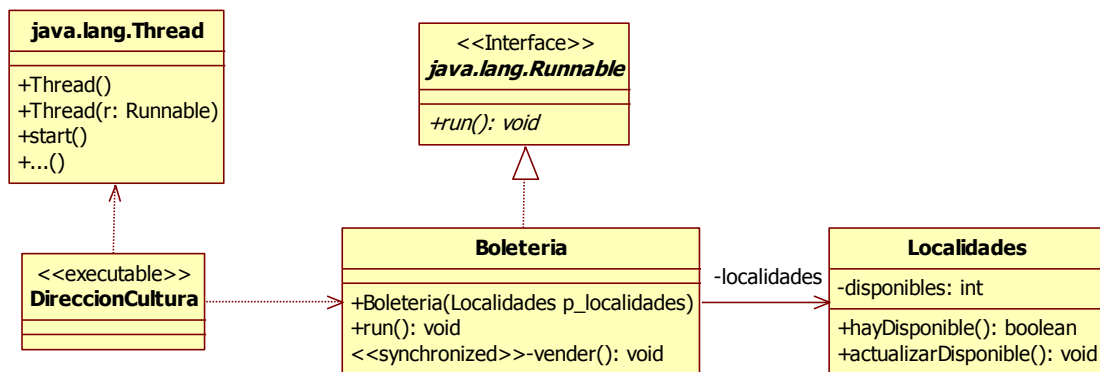
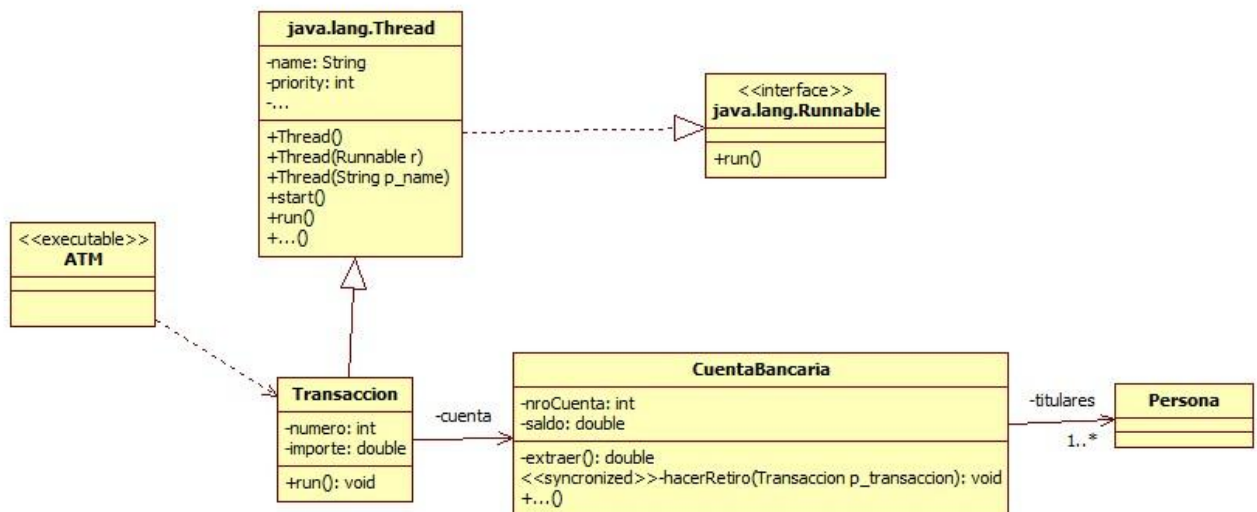


**Objetivo:** Que el alumno comprenda, a través de la aplicación práctica, los conceptos avanzados que implementan los lenguajes de programación orientados a objetos, tales como la concurrencia.

