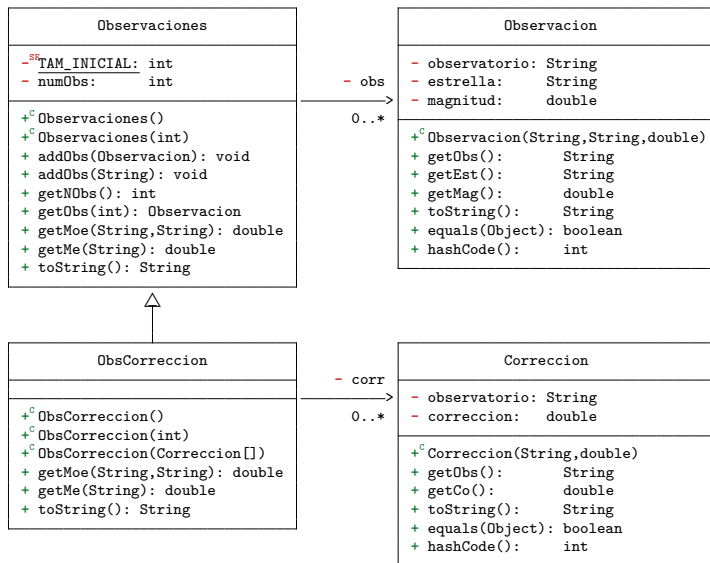


## El diagrama de clases UML



**Nota:** se pueden añadir los métodos **privados** y **protegidos** que se consideren necesarios. Además, si se considera que la implementación algún método de la superclase es adecuado, entonces se puede dejar sin implementar el método de la subclase.

### ✓ La clase Observacion

La clase `Observacion` (del paquete `prControl16`) contiene información sobre la magnitud luminosa de una determinada estrella registrada por un determinado observatorio.

- El constructor recibe, en el siguiente orden, el nombre del observatorio, el nombre de la estrella, y el valor de la magnitud luminosa observada.
- Tres métodos, `getObs()`, `getEst()` y `getMag()`, permiten acceder a los valores almacenados en el objeto.
- El método `toString()` proporciona la representación textual del objeto, en el siguiente formato: `(Mauna Kea; Rigel; 0.13248)`.
- Se considera que dos observaciones son *iguales* si el nombre del observatorio y de la estrella son iguales (indistintamente del caso de las letras).

### ✓ La clase Observaciones

La clase `Observaciones` (del paquete `prControl16`) almacena múltiples instancias de la clase `Observacion` en un array, así como el número total de observaciones que contiene almacenados.

- ▶ `Observaciones()`  
// Construye el objeto vacío (sin observaciones) con un array inicial de tamaño 16.
- ▶ `Observaciones(int)`  
// Construye el objeto vacío (sin observaciones) con un array inicial del tamaño recibido como parámetro.
- ▶ `addObs(Observacion): void`  
// Añade la observación recibida como parámetro. Si el array está lleno, se debe duplicar su capacidad.  
// Si ya existe una observación para esa misma estrella, por el mismo observatorio, entonces se reemplaza la anterior observación por la nueva.
- ▶ `addObs(String): void`  
// Añade la observación recibida como parámetro. Si el array está lleno, se debe duplicar su capacidad.  
// Si ya existe una observación para esa misma estrella, por el mismo observatorio, entonces se reemplaza la anterior observación por la nueva.  
// El parametro representa la información relativa a una observación, como un `String`, separadas por el símbolo `;` (seguido por espacios opcionales), donde el punto `.` es el separador de los decimales.

```

// Por ejemplo: Mauna Kea; Rigel; 0.13248
// Lanza la excepcion IllegalArgumentException si el formato del parámetro es erróneo
► getNObs(): int
// Devuelve el número de observaciones actualmente almacenadas.
► getObs(int): Observacion
// Devuelve la Observación almacenada en la posición especificada por el parámetro.
// Lanza la excepción NoSuchElementException si el valor del parámetro es erróneo.
► getMoe(String,String): double
// Devuelve (Moe) la magnitud registrada por el observatorio especificado, para la estrella especificada,
// indistintamente del caso de las letras que los componen.
// Lanza la excepción NoSuchElementException si la observación no existe.
► getMe(String): double
// Devuelve (Me) la magnitud media para la estrella especificada como parametro, indistintamente del caso
// de las letras que los componen. Para ello, se calcula la media de todas las observaciones (Moe) de
// esa estrella que estén almacenadas.  $Me = 1/n \sum_i Moe$ 
// Lanza la excepción NoSuchElementException si no hay observaciones de esa estrella.
//
//


| Moe       | Rigel   | Vega    |
|-----------|---------|---------|
| Mauna Kea | 0.13248 | 0.02673 |
| Roque ... | 0.13064 | 0.02627 |


//


| Me | 0.13156 | 0.02650 |
|----|---------|---------|
|----|---------|---------|


//
► toString(): String // [Redefinición]
// Devuelve la representación textual de todas las observaciones almacenadas, en el siguiente formato:
// [(Roque de los Muchachos; Rigel; 0.13064), (Mauna Kea; Vega; 0.02673),
// (Mauna Kea; Rigel; 0.13248), (Roque de los Muchachos; Vega; 0.02627)]

```

## ✓ La aplicación PruebaObservaciones

Desarrolle una aplicación (en el paquete anónimo) que permita realizar una prueba de las clases anteriores. Por ejemplo, para los datos anteriormente expuestos:

```

Observaciones:
[(Roque de los Muchachos; Rigel; 0.13064), (Mauna Kea; Vega; 0.02673), (Mauna Kea; Rigel; 0.13248),
(Roque de los Muchachos; Vega; 0.02627)]
Moe(Mauna Kea, Rigel): 0.13248
Moe(Mauna Kea, Vega): 0.02673
Moe(Roque de los Muchachos, Rigel): 0.13064
Moe(Roque de los Muchachos, Vega): 0.02627
Me(Rigel): 0.13156
Me(Vega): 0.026500000000000000

```

## ✓ La clase Correccion

La clase Correccion (del paquete prControl16) contiene información sobre el factor de corrección a aplicar a las magnitudes registradas por un determinado observatorio.

