**CONTROL 1**

**Programación Orientada a Objeto, 2024-2**

Profesores: Ma. Antonieta Soto Ch. (S1), Miguel Romero V. (S2) Fecha: 19-08-2024

Estudiante: Tiempo: 11:10 a 13:00

La Figura 1 muestra la representación UML de un diagrama de clases parcial que permitiría implementar un sistema de una institución financiera para llevar registro de las transacciones asociadas a sus cajeros automáticos.

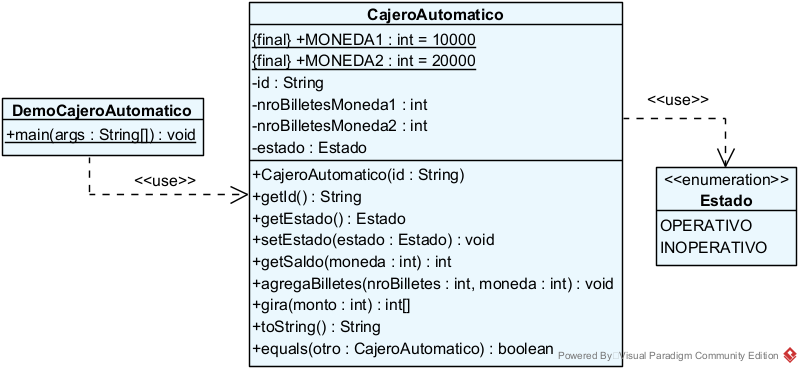


Figura 1. Diagrama de clases de Cajeros Automáticos.

**Se pide**:

Implementar los elementos presentes en el diagrama de clases, tal y como allí aparecen.

Para lo anterior tome en consideración los antecedentes adicionales que se dan a continuación.

**a) Debe ajustarse al diagrama de clases.** No agregue atributos, métodos, constructores u otro elemento en su implementación. Solo puede agregar métodos privados, si ellos permiten simplificar, reutilizar o hacer más legible el código.

**b) Clase CajeroAutomatico**

Respecto de esta clase contemple lo siguiente:

* Cuando se crea un cajero automático, este deberá tener estado operativo y no incluir dinero.
* El método getId() retorna el identificador del cajero con todas los caracteres alfabéticos en mayúsculas.
* El método getSaldo retorna el monto almacenado en el cajero considerando solo los billetes que posea de la moneda indicada como parámetro.
* El método toString retorna el id (con caracteres alfabéticos en mayúsculas), el saldo total considerando todos los billetes del cajero y el estado de este en minúsculas. Los datos se deben separar mediante una coma y un espacio.
* Dos objetos CajeroAutomatico son iguales si sus identificadores son iguales.
* El método agregaBilletes aumenta la cantidad de billetes que corresponda considerando la moneda que se pasa como parámetro. Si el cajero está inoperativo, no se debe realizar la acción.
* El método gira retorna un arreglo unidimensional con dos posiciones, la primera indica la cantidad de billetes que el cajero entregó en MONEDA1 y la segunda la cantidad de billetes que el cajero entregó en MONEDA2.

Las reglas para definir qué billetes entregar dado el monto pasado como parámetro son:

* Primero tratar de cubrir el monto con billetes de la moneda mayor. Si queda un saldo no cubierto, completar con billetes de la moneda menor.
* Si no se puede cubrir completamente el monto solicitado no se realiza el giro, ello implica que se debe indicar 0 billetes de cada moneda.
* Si un cajero está inoperativo, se debe retornar 0 en cada posición del arreglo.

**c) Clase DemoCajeroAutomatico**

El método main debe:

* Utilizar un objeto de la clase ArrayList para almacenar objetos CajeroAutomatico.
* Crear 10 cajeros, leyendo los datos necesarios. No olvide almacenar cada cajero en la colección.
* Dejar inoperativo en octavo cajero creado.
* Agregar $50.000 en billetes de $10.000 y $80.000 en billetes de $20.000 a todos los cajeros.
* Girar $10.000 desde el primer cajero creado, $30.000 desde el tercer cajero creado y $90.000 desde el último cajero creado. Se debe desplegar cuantos billetes de cada moneda entregó cada cajero desde el cual se hizo un giro.
* Desplegar un listado con los datos de los cajeros automáticos existentes.

