

Chapter 18: Objects and Classes - Exercises

Exercise 18 - 01: The Fan class

Design a class named `Fan` to represent a fan. The class contains:

- Three constants named `SLOW`, `MEDIUM`, and `FAST` with values `1`, `2`, and `3` to denote the fan speed.
- An `int` data field named `speed` that specifies the speed of the fan (default `SLOW`).
- A `boolean` data field named `on` that specifies whether the fan is on (default `false`).
- A `double` data field named `radius` that specifies the radius of the fan (default `5`).
- A string data field named `color` that specifies the color of the fan (default `blue`).
- A no-arg constructor that creates a default fan.
- The accessor and mutator methods for all four data fields.
- A method named `toString()` that returns a string description for the fan. If the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns fan color and radius along with the string `"fan is off"` in one combined string.

Draw the UML diagram for the class. Implement the class. Write a test program that creates two `Fan` objects. Assign maximum speed, radius `10`, color `yellow`, and turn it on to the first object. Assign medium speed, radius `5`, color `blue`, and turn it off to the second object. Display the objects by invoking their `toString` method.

Exercise 18 - 02: The Stock class

Design a class named `Stock` that contains:

- A string data field named `symbol` for the stock's symbol.
- A string data field named `name` for the stock's name.
- A `double` data field named `previousClosingPrice` that stores the stock price for the previous day.
- A `double` data field named `currentPrice` that stores the stock price for the current time.
- A constructor that creates a stock with specified symbol and name.
- The accessor methods for all data fields.
- The mutator methods for `previousClosingPrice` and `currentPrice`.
- A method named `changePercent()` that returns the percentage changed from `previousClosingPrice` to `currentPrice`.

Draw the UML diagram for the class. Implement the class. Write a test program that creates a `Stock` object with the stock symbol `SUNW`, the name `Sun Microsystems Inc`, and the previous closing price of `100`. Set a new current price to `90` and display the price-change percentage.