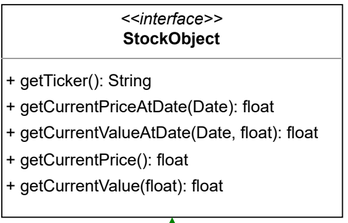
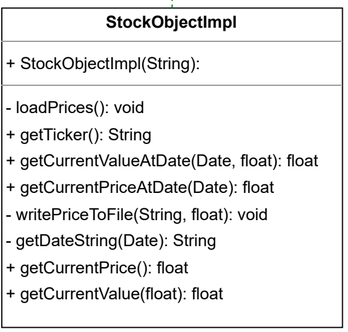
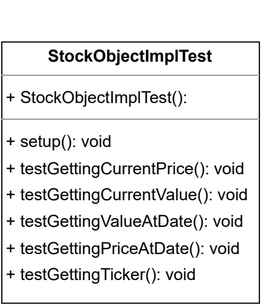
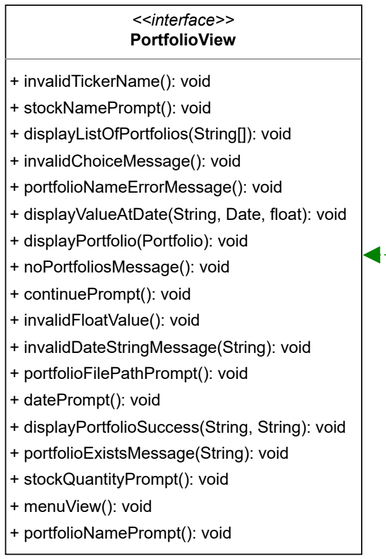
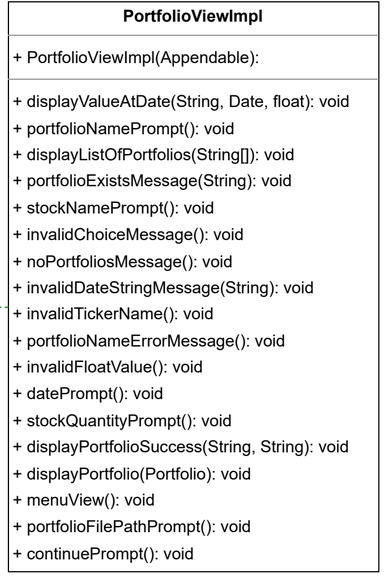
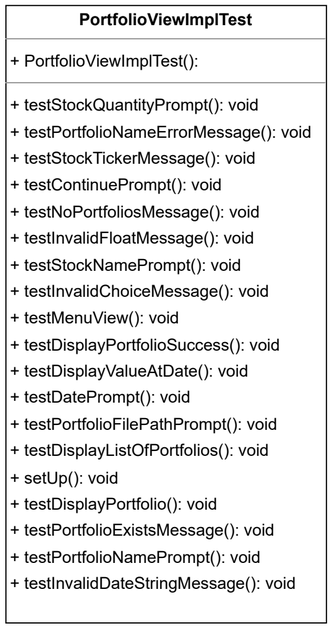

<<interface>>
StockObject
+ getTicker(): String
+ getCurrentPriceAtDate(Date): float
+ getCurrentValueAtDate(Date, float): float
+ getCurrentPrice(): float
+ getCurrentValue(float): float

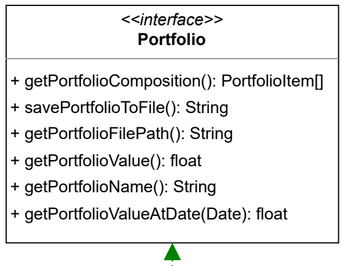

StockObjectImpl
+ StockObjectImpl(String):
- loadPrices(): void
+ getTicker(): String
+ getCurrentValueAtDate(Date, float): float
+ getCurrentPriceAtDate(Date): float
- writePriceToFile(String, float): void
- getDateString(Date): String
+ getCurrentPrice(): float
+ getCurrentValue(float): float

