los métodos son observadores de los distintos atributos, excepto `toString()` que retorna un texto que contiene todos los datos de la medida

Diagrama de clases



Clase `DatosPluviometro`

Se pide implementar en Java la clase `DatosPluviometro`

Atributos: un único atributo que es una lista de objetos de la clase `MedidaPrecipitacion`

Métodos:

- `DatosPluviomentro`(): Crea la lista vacía
- `precipitacionDia`(): recorre la lista en busca de medidas de precipitación correspondientes al día, mes y año indicados por los parámetros. Suma las precipitaciones de todas estas medidas (que corresponden a diferentes horas) y retorna esa suma
- `precipSuperior`(): Busca en la lista el primer objeto cuya precipitación en una hora sea superior a `precipitacion`, y lo retorna. Si no hay ninguno, retorna `null`

Clase **DatosPluviometro**

- `obtenerDatos`(): Lee los datos de precipitación del fichero cuyo nombre es `nombreFichero` y los mete en la lista. Si el fichero no existe se muestra un mensaje de error y se deja la lista vacía. El fichero tiene una línea de encabezamiento que habrá que ignorar y datos de precipitación, uno por línea, con el formato (en inglés) que se muestra en este ejemplo:

```
Fecha       Hora      l/m^2  Temp(C)
08/05/2013 10:00     0.2    21.8
08/05/2013 11:00     0.1    21.1
08/05/2013 12:00     0.0    20.4
08/05/2013 13:00     0.0    19.7
08/05/2013 14:00     0.1    18.9
...
```

Programa principal

El programa principal suministrado en una clase aparte hace lo siguiente:

- crea un objeto de la clase `DatosPluviometro`
- invoca a su método `obtenerDatos()`
- muestra en pantalla la precipitación del 20 de mayo de 2014
- muestra en pantalla los datos del primer objeto con precipitación superior a 4 l/m^2
- muestra en pantalla los datos del primer objeto con precipitación superior a 20 l/m^2

Parte avanzada: usar excepciones

Crear una excepción llamada `NoHayMedidas` en una clase aparte

Modificar `precipitacionDia`() para que si no encuentra ningún objeto de la fecha indicada lance `NoHayMedidas`

Añadir instrucciones al programa principal para que si se lanza `NoHayMedidas` se ponga un mensaje de error en pantalla

- Se deben probar las dos situaciones: una con una fecha que exista y otra con una fecha para la que no haya datos
- Los dos últimos pasos del programa principal (los que usan `precipSuperior`()) deben seguir haciéndose aunque haya una excepción

Entrega

El proyecto Bluej en un archivo comprimido

Informe:

- Parte básica:
  - Código de la clase `DatosPluviometro`
  - Captura de los *resultados* de la ejecución del programa principal
- Parte avanzada:
  - Código del método `precipitacionDia()` y del *programa principal*, modificados
  - Captura de los *resultados* de la ejecución del programa principal