|  |
| --- |
| cifp Juan de colonia |
| Programación de Procesos y Servicios |
| UT2: Práctica Cuenta Bancaria |
|  |
| **Alexis López Briongos Dam2t** |
| **24/11/2023** |

|  |
| --- |
|  |

*Índice*

[Clase CuentaBancaria 2](#_Toc151740266)

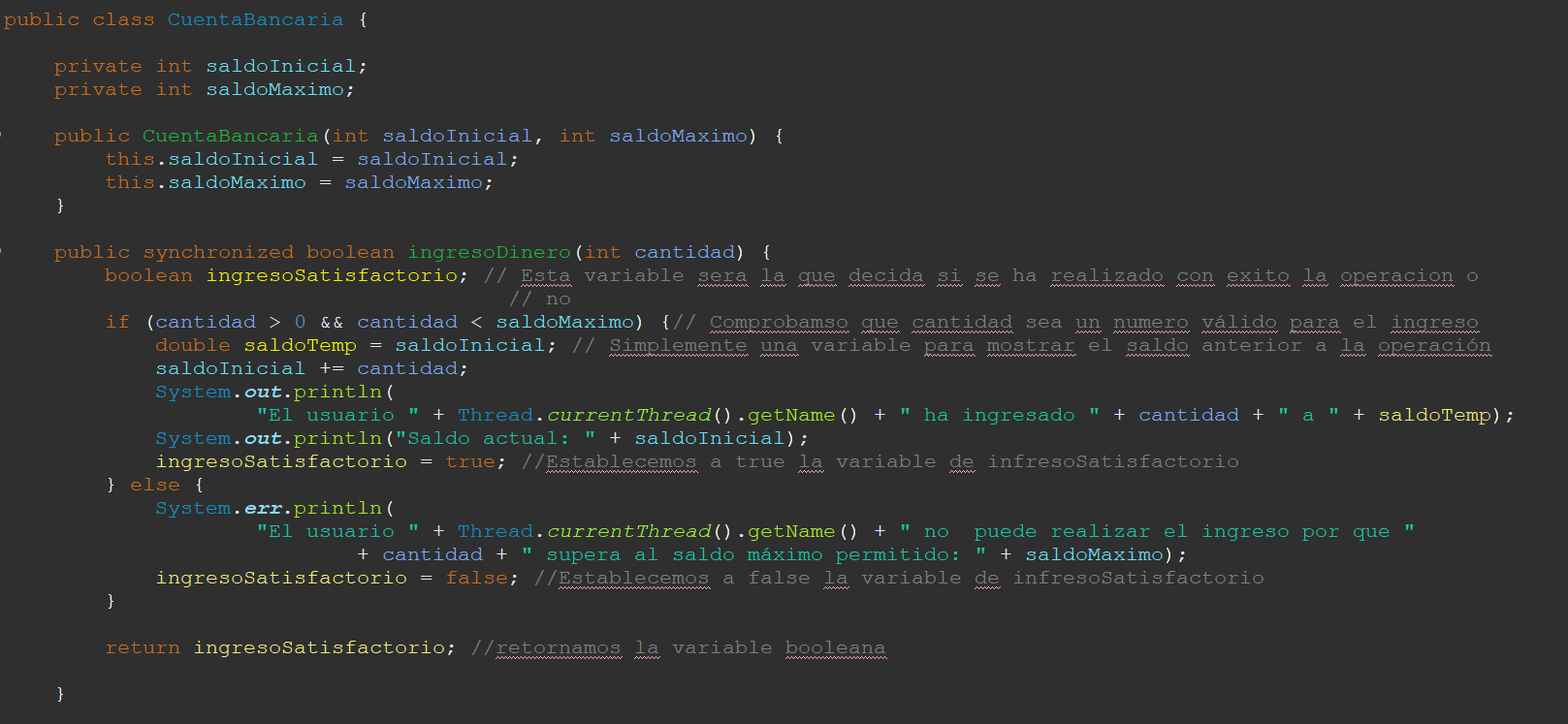
[Clase Persona 3](#_Toc151740267)

[Clase Cajero Automático 4](#_Toc151740268)

[Resultado consola 5](#_Toc151740269)

# Clase CuentaBancaria

* En esta clase vamos a controlar las operaciones de ingreso o retiro de una cuenta bancaria.



