Solutions for UML Class Diagrams

Chapter 9

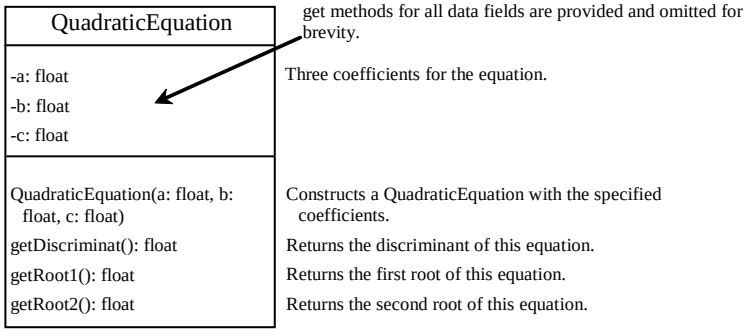
Exercise 9.2: Stock

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

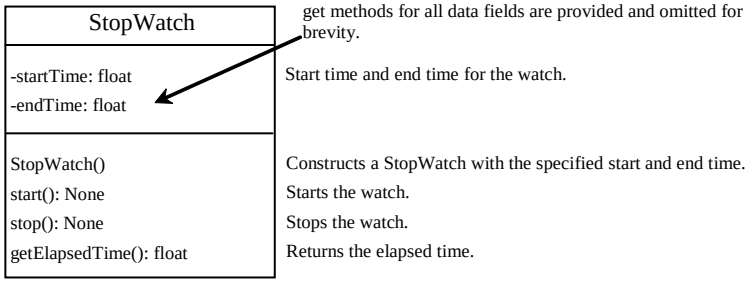
Exercise 9.4: Fan

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

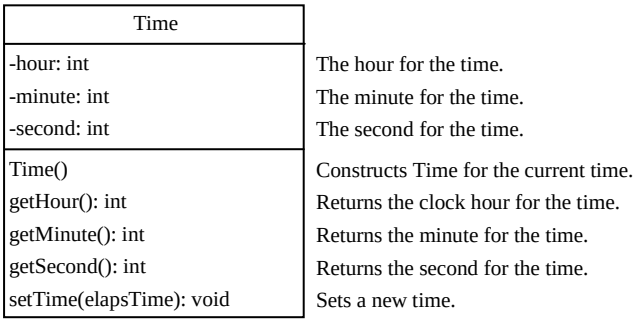
Exercice 9.6: QuadraticEquation



Exercise 9.8: StopWatch



Exercise 9.10: 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__