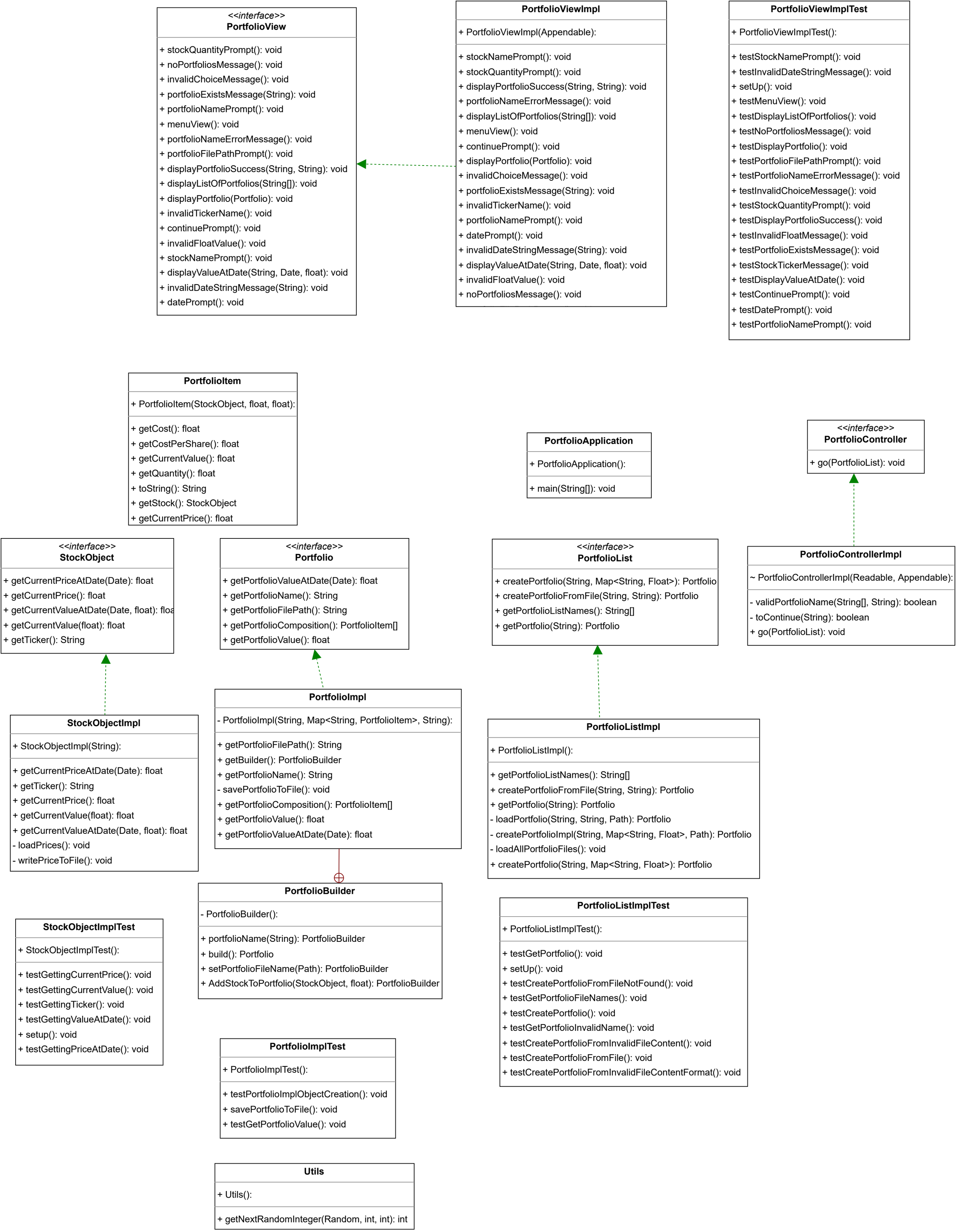
<<interface>>
PortfolioView
+ stockQuantityPrompt(): void
+ noPortfoliosMessage(): void
+ invalidChoiceMessage(): void
+ portfolioExistsMessage(String): void
+ portfolioNamePrompt(): void
+ menuView(): void
+ portfolioNameErrorMessage(): void
+ portfolioFilePathPrompt(): void
+ displayPortfolioSuccess(String, String): void
+ displayListOfPortfolios(String[]): void
+ displayPortfolio(Portfolio): void
+ invalidTickerName(): void
+ continuePrompt(): void
+ invalidFloatValue(): void
+ stockNamePrompt(): void
+ displayValueAtDate(String, Date, float): void
+ invalidDateStringMessage(String): void
+ datePrompt(): void
PortfolioViewImpl
+ PortfolioViewImpl(Appendable):
+ stockNamePrompt(): void
+ stockQuantityPrompt(): void
+ displayPortfolioSuccess(String, String): void
+ portfolioNameErrorMessage(): void
+ displayListOfPortfolios(String[]): void
+ menuView(): void
+ continuePrompt(): void
+ displayPortfolio(Portfolio): void
+ invalidChoiceMessage(): void
+ portfolioExistsMessage(String): void
+ invalidTickerName(): void
+ portfolioNamePrompt(): void
+ datePrompt(): void
+ invalidDateStringMessage(String): void
+ displayValueAtDate(String, Date, float): void
+ invalidFloatValue(): void
+ noPortfoliosMessage(): void
PortfolioViewImplTest
+ PortfolioViewImplTest():
+ testStockNamePrompt(): void
+ testInvalidDateStringMessage(): void
+ setUp(): void
+ testMenuView(): void
+ testDisplayListOfPortfolios(): void
+ testNoPortfoliosMessage(): void
+ testDisplayPortfolio(): void
+ testPortfolioFilePathPrompt(): void
+ testPortfolioNameErrorMessage(): void
+ testInvalidChoiceMessage(): void
+ testStockQuantityPrompt(): void