El formato del listado debe ser el que se muestra a continuación:

LISTADO DE CAJEROS --- Fecha: dd-mm-aaaa

----------------------------------------------------------

Id, Saldo ($), Estado

XXXXXX, 9999999, xxxxxxxxx

XXXXXX, 9999999, xxxxxxxxx

Note que el título del listado incluye la fecha al ejecutar el programa (fecha del sistema).

**Importante**:

* Debe usar las constantes indicadas y no los valores que representan. El no cumplir con esto implicará un descuento de puntaje en cada instancia que corresponda usar.
* Los datos leídos son siempre válidos, por lo que no necesita hacer validaciones, salvo lo que se haya indicado expresamente en el enunciado.
* Las copias desde cualquier fuente serán evaluadas con nota 1.0. Lo mismo aplica a quienes faciliten la copia.

**Constructores ArrayList**

|  |  |
| --- | --- |
| **Constructor** | **Descripción** |
| [**ArrayList**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#%3Cinit%3E())() | Construye una lista vacía con una capacidad inicial de diez. |
| [**ArrayList**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#%3Cinit%3E(int))(int initialCapacity) | Construye una lista vacía con la capacidad inicial especificada. |

**Métodos ArrayList**

|  |  |
| --- | --- |
| **Modificador y Tipo + Método** | **Descripción** |
| void [**add**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#add(int,E))(int index, [**E**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html) element) | Inserta el elemento especificado en la posición especificada en esta lista. |
| boolean [**add**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#add(E))([**E**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html) e) | Agrega el elemento especificado al final de esta lista. |
| void [**clear**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#clear())() | Quita todos los elementos de esta lista. |
| boolean [**contains**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#contains(java.lang.Object))(**[Object](file:///D:\\msoto-Software\\Instaladores%20sw\\JSE\\jdk-16.0.1_doc-all\\api\\java.base\\java\\lang\\Object.html" \o "class in java.lang)** o) | Devuelve true si esta lista contiene el elemento especificado. |
| void [**E**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html) [**get**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#get(int))(int index) | Devuelve el elemento en la posición especificada en esta lista. |
| int [**indexOf**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#indexOf(java.lang.Object))(**[Object](file:///D:\\msoto-Software\\Instaladores%20sw\\JSE\\jdk-16.0.1_doc-all\\api\\java.base\\java\\lang\\Object.html" \o "class in java.lang)** o) | Devuelve el índice de la primera aparición del elemento especificado en esta lista, o -1 si esta lista no contiene el elemento. |
| boolean [**isEmpty**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#isEmpty())() | Devuelve true si esta lista no contiene ningún elemento. |
| [**E**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html) [**remove**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#remove(int))(int index) | Quita el elemento en la posición especificada en esta lista. |
| boolean [**remove**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#remove(java.lang.Object))(**[Object](file:///D:\\msoto-Software\\Instaladores%20sw\\JSE\\jdk-16.0.1_doc-all\\api\\java.base\\java\\lang\\Object.html" \o "class in java.lang)** o) | Quita la primera aparición del elemento especificado de esta lista, si está presente. |
| int [**size**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#size())() | Devuelve el número de elementos de esta lista. |
| [**Object**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\lang\Object.html)[] [**toArray**](file:///D:\msoto-Software\Instaladores%20sw\JSE\jdk-16.0.1_doc-all\api\java.base\java\util\ArrayList.html#toArray())() | Devuelve un arreglo que contiene todos los elementos de esta lista (cada uno de tipo Object) en la secuencia correcta (del primer al último elemento). |