- El constructor recibe, en el siguiente orden, el nombre del observatorio, y el valor del factor de corrección a aplicar.
- Dos métodos, getObs() y getCo() permiten acceder a los valores almacenados en el objeto.
- El método toString() proporciona la representación textual del objeto, en el siguiente formato: (Mauna Kea; 0.99222).
- Se considera que dos correcciones son iguales si el nombre del observatorio es igual (indistintamente del caso de las letras).

## ✓ La clase ObsCorreccion

La clase ObsCorreccion (del paquete prControl16) extiende y redefine el comportamiento de la clase Observaciones. Para ello, almacena múltiples instancias de la clase Correccion en un array.

```

► ObsCorreccion()
  // Construye el objeto vacío, sin observaciones ni correcciones.
► ObsCorreccion(int)
  // Construye el objeto vacío, sin observaciones ni correcciones, pero el array de observaciones
  // (de la clase padre) con un tamaño inicial del tamaño recibido como parámetro.
► ObsCorreccion(Correccion[])
  // Construye el objeto vacío, sin observaciones, pero recibe y almacena el array
  // conteniendo el factor de corrección (Co) para los observatorios.
► getMoe(String,String): double // [Redefinición]
  // Devuelve la magnitud corregida (M'oe) registrada por el observatorio especificado, para la estrella
  // especificada, indistintamente del caso de las letras que los componen, después de aplicar el factor
  // de corrección almacenado para el observatorio especificado.  $M'oe = (Moe * Co)$ 
  // Si el factor de corrección (Co) para ese observatorio no existe, entonces será considerado como (1).
  // Lanza la excepción NoSuchElementException si la observación no existe.
  //
  //


| Moe       | Rigel   | Vega    | Co      | M'oe      | Rigel   | Vega    |
|-----------|---------|---------|---------|-----------|---------|---------|
| Mauna Kea | 0.13248 | 0.02673 | 0.99222 | Mauna Kea | 0.13144 | 0.02652 |
| Roque ... | 0.13064 | 0.02627 | 1.00789 | Roque ... | 0.13167 | 0.02647 |


  //
  //
► getMe(String): double // [Redefinición]
  // Devuelve la magnitud media corregida (M'e) para la estrella especificada como parámetro,
  // indistintamente del caso de las letras que los componen. Para ello, se calcula la media
  // de todas las observaciones corregidas de esa estrella que estén almacenadas.
  //  $M'e = 1/n \sum_i M'oe$ 
  // Lanza la excepción NoSuchElementException si no hay observaciones de esa estrella.
  //


| Moe       | Rigel   | Vega    | Co      | M'oe      | Rigel   | Vega    |
|-----------|---------|---------|---------|-----------|---------|---------|
| Mauna Kea | 0.13248 | 0.02673 | 0.99222 | Mauna Kea | 0.13144 | 0.02652 |
| Roque ... | 0.13064 | 0.02627 | 1.00789 | Roque ... | 0.13167 | 0.02647 |


  //
  //


| M'e | Rigel   | Vega    |
|-----|---------|---------|
| M'e | 0.13156 | 0.02649 |


  //
  //
► toString(): String // [Redefinición]
  // Devuelve la representación textual de todas las observaciones y correcciones almacenadas,
  // en el siguiente formato:
  // [[(Roque de los Muchachos; Rigel; 0.13064), (Mauna Kea; Vega; 0.02673),
  // (Mauna Kea; Rigel; 0.13248), (Roque de los Muchachos; Vega; 0.02627)],
  // [(Roque de los Muchachos; 1.00789), (Mauna Kea; 0.99222)]]

```

## ✓ La aplicación PruebaObsCorreccion

Desarrolle una aplicación (en el paquete anónimo) que permita realizar una prueba de las clases anteriores. Esta aplicación deberá utilizar explícitamente los mecanismos de **herencia**, **polimorfismo** y **vinculación dinámica** proporcionados por Java, se deberán identificar mediante un comentario aquellos lugares donde sean utilizadas estas características. Por ejemplo, para los datos anteriormente expuestos:

```

Observaciones/Correcciones:
[[ (Roque de los Muchachos; Rigel; 0.13064), (Mauna Kea; Vega; 0.02673), (Mauna Kea; Rigel; 0.13248),
  (Roque de los Muchachos; Vega; 0.02627)], [(Roque de los Muchachos; 1.00789), (Mauna Kea; 0.99222)]]
M'oe(Mauna Kea, Rigel): 0.13144930559999998
M'oe(Mauna Kea, Vega): 0.0265220406
M'oe(Roque de los Muchachos, Rigel): 0.1316707496
M'oe(Roque de los Muchachos, Vega): 0.0264772703
M'e(Rigel): 0.1315600276
M'e(Vega): 0.02649965545

```