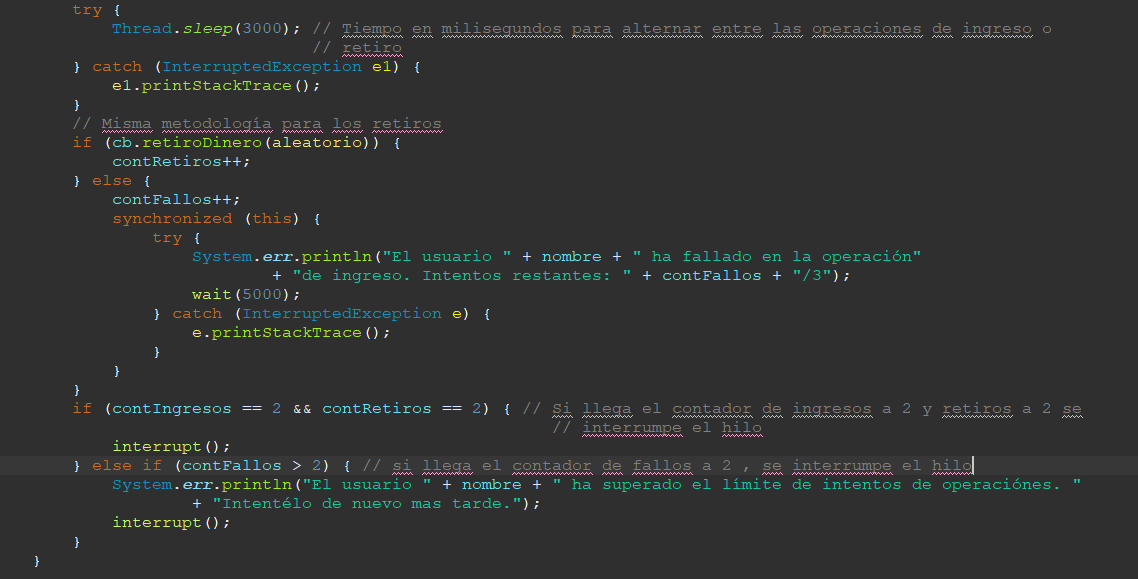
* Comprobamos si la cantidad a ingresar es válida y en consecuencia agregamos el saldo o no y cambiamos el valor de la variable booleana con lo que trabajaremos en la clase Persona para poder identificar los ingresos o retiros satisfactorios o los errores cometidos en las operaciones.
* Misma metodología para el método de retirar dinero.

# Clase Persona

* En esta clase controlaremos el acceso de las operaciones de la clase CuentaBancaria.