**Métodos LocalDate**

|  |  |
| --- | --- |
| **Modificador y tipo + Método** | **Descripción** |
| int [**compareTo**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#compareTo(java.time.chrono.ChronoLocalDate))(**[ChronoLocalDate](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\time\\chrono\\ChronoLocalDate.html" \o "interface in java.time.chrono)** otro) | Compara esta fecha con otra fecha. |
| boolean [**equals**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#equals(java.lang.Object))([**Object**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\lang\Object.html) objeto) | Comprueba si esta fecha es igual a otra fecha. |
| [**String**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\lang\String.html) [**format**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#format(java.time.format.DateTimeFormatter))(**[DateTimeFormatter](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\time\\format\\DateTimeFormatter.html" \o "class in java.time.format)** formateador) | Da formato a esta fecha con el formateador indicado. |
| int [**getDayOfMonth**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#getDayOfMonth())() | Obtiene el día del mes. |
| int [**getDayOfYear**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#getDayOfYear())() | Obtiene el día del año. |
| [**Month**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\Month.html) [**getMonth**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#getMonth())() | Obtiene el mes del año desde la enumeración Month. |
| int [**getMonthValue**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#getMonthValue())() | Obtiene el mes del año del 1 al 12. |
| int [**getYear**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#getYear())() | Obtiene el año. |
| boolean [**isAfter**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#isAfter(java.time.chrono.ChronoLocalDate))(**[ChronoLocalDate](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\time\\chrono\\ChronoLocalDate.html" \o "interface in java.time.chrono)** otro) | Comprueba si esta fecha es posterior a la fecha especificada. |
| boolean [**isBefore**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#isBefore(java.time.chrono.ChronoLocalDate))(**[ChronoLocalDate](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\time\\chrono\\ChronoLocalDate.html" \o "interface in java.time.chrono)** otro) | Comprueba si esta fecha es anterior a la fecha especificada. |
| boolean [**isEqual**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#isEqual(java.time.chrono.ChronoLocalDate))(**[ChronoLocalDate](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\time\\chrono\\ChronoLocalDate.html" \o "interface in java.time.chrono)** otro) | Comprueba si esta fecha es igual a la fecha especificada. |
| static [**LocalDate**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html) [**now**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#now())() | Obtiene la fecha actual del reloj del sistema en la zona horaria predeterminada. |
| static [**LocalDate**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html) [**now**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#now(java.time.Clock))([**Clock**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\Clock.html) clock) | Obtiene la fecha actual del reloj especificado. |
| static [**LocalDate**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html) [**of**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#of(int,int,int))(int agno, int mes, int dia) | Obtiene una instancia de LocalDate de un año, un mes y un día. |
| static [**LocalDate**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html) [**parse**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#parse(java.lang.CharSequence))(**[CharSequence](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\lang\\CharSequence.html" \o "interface in java.lang)** texto) | Obtiene una instancia de LocalDate a partir de una cadena de texto como 2007-12-03. |
| static [**LocalDate**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html) [**parse**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#parse(java.lang.CharSequence,java.time.format.DateTimeFormatter))(**[CharSequence](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\lang\\CharSequence.html" \o "interface in java.lang)** texot, [**DateTimeFormatter**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\format\DateTimeFormatter.html) fomateador) | Obtiene una instancia de LocalDate a partir de una cadena de texto mediante un formateador específico. |
| [**String**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\lang\String.html) [**toString**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#toString())() | Genera esta fecha como una cadena, p.e., 2007-12-03. |
| [**Period**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\Period.html) [**until**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#until(java.time.chrono.ChronoLocalDate))(**[ChronoLocalDate](file:///C:\\msoto\\msoto-Software\\docs_jdk-22.0.2\\api\\java.base\\java\\time\\chrono\\ChronoLocalDate.html" \o "interface in java.time.chrono)** fechaFinalExluyente) | Calcula el período entre esta fecha y la fecha final (se incluye hasta el día anterior) como un Período. |
| long [**until**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#until(java.time.temporal.Temporal,java.time.temporal.TemporalUnit))([**Temporal**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\temporal\Temporal.html) fechaFinalExcluyente, [**TemporalUnit**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\temporal\TemporalUnit.html) unidad) | Calcula la cantidad de tiempo hasta la fecha final (se incluye hasta el día anterior) en términos de la unidad especificada (ChronoUnit.DAYS, ChronoUnit.MONTHS, ChronoUnit.YEARS). |

**Método DateTimeFormatter**

|  |  |
| --- | --- |
| **Modificador y tipo + Método** | **Descripción** |
| static **DateTimeFormatter** [**ofPattern**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\LocalDate.html#compareTo(java.time.chrono.ChronoLocalDate))([**String**](file:///C:\msoto\msoto-Software\docs_jdk-22.0.2\api\java.base\java\time\chrono\ChronoLocalDate.html) patron) | Crea un formateador utilizando el patrón especificado. P.e., el patrón podría ser “dd/MM/yyyy”. |