

Solutions for UML Class Diagrams

Chapter 9

Exercise 9.2: Stock

Stock	
-symbol: str -name: str -previousClosingPrice: float -currentPrice: float	The symbol of this stock. The name of this stock. The previous closing price of this stock. The current price of this stock.
Stock(symbol: str, name: str) getChangePercent(): float getSymbol(): str getName(): str getPreviousClosingPrice(): float setPreviousClosingPrice(price: float): None getCurrentPrice(): float setCurrentPrice(price: float): None	Constructs a stock with a specified symbol and a name. Returns the percentage of change of this stock. Returns the symbol of this stock. Returns the name of this stock. Returns the previous closing price of this stock. Sets a new previous closing price of this stock. Returns the current price of this stock. Sets a new current price of this stock.

Exercise 9.4: Fan

Fan	
<u>SLOW = 1</u> <u>MEDIUM = 2</u> <u>FAST = 3</u> -speed: int -on: bool -radius: float -color: str	Constant defined outside of the class. Constant defined outside of the class. Constant defined outside of the class. The speed of this fan (default 1). Indicates whether the fan is on (default false). The radius of this fan (default 5). The color of this fan (default white).
Fan() getSpeed(): int setSpeed(speed: int): None isOn(): bool setOn(on: bool): None getRadius(): float setRadius(radius: float): None getColor(): str setColor(color: str): None	Constructs a fan with default values. Returns the speed of this fan. Sets a new speed for this fan. Returns true if this fan is on. Sets this fan on to true or false. Returns the radius of this fan. Sets a new radius for this fan. Returns the color of this fan. Sets a new color for this fan.

Exercise 9.6: QuadraticEquation

QuadraticEquation	get methods for all data fields are provided and omitted for brevity.
-a: float -b: float -c: float	Three coefficients for the equation.
QuadraticEquation(a: float, b: float, c: float) getDiscriminant(): float getRoot1(): float getRoot2(): float	Constructs a QuadraticEquation with the specified coefficients. Returns the discriminant of this equation. Returns the first root of this equation. Returns the second root of this equation.

Exercise 9.8: Stopwatch

StopWatch	get methods for all data fields are provided and omitted for brevity.
-startTime: float -endTime: float	Start time and end time for the watch.
StopWatch() start(): None stop(): None getElapsedTime(): float	Constructs a StopWatch with the specified start and end time. Starts the watch. Stops the watch. Returns the elapsed time.

Exercise 9.10: Time

Time	
-hour: int -minute: int -second: int	The hour for the time. The minute for the time. The second for the time.
Time() getHour(): int getMinute(): int getSecond(): int setTime(elapsTime): void	Constructs Time for the current time. Returns the clock hour for the time. Returns the minute for the time. Returns the second for the time. Sets a new time.

Exercise 9.12: Circle2D

Circle2D
-x: float -y: float -radius: float
Circle2D(x: float, y: float, radius: float) getX(): float getY(): float setX(x: float): void setY(y: float): void getRadius(): float setRadius(radius: float): void getPerimeter(): float getArea(): float contains(x: float, y: float): bool contains(circle: Circle2D): bool __contains__(circle: Circle2D): bool __lt__, __le__, __gt__, __ge__, __ne__, __eq__