

Técnicas Actuales de Programación 2022/2023

Examen Final - Enero

Responda al examen con un proyecto que lleve su nombre (elija Java with Ant -> Java Application).

Además de que el código sea razonablemente correcto y dé los resultados esperados, se valorará que esté limpio de restos de pruebas y de código anulado con comentarios, que esté bien formateado (Nebeans ayuda), y que incluya algún comentario donde se considere necesario. En el GUI se valorará el redimensionamiento razonable.

Una empresa financiera quiere desarrollar un software que analice la información de inversión en torno al NASDAQ100. El NASDAQ100 es un índice bursátil que agrupa a las 100 empresas más importantes del Nasdaq Stock Market. La información que analizará la empresa financiera será el histórico de cotizaciones de empresas del NASDAQ100 durante el último mes. En la siguiente dirección "<https://gtts.ehu.es/German/Docencia/2223/TAP/Exámenes/Nasdaq100>" es una carpeta donde se encuentra toda la información necesaria (un fichero "[index.txt](#)" con el listado de compañías y un fichero para cada compañía con sus cotizaciones -hay 101 porque Google tiene dos tickers [GOOG](#) y [GOOGL](#)-). Todo está codificado en UTF8).

1 – Clases DayData y Asset

Cree la clase **DayData** que almacenará información sobre la cotización de un día de un activo financiero bursátil. Esta consiste en lo siguiente:

open → primer precio del día (precio de apertura)

high → precio más alto registrado en la sesión

low → precio más bajo registrado en la sesión

close → último precio del día (precio de cierre)

volume → número de activos

comprados/vendidos durante la jornada



Cree la clase **Asset** que representa la cotización del último mes de un activo financiero bursátil. La clase deberá contener al menos la siguiente información (que se encuentra en los ficheros descargados):

- El Ticker del activo (un identificador de texto como "GOOG", "AAPL", "MSFT" ...)
- Descripción (un texto: "Alphabet Inc", "Apple Inc", "Microsoft Corp" ...)
- La secuencia de cotizaciones (DayData) del último mes

Y contará además con los siguientes métodos:

```
//Constructor que inicializará el objeto a partir de un fichero de texto como los descargados.  
public Asset(String fileName, String encoding, String ticker, String descr)
```

```
public double getAvgClose() // Precio de cierre promedio
```

```
public double getAvgCloseDiff() //Promedio del cambio entre precios de cierre consecutivos
```

```
//Desviación estándar del cambio entre precios de cierre consecutivos
(volatilidad):
public double getVolatility()
```

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

```
//Correlación entre los precios de cierre de dos Assets:
public static double correlation(Asset a1, Asset a2)
```

$$\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

Se añadirá también un método “main” para comprobar el funcionamiento de la clase. Debe mostrar por pantalla la correlación entre las 4 empresas más importantes del NASDAQ100 (MSFT, AAPL, AMZN y GOOG)

	AAPL	AMZN	GOOG
MSFT	0,43	0,32	0,44
AAPL		0,87	0,84
AMZN			0,84

2 – Clase AssetDB

Cree una clase de **AssetDB** que almacene la información de un conjunto de activos financieros que pueda ser cargado a partir de un directorio cuya estructura coincida con el archivo descargado. La clase contendrá los siguientes métodos:

```
//Constructor. Inicializa la BD a partir de un directorio.
public AssetDB(String dirName)
```

```
//Devuelve el Asset correspondiente a un ticker.
public Asset get(String ticker)
```

```
//Devuelve la lista de Assets cuyo ticker o descripción contengan la cadena de caracteres “s” (case-insensitive) .
public List<Asset> search(String s)
```

```
/*Una manera de minimizar el riesgo de las inversiones es seleccionar un conjunto de activos que no
* estén correlacionados entre sí. Esta función devuelve la lista de Assets de tamaño n (ordenada de
* menor a mayor) con correlación absoluta mínima frente al Asset indicado (ticker). */
public List<Asset> lessCorrelated(String ticker, int n)
```