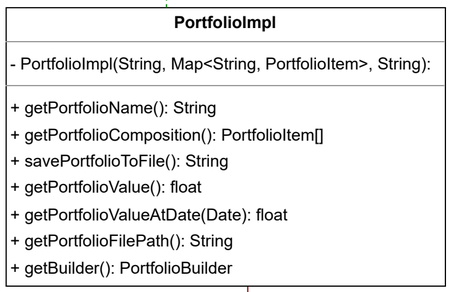

StockObjectImplTest
+ StockObjectImplTest():
+ setup(): void
+ testGettingCurrentPrice(): void
+ testGettingCurrentValue(): void
+ testGettingValueAtDate(): void
+ testGettingPriceAtDate(): void
+ testGettingTicker(): void

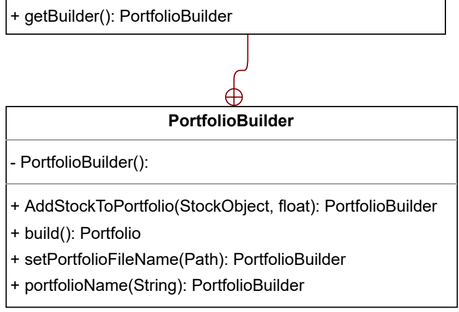

<<interface>>
PortfolioView
+ invalidTickerName(): void
+ stockNamePrompt(): void
+ displayListOfPortfolios(String[]): void
+ invalidChoiceMessage(): void
+ portfolioNameErrorMessage(): void
+ displayValueAtDate(String, Date, float): void
+ displayPortfolio(Portfolio): void
+ noPortfoliosMessage(): void
+ continuePrompt(): void
+ invalidFloatValue(): void
+ invalidDateStringMessage(String): void
+ portfolioFilePathPrompt(): void
+ datePrompt(): void
+ displayPortfolioSuccess(String, String): void
+ portfolioExistsMessage(String): void
+ stockQuantityPrompt(): void
+ menuView(): void
+ portfolioNamePrompt(): void

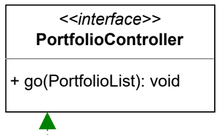

PortfolioViewImpl
+ PortfolioViewImpl(Appendable):
+ displayValueAtDate(String, Date, float): void
+ portfolioNamePrompt(): void
+ displayListOfPortfolios(String[]): void
+ portfolioExistsMessage(String): void
+ stockNamePrompt(): void
+ invalidChoiceMessage(): void
+ noPortfoliosMessage(): void
+ invalidDateStringMessage(String): void
+ invalidTickerName(): void
+ portfolioNameErrorMessage(): void
+ invalidFloatValue(): void
+ datePrompt(): void
+ stockQuantityPrompt(): void
+ displayPortfolioSuccess(String, String): void
+ displayPortfolio(Portfolio): void
+ menuView(): void
+ portfolioFilePathPrompt(): void
+ continuePrompt(): void

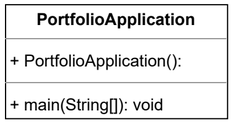

PortfolioViewImplTest
+ PortfolioViewImplTest():
+ testStockQuantityPrompt(): void
+ testPortfolioNameErrorMessage(): void
+ testStockTickerMessage(): void
+ testContinuePrompt(): void
+ testNoPortfoliosMessage(): void
+ testInvalidFloatMessage(): void
+ testStockNamePrompt(): void
+ testInvalidChoiceMessage(): void
+ testMenuView(): void
+ testDisplayPortfolioSuccess(): void
+ testDisplayValueAtDate(): void
+ testDatePrompt(): void
+ testPortfolioFilePathPrompt(): void
+ testDisplayListOfPortfolios(): void
+ setUp(): void
+ testDisplayPortfolio(): void
+ testPortfolioExistsMessage(): void
+ testPortfolioNamePrompt(): void
+ testInvalidDateStringMessage(): void

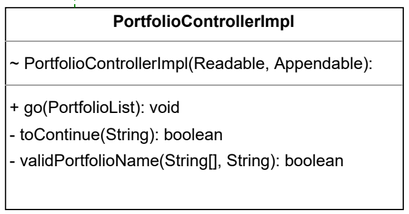

<<interface>>
Portfolio
+ getPortfolioComposition(): PortfolioItem[]
+ savePortfolioToFile(): String
+ getPortfolioFilePath(): String
+ getPortfolioValue(): float
+ getPortfolioName(): String
+ getPortfolioValueAtDate(Date): float

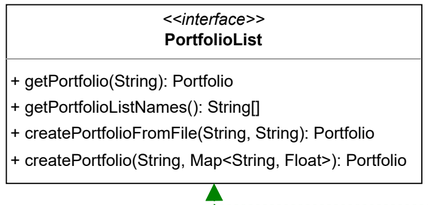

PortfolioImpl
- PortfolioImpl(String, Map<String, PortfolioItem>, String):
+ getPortfolioName(): String
+ getPortfolioComposition(): PortfolioItem[]
+ savePortfolioToFile(): String
+ getPortfolioValue(): float
+ getPortfolioValueAtDate(Date): float
+ getPortfolioFilePath(): String
+ getBuilder(): PortfolioBuilder

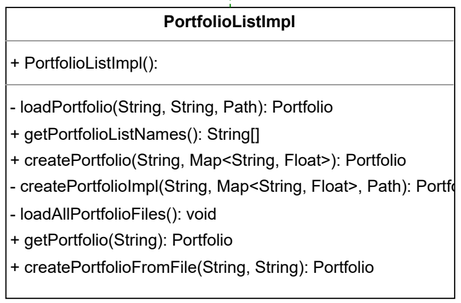

PortfolioBuilder
- PortfolioBuilder():
+ AddStockToPortfolio(StockObject, float): PortfolioBuilder
+ build(): Portfolio
+ setPortfolioFileName(Path): PortfolioBuilder
+ portfolioName(String): PortfolioBuilder

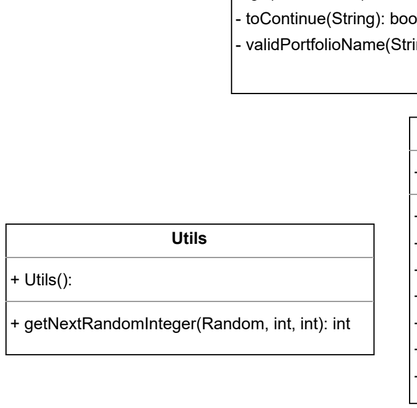

<<interface>>
PortfolioController
+ go(PortfolioList): void

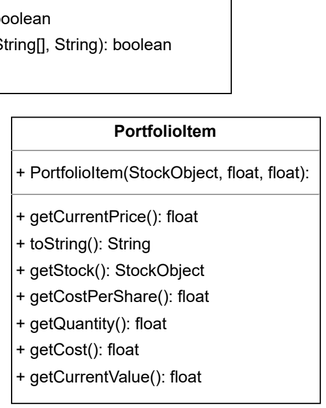

PortfolioApplication
+ PortfolioApplication():
+ main(String[]): void

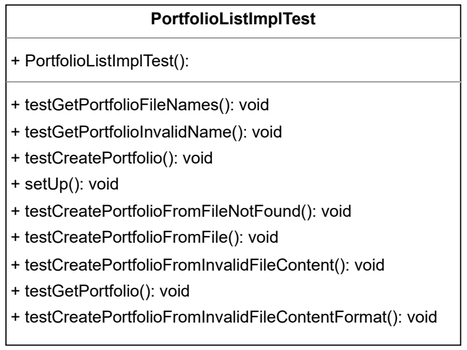

PortfolioControllerImpl
~ PortfolioControllerImpl(Readable, Appendable):
+ go(PortfolioList): void
- toContinue(String): boolean
- validPortfolioName(String[], String): boolean

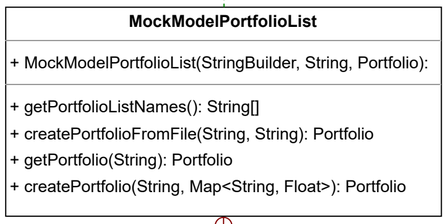

<<interface>>
PortfolioList
+ getPortfolio(String): Portfolio
+ getPortfolioListNames(): String[]
+ createPortfolioFromFile(String, String): Portfolio
+ createPortfolio(String, Map<String, Float>): Portfolio

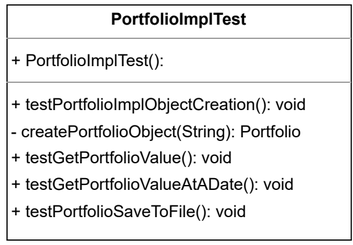

PortfolioListImpl
+ PortfolioListImpl():
- loadPortfolio(String, String, Path): Portfolio
+ getPortfolioListNames(): String[]
+ createPortfolio(String, Map<String, Float>): Portfolio
- createPortfolioImpl(String, Map<String, Float>, Path): Portf
- loadAllPortfolioFiles(): void
+ getPortfolio(String): Portfolio
+ createPortfolioFromFile(String, String): Portfolio

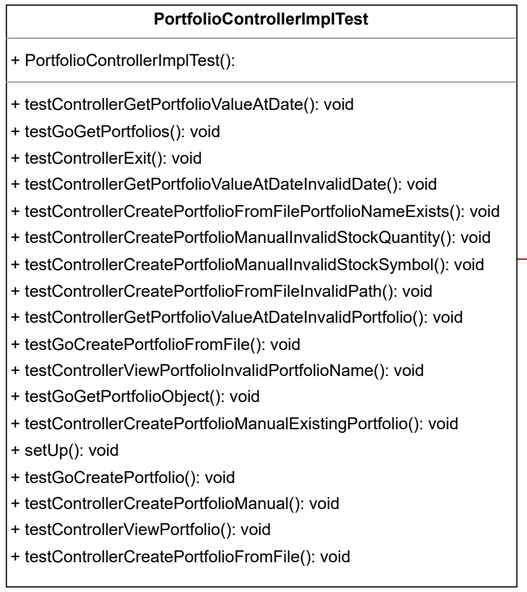

Utils
+ Utils():
+ getNextRandomInteger(Random, int, int): int


PortfolioItem
+ PortfolioItem(StockObject, float, float):
+ getCurrentPrice(): float
+ toString(): String
+ getStock(): StockObject
+ getCostPerShare(): float
+ getQuantity(): float
+ getCost(): float
+ getCurrentValue(): float


PortfolioListImplTest
+ PortfolioListImplTest():
+ testGetPortfolioFileNames(): void
+ testGetPortfolioInvalidName(): void
+ testCreatePortfolio(): void
+ setUp(): void
+ testCreatePortfolioFromFileNotFound(): void
+ testCreatePortfolioFromFile(): void
+ testCreatePortfolioFromInvalidFileContent(): void
+ testGetPortfolio(): void
+ testCreatePortfolioFromInvalidFileContentFormat(): void


MockModelPortfolioList
+ MockModelPortfolioList(StringBuilder, String, Portfolio):
+ getPortfolioListNames(): String[]
+ createPortfolioFromFile(String, String): Portfolio
+ getPortfolio(String): Portfolio
+ createPortfolio(String, Map<String, Float>): Portfolio


PortfolioImplTest
+ PortfolioImplTest():
+ testPortfolioImplObjectCreation(): void
- createPortfolioObject(String): Portfolio
+ testGetPortfolioValue(): void
+ testGetPortfolioValueAtADate(): void
+ testPortfolioSaveToFile(): void


PortfolioControllerImplTest
+ PortfolioControllerImplTest():
+ testControllerGetPortfolioValueAtDate(): void
+ testGoGetPortfolios(): void
+ testControllerExit(): void
+ testControllerGetPortfolioValueAtDateInvalidDate(): void
+ testControllerCreatePortfolioFromFilePortfolioNameExists(): void
+ testControllerCreatePortfolioManualInvalidStockQuantity(): void
+ testControllerCreatePortfolioManualInvalidStockSymbol(): void
+ testControllerCreatePortfolioFromFileInvalidPath(): void
+ testControllerGetPortfolioValueAtDateInvalidPortfolio(): void
+ testGoCreatePortfolioFromFile(): void
+ testControllerViewPortfolioInvalidPortfolioName(): void
+ testGoGetPortfolioObject(): void
+ testControllerCreatePortfolioManualExistingPortfolio(): void
+ setUp(): void
+ testGoCreatePortfolio(): void
+ testControllerCreatePortfolioManual(): void
+ testControllerViewPortfolio(): void
+ testControllerCreatePortfolioFromFile(): void