+ testDisplayPortfolioSuccess(): void
+ testInvalidFloatMessage(): void
+ testPortfolioExistsMessage(): void
+ testStockTickerMessage(): void
+ testDisplayValueAtDate(): void
+ testContinuePrompt(): void
+ testDatePrompt(): void
+ testPortfolioNamePrompt(): void
PortfolioItem
+ PortfolioItem(StockObject, float, float):
+ getCost(): float
+ getCostPerShare(): float
+ getCurrentValue(): float
+ getQuantity(): float
+ toString(): String
+ getStock(): StockObject
+ getCurrentPrice(): float
PortfolioApplication
+ PortfolioApplication():
+ main(String[]): void
<<interface>>
PortfolioController
+ go(PortfolioList): void
<<interface>>
StockObject
+ getCurrentPriceAtDate(Date): float
+ getCurrentPrice(): float
+ getCurrentValueAtDate(Date, float): floa
+ getCurrentValue(float): float
+ getTicker(): String
<<interface>>
Portfolio
+ getPortfolioValueAtDate(Date): float
+ getPortfolioName(): String
+ getPortfolioFilePath(): String
+ getPortfolioComposition(): PortfolioItem[]
+ getPortfolioValue(): float
<<interface>>
PortfolioList
+ createPortfolio(String, Map<String, Float>): Portfolio
+ createPortfolioFromFile(String, String): Portfolio
+ getPortfolioListNames(): String[]
+ getPortfolio(String): Portfolio
PortfolioControllerImpl
~ PortfolioControllerImpl(Readable, Appendable):
- validPortfolioName(String[], String): boolean
- toContinue(String): boolean
+ go(PortfolioList): void
StockObjectImpl
+ StockObjectImpl(String):
+ getCurrentPriceAtDate(Date): float
+ getTicker(): String
+ getCurrentPrice(): float
+ getCurrentValue(float): float
+ getCurrentValueAtDate(Date, float): float
- loadPrices(): void
- writePriceToFile(): void
PortfolioImpl
- PortfolioImpl(String, Map<String, PortfