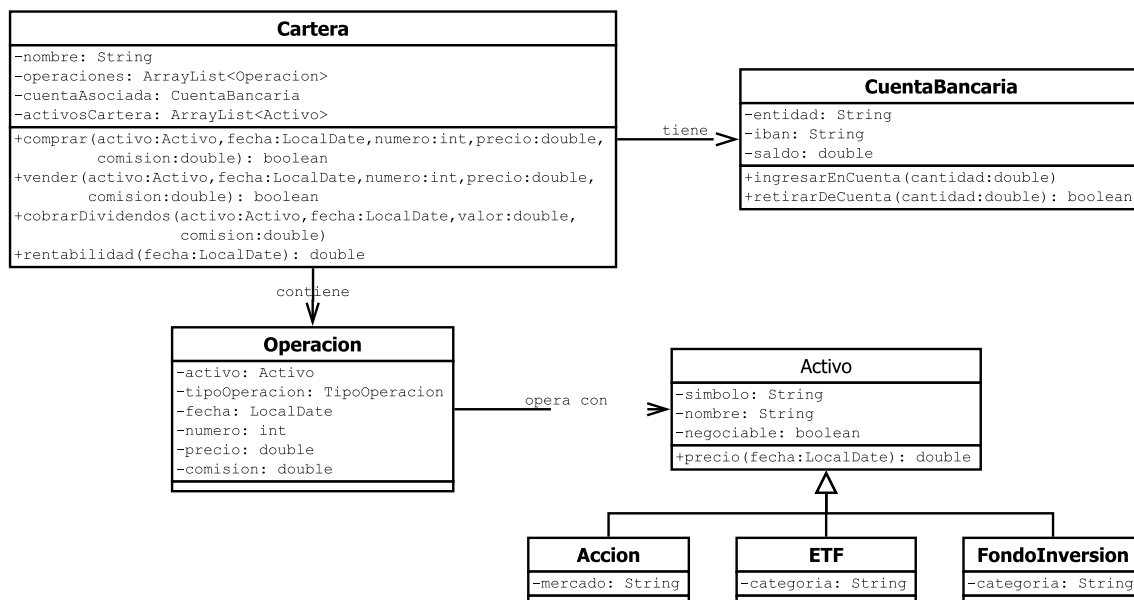


Práctica 2. Gestión de inversiones.

Programación II. Grado en Ingeniería Informática.

12 de abril de 2023

El objetivo de esta práctica es realizar un programa de ayuda a la gestión de las inversiones en activos que podemos realizar y llevar un seguimiento de ellas. Para ello hemos contratado una empresa de análisis y nos ha mandado el siguiente diagrama de clases. En esta práctica vamos a implementar en Java el programa utilizando este diagrama de clases.



Vamos a empezar desarrollando el elemento que describe qué es un **Activo**. Un activo es cualquier inversión que vamos a realizar. En este sentido se trata de una *clase abstracta* de la que extenderemos los distintos tipos de activos. Los activos que vamos a crear con Acciones de empresa, Fondos cotizados (ETF) y Fondos de inversión. Un activo tendrá un símbolo que lo identifica, un nombre y si es negociable o no.

Para realizar las inversiones necesitamos una **Cartera** de inversión que tendrá un nombre y una **Cuenta Bancaria** asociada. Las compras se realizarán con el saldo de dicha cuenta y las ventas se ingresarán en la cuenta bancaria.

Cuando se compra o vende un activo se necesita pedir al usuario el símbolo del activo (ejemplos al final del documento), la fecha de la operación, el número de acciones o participaciones, el precio de cada una y el valor de las comisiones pagadas. Aunque las compras puedan hacerse en distintos mercados, para simplificar el programa supondremos que siempre están valoradas en euros. Solo se pueden realizar operaciones con un activo si es negociable.

Cuando se pide al usuario el activo sobre el que se realiza una operación se le pedirá el símbolo, el nombre del activo y el detalle concreto, el mercado si es una acción o la categoría si es un fondo. En el caso de que el usuario ya hubiese realizado alguna operación con dicho activo será suficiente con pedir el símbolo, pues el resto de información ya la tenemos.

Los tipos de operación que se pueden realizar sobre un activo son COMPRA, VENTA, DIVIDENDOS. La compra retira el valor correspondiente de la cuenta bancaria para realizar la compra sumando la comisión. La venta deposita en la cuenta bancaria el valor correspondiente a la venta restando la comisión. Los dividendos depositan en la cuenta bancaria el valor correspondiente a los dividendos restando la comisión.

Cada una de las clases que heredan de **Activo** deben tener:

- Un constructor con la información del activo.
- Métodos para obtener la información de los atributos.
- Método para modificar si es negociable.

Las clases **Operacion** y **CuentaBancaria** tendrán:

- Un constructor con la información necesaria.
- Métodos get y set según considere.

En la clase **Cartera** tendrá:

- Un constructor con el nombre de la cartera y la cuenta bancaria asociada.
- Métodos para la compra y venta de activos, así como para realizar el cobro de dividendos.
- Método para calcular la rentabilidad de las operaciones hasta una fecha dada.
- Métodos auxiliares que necesite.

Para el cálculo de la rentabilidad utilice el siguiente esquema.

$$\left(\frac{\text{Valor final} + \text{Valor de las ventas} + \text{Valor de los dividendos}}{\text{Valor de las compras}} - 1 \right) \%$$

Para obtener el valor de los activos en una determinada fecha se le proporciona una Clase de nombre **Precio** que dado un símbolo de un activo y una fecha devuelve el valor del activo en dicha fecha. El valor del precio se busca en la web finance.yahoo.com.

En el programa principal de nombre **GestionInversiones** se debe crear, en primer lugar una Cartera con los siguientes datos: nombre del cliente y cuenta bancaria (nombre de la entidad, IBAN y saldo inicial). Además, se implementará un menú con las siguientes opciones:

- 1- Registrar un Activo. Para ello se preguntará al usuario qué tipo de Activo quiere registrar: Acciones, ETFs o Fondos de inversión. También habrá que indicar el símbolo, el nombre y el mercado o categoría dependiendo del tipo escogido.
- 2- Ingresar dinero en cuenta. Solicitará al usuario la cantidad a ingresar y la ingresará en la cuenta.
- 3- Retirar dinero de la cuenta. Solicitará al usuario la cantidad a retirar y hará la operación solamente si existe saldo suficiente. En caso contrario no hará nada.
- 4- Obtener información de la cuenta y su saldo.
- 5- Realizar la compra de un activo. Debe solicitar al usuario el símbolo del activo (si no hay otras compras anteriores de ese activo debe solicitar también su nombre y el mercado o categoría, dependiendo del activo), el número de acciones o participaciones, el precio unitario, y la comisión de la operación.
- 6- Realizar una venta. Debe solicitar al usuario el símbolo del activo. Si no existen compras anteriores o ya no se tiene el activo en cartera, debe indicar que no se pueden realizar ventas de los mimos. En caso contrario debe solicitar la cantidad de acciones o participaciones a vender. Si no se dispone de suficientes debe indicarlo y cancelar la

Práctica 2. Gestión de inversiones.

Programación II. Grado en Ingeniería Informática.

12 de abril de 2023

- operación. Si se pueden vender debe solicitar el precio unitario y la comisión de la operación.
- 7- Cobrar dividendos. Debe solicitar al usuario el símbolo del activo. Si no se dispone de dicho activo en cartera debe indicar que no aplica la operación y cancelarla. Si se dispone de activos de dicho símbolo debe solicitar los dividendos recibidos y la comisión de la operación.
 - 8- Informe de las operaciones realizadas. Genera una lista de todas las operaciones realizadas.
 - 9- Informe de operaciones entre fechas. Solicita al usuario una fecha inicial y una fecha final y genera una lista de todas las operaciones realizadas entre ambas fechas inclusive.
 - 10- Calcular rentabilidad. Genera un informe con el cálculo de la rentabilidad de todas las operaciones realizadas hasta la fecha actual. El formato ha de ser el especificado en la siguiente opción.
 - 11- Calcular rentabilidad hasta una fecha. Solicita al usuario la fecha de cálculo y genera un informe con el cálculo de la rentabilidad de todas las operaciones realizadas hasta la fecha indicada.
 - 12- Finalizar. Termina el programa.

NOTA: Será necesario crear métodos auxiliares para realizar el programa.

Para ejecutar el programa se proporciona un archivo que contiene la entrada de prueba del programa. El archivo se puede cargar en IntelliJ Idea. De este modo, no será necesario introducir los datos manualmente cada vez que queráis probar el programa. Además, en la plantilla se dan de alta todos los Activos de ejemplo proporcionados y se prueban todas las funcionalidades del programa.

La entrega se realizará a través de Canvas en un único archivo comprimido (.zip) que contenga todos los archivos necesarios para la evaluación de la práctica. **La fecha límite de entrega es el día 7 de mayo a las 23:59h.** Solo es necesario hacer una entrega por cada grupo de trabajo de dos personas, sin importar cuál de los dos alumnos realice la entrega.

Para la evaluación de la práctica se tendrá en cuenta no solo el correcto funcionamiento del programa, sino que también influirán de manera sustancial en la calificación los siguientes aspectos:

- Definiciones de todas las clases, atributos, métodos y variables que los alumnos consideren oportunos para mejorar el programa en cuanto a eficiencia, simplicidad y legibilidad del código.
- **Documentación de código correcta en todos los archivos de código entregados.**
- Código legible y ordenado, siguiendo las recomendaciones de estilo comentadas en clase y detalladas en las presentaciones de los temas.
- Nombres de clases, atributos, métodos, variables y constantes adecuados.
- Cualquier entrega que no compile y ejecute será calificada automáticamente con un 0.
- Se valorará positivamente que el código sea robusto frente a errores típicos como la entrada de datos incorrecta del usuario, diseño del flujo de ejecución adecuado, etc.

Práctica 2. Gestión de inversiones.

Programación II. Grado en Ingeniería Informática.

12 de abril de 2023

Acciones

| Símbolo | Nombre | Mercado |
|---------|-----------------|---------|
| SAN | Banco Santander | BME |
| TSLA | Tesla, Inc. | NASDAQ |
| ACCN | Accenture | NYSE |

ETF

| Símbolo | Nombre | Categoría |
|----------|--|-------------------------------|
| CSNDX.SW | iShares NASDAQ 100 UCITS ETF | RV USA Cap. Grande Growth |
| TNOW.PA | Lyxor MSCI World Information Technology TR UCITS ETF - C-EUR | RV Sector Tecnología |
| F701.DE | Lyxor Portfolio Strategy UCITS ETF | Mixtos Moderados EUR – Global |

Fondos de inversión

| Símbolo | Nombre | Categoría |
|--------------|--|--------------------------------------|
| OP0001C2ZB.F | Nordea 1 - Low Duration European Covered Bond Fund BI EUR | RF Deuda Corporativa Corto Plazo EUR |
| OP0000Z32X.F | BlackRock Global Funds - European Value Fund I2 EUR | RV Europa Cap. Grande Value |
| OP0000WUYX | Morgan Stanley Investment Funds - Emerging Leaders Equity Fund Z | RV Global Emergente |