Se añadirá también un método “main” para comprobar el funcionamiento de la clase. Debe mostrar por pantalla las 10 empresas menos correlacionadas con la empresa más importante del NASDAQ100 (MSFT):

```
1: [+]0,0010 ZM (Zoom Video Communications Inc)
2: [-]0,0115 XEL (Xcel Energy Inc)
3: [+]0,0196 ALGN (Align Technology Inc)
4: [+]0,0206 CSCO (Cisco Systems Inc)
5: [+]0,0214 CSX (CSX Corp)
6: [-]0,0246 EBAY (eBay Inc)
7: [+]0,0291 AMAT (Applied Materials Inc)
8: [+]0,0306 IDXX (IDEXX Laboratories Inc)
9: [-]0,0343 LULU (Lululemon Athletica Inc)
10: [+]0,0349 MCHP (Microchip Technology Inc)
1: [+]0,0010 ZM (Zoom Video Communications Inc)
```

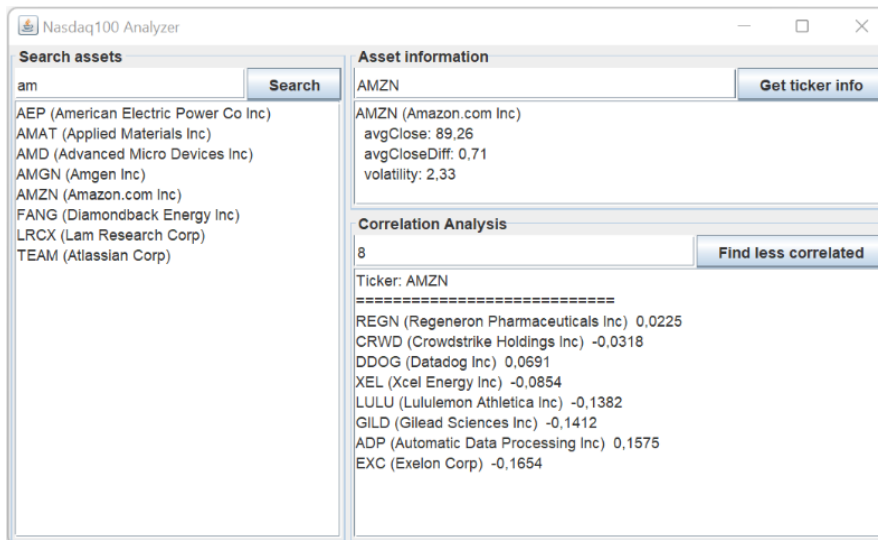
3 – Aplicación gráfica NASDAQ100

La aplicación construirá un objeto AssetDB del apartado anterior a partir de los ficheros descargados y facilitará el análisis financiero de la información. Su funcionalidad consistirá en poder seleccionar/localizar un asset entre los 101 para obtener sus valores de media de cierre, media de los diferenciales de cierre y volatilidad, así como una lista de las N empresas menos correlacionadas con ella.

Para ello pueden desarrollarse diferentes GUI como los que se muestran a continuación.

Ejemplo 1: La aplicación gráfica constará de tres zonas/secciones que inicialmente aparecen en blanco.

1. Al pulsar “search”, mostrará la lista de empresas que contengan el término de búsqueda en su ticker o descripción (case-insensitive).
2. Al pulsar “Get ticker info” mostrará la información referente al ticker consignado en el campo de texto.
3. Al pulsar “Find less correlated” Mostrará la lista de n empresas menos correlacionadas con la empresa indicada por el ticker de la sección anterior, donde n es el número consignado en el campo de texto.



Ejemplo 2: La aplicación gráfica consta de tres zonas, donde la de la izquierda incluye un área de texto que aparece inicialmente con la lista de los 101 assets, y las otras dos áreas de texto estarán vacías

Al escribir en el campo de texto, con cada nueva letra la lista de empresas se va restringiendo a sólo aquellas que contengan la secuencia de texto. Cuando la lista muestre una única empresa, las otras dos áreas de texto mostrarán la información referida a ella.

Si se altera el valor del campo selector del número de empresas menos correladas, la representación se adapta a su valor.