1. Como cierre de una serie de eventos programados para la celebración de los 100 años del Teatro Vera de la Ciudad de Corrientes, la Dirección de Cultura organiza un evento de categoría internacional, de acceso libre y gratuito. A fin de no superar la capacidad física del predio en el que se realiza el evento se prevé la entrega de 12000 entradas. Para la entrega de las mismas la Dirección de Cultura habilita tres boleterías (Anfiteatro, Teatro Vera y Dirección de Cultura), que atenderán en forma simultánea.  
Se solicita implementar una aplicación que permita emitir automáticamente las entradas, numerándolas desde el 1 hasta el número total de localidades disponibles. Para darle mayor flexibilidad a la aplicación este valor se ingresará como parámetro. Cabe señalar que como las ventas se realizarán en forma simultánea, más de un proceso estará accediendo a la vez a un mismo recurso (las localidades), razón por la cual será necesario sincronizar el acceso a dicho recurso, a fin de utilizarlo de forma consistente.

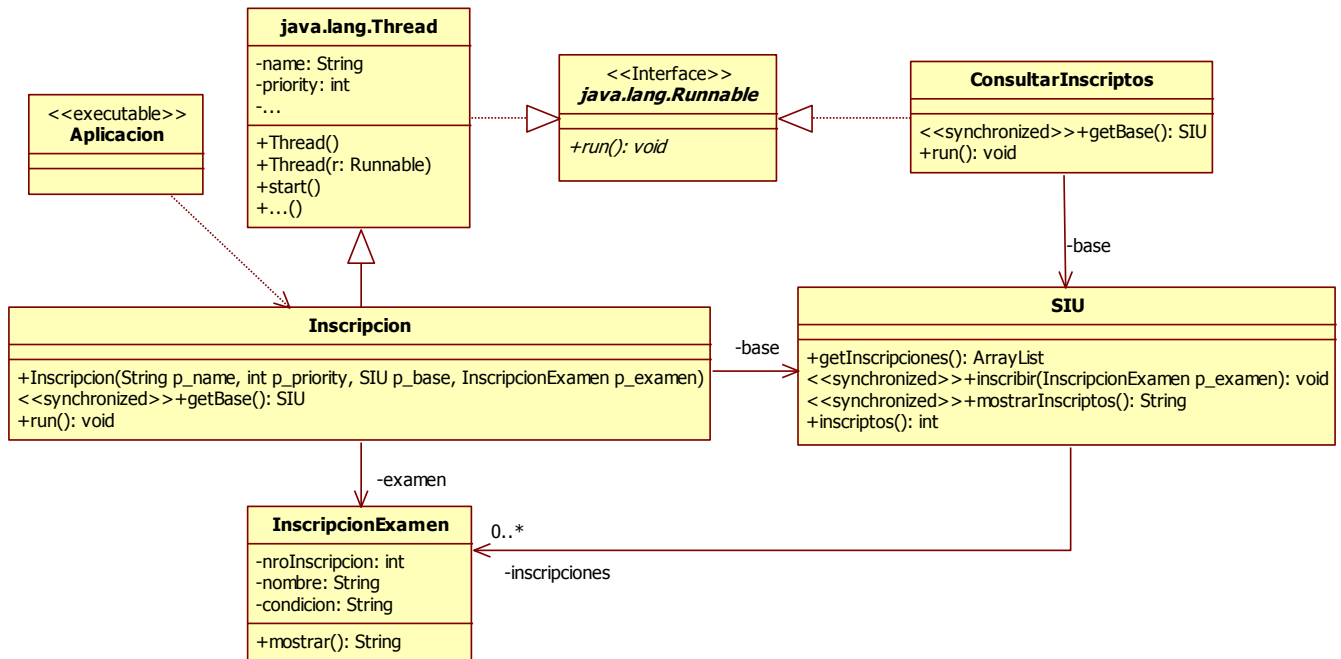


2. Se desea mantener el control del saldo de una cuenta corriente de un banco, que puede tener más de un titular. Cada titular puede realizar extracciones de dinero siempre que la cuenta tenga el saldo necesario para realizar la operación, de lo contrario, se avisará al mismo con un mensaje indicando que no hay dinero suficiente, y se mostrará el saldo actual. El diseño del problema se presenta en el siguiente diagrama de clases:



**Nota:** puede utilizar las clases definidas en TP anteriores, o crear nuevas clases simplificadas, que sirvan al propósito de este enunciado (CuentaBancaria-Persona).

