

Interfaces Between System Components

The stock monitoring system uses several key interfaces to let different parts of the system communicate to each other. These interfaces make sure that each part can work on its own while still sharing data smoothly.

IStockMonitoring

The IStockMonitoring interface is the main part that watches stock prices. It's used by classes like StockPriceMonitor and MonitoringRing. The main method, `monitorStockPrices()`, starts the process of tracking stock prices. In the StockMonitorApplication class, this interface helps the system update and observe stocks at the same time. The StockPriceMonitor class shows how the system keeps track of prices in real-time and logs any changes for debugging.

IStockPrices

The IStockPrices interface, used by AbstractStockPrices and classes like StockPriceTracker, handles everything to do with stock prices. It has two important methods: `updateStockPrice(double price)` to update current prices and `getCurrentPrice()` to get the latest price. The StockPriceTracker class shows how these methods keep and access price information for specific stocks.

IDataBase

The IDataBase interface, used by classes like StockDatabase, takes care of saving and getting stock data. It has two main methods: `saveData(String symbol, double price)` to store price information and `fetchData(String symbol)` to get it back. The StockDatabase class shows how to use SQLite to store data safely and handle any errors that come up.

IChartDisplay

The IChartDisplay interface, used by classes like StockChartDisplay and DisplayCharts, handles showing stock data in charts. The `displayChart()` method makes sure the charts are up-to-date. The StockChartDisplay class shows how the system can show different types of charts, with the type of chart set when the class is created using the AbstractChartDisplay constructor.

Other Interfaces

There are also interfaces like `IDataCollector` for getting market data and `ILoggingData` for logging system activity. These are used a lot in the `StockMonitorApplication` class, which is the main part that brings all these interfaces together. This class shows how to set up and manage all the different parts of the system.

How the Interfaces Work Together

All these interfaces work together to make the system run smoothly. For example, when a stock price changes, the system uses `IStockPrices` to update the price, `IDataBase` to save the new price, `IChartDisplay` to update the chart, and `ILoggingData` to log the change. This is all shown in the `updateStock()` method of the `StockMonitorApplication` class.

System Design

The system is designed using clean architecture principles, which means each part has a clear job and can be changed without affecting the rest of the system. Each interface has an abstract class (like `AbstractStockPrices`, `AbstractChartDisplay`, `AbstractDatabase`) that does the basic work, and the specific details are handled by concrete classes. This makes the system easy to maintain and update.

1. StockMonitoring

- a. **monitorStockPrices()**: Starts tracking stock prices.
- b. Used by: `StockPriceMonitor`, `MonitoringRing`.
- c. Started by: `StockMonitorApplication`.

2. IStockPrices

- a. **updateStockPrice(double price)**: Updates the current price.
- b. **getCurrentPrice()**: Gets the latest price.
- c. Used by: `AbstractStockPrices`, `StockPriceTracker`.

3. IDataBase

- a. **saveData(String symbol, double price)**: Saves stock price data.
- b. **fetchData(String symbol)**: Retrieves saved data.
- c. Used by: `StockDatabase`.

4. IChartDisplay

- a. **displayChart()**: Shows stock data in charts.
- b. Used by: `StockChartDisplay`, `DisplayCharts`.

5. **IDataCollector**

- a. **collectData()**: Gets market data from outside sources.
- b. Used by: DataCollector.

6. **ILoggingData**

- a. **log(String message)**: Records system events.
- b. Used by: StockLogger.

7. **IChartMonitor**

- a. **monitorCharts()**: Keeps an eye on chart updates.
- b. Used by: AbstractChartMonitor.

8. **IChartStocks**

- a. **updateStockChart()**: Updates charts with new data.
- b. Used by: AbstractChartStocks.

How They Work Together

- **StockMonitorApplication** uses these interfaces to manage the whole process.
- Price updates go from **IDataCollector** to **IStockPrices**.
- Data is saved using **IDataBase**.
- Charts are updated via **IChartDisplay**.
- Everything is logged with **ILoggingData**.
- Monitoring is handled by **IStockMonitoring**.