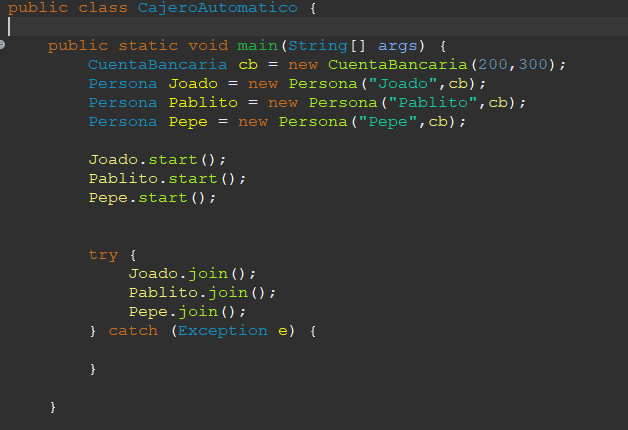


* En el constructor agregamos el nombre de la persona y la cuenta bancaria con la que trabajaremos.
* En el método run() realizamos un bucle while hasta que el hilo sea interrumpido (más adelante veremos cómo lo hacemos).
* Inicializamos una variable de tipo int para que la cantidad de retiro o ingreso sea aleatoria. Comprobamos con el método ingresoDinero() pasándole la cantidad aleatoria si es válida y se ha podido realizar el ingreso o no.
* En consecuencia al resultado del método ingresoDinero() vamos a aumentar el contadorIngresos el cual controla el numero de ingresos que podemos realizar por usuario (2 por operación). Si no se cumple la condición aumentamos el contador de Fallos (3 max) y ponemos en espera al hilo durante 5000 milisegundos (5 segundos) siendo necesario establecer como sincronizado el bloque en el que vamos a ejecutar este método wait().
* Si estas condiciones anteriormente mencionadas se cumplen se interrumpe el hilo.



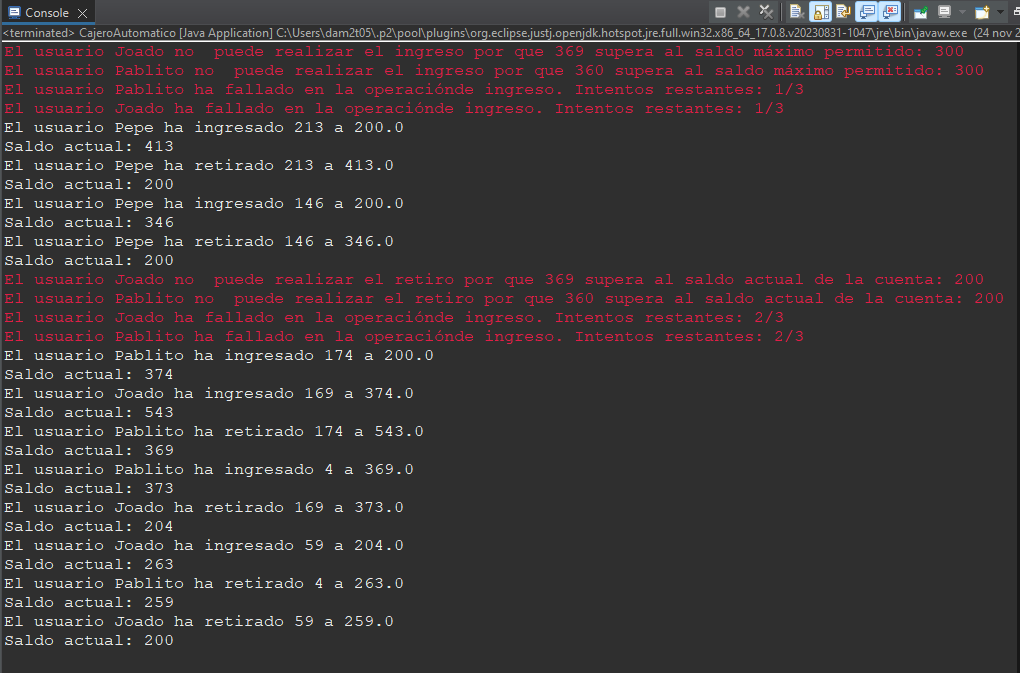
* Establecemos un periodo de espera entre las operaciones Ingreso y retiro con un Thread.sleep() de 3000 milisegundos(3 segundos).
* Evaluamos las variables contIngresos , contRetiros y contFallos por cada iteración. Si se cumple cualquiera de estas condiciones definidas el hilo se interrumpe.

# Clase Cajero Automático



* En esta clase simplemente instanciamos las clases, asignamos misma cuenta bancaria para que todas operen en la misma e inicializamos los hilos recordando que la clase Persona extiende de Thread.

# Resultado consola



* Este sería el resultado de la ejecución de los hilos Persona.
* Podemos visualizar que Joado realiza una operación errónea y no puede volver a intentar una operación hasta pasado 5 segundos. Mismo caso con Pablito.
* Sin embargo Pepe realiza los ingresos y los retiros de forma intercalada sin ningún error ocasionado pudiendo finalizar su hilo con total normalidad.
* En el caso de Joado y Pablito al no haber cometido 3 errores pueden volver a intentar realizar operaciones y así las realizan de forma intercalada hasta completar su número de operaciones.