3. El Sistema de Información Universitario (SIU) permite la consulta de la lista de inscriptos a un examen, en forma simultánea a la inscripción de los alumnos, a la cual los alumnos también pueden acceder en forma simultánea.  
El diseño de la solución a este requerimiento se presenta en el siguiente diagrama de clases:



La clase **IncripcionExamen** es la responsable de administrar los datos del examen al que se inscribe un alumno, y genera una cadena con el siguiente formato:

```
Numero inscripcion: 1 - Andres Perez - Libre
```

La clase **SIU** mantiene la colección de inscripciones, la cual representa a la "base de datos" que contiene las inscripciones a exámenes. Permite inscribir un alumno, mediante el registro del examen que recibe como parámetro. Al solo efecto de visualizar el correcto funcionamiento, al momento de inscribir también muestra un mensaje con el nombre del hilo que se inscribió, con el siguiente formato:

```
Pedro se inscribio
```

En este ejemplo, "Pedro" es el nombre del hilo.

Esta clase también permite conocer la cantidad de inscriptos, y visualizar la lista de inscriptos (los elementos de la colección), con el siguiente formato:

```
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Marcos Acosta - Libre
*****
```

Las clases **Incripcion** y **ConsultaInscriptos** serán los hilos que correrán en forma simultánea, por tanto más de un proceso estará accediendo a la vez a un mismo recurso (las inscripciones), razón por la cual será necesario sincronizar el acceso a dicho recurso, a fin de utilizarlo de forma consistente.

La clase **Incripcion**, en la implementación del método `run()`, usa el método `inscribir()` de su base, al que proporciona como parámetro un examen.

La clase **ConsultarInscriptos**, en la implementación del método `run()`, simula varias consultas, que podrían implementarse con una estructura repetitiva para un cierto número de veces, en las cuales, además de mostrar el nombre del hilo, usa el método `mostrarInscriptos()` de su base. Produce una salida en pantalla con el siguiente formato, mostrando todos los inscriptos al momento de la consulta:

```
Profesor consulto los estudiantes inscriptos:
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Marcos Acosta - Libre
*****
```

En este ejemplo, "Profesor" es el nombre del hilo.

En la clase ejecutable se debe instanciar una base SIU, que luego se pasará como parámetro al instanciar una consulta. Luego instanciar varios exámenes, que serán los parámetros al momento de crear los hilos de la clase **Inscripcion**, a la que además se debe proporcionar como parámetros un nombre y una prioridad, y la base creada previamente.

Por último, se debe iniciar la ejecución de todos los hilos.

Una posible salida en pantalla al ejecutar es la siguiente:

```
Andres se inscribio
Nicolas se inscribio
Pedro se inscribio

Profesor consulto los estudiantes inscriptos:
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Pedro Gomez - Regular
*****

Profesor consulto los estudiantes inscriptos:
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Pedro Gomez - Regular
*****

Jose se inscribio

Profesor consulto los estudiantes inscriptos:
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Pedro Gomez - Regular
Numero inscripcion: 4 - Jose Lopez - Regular
*****

Marcos se inscribio
Juan se inscribio

Profesor consulto los estudiantes inscriptos:
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Pedro Gomez - Regular
Numero inscripcion: 4 - Jose Lopez - Regular
Numero inscripcion: 5 - Marcos Acosta - Libre
Numero inscripcion: 6 - Juan Gonzalez - Regular
*****

Profesor consulto los estudiantes inscriptos:
*****
Numero inscripcion: 1 - Andres Perez - Libre
Numero inscripcion: 2 - Nicolas Marquez - Regular
Numero inscripcion: 3 - Pedro Gomez - Regular
Numero inscripcion: 4 - Jose Lopez - Regular
Numero inscripcion: 5 - Marcos Acosta - Libre
Numero inscripcion: 6 - Juan Gonzalez - Regular
*****

Cerro la inscripcion !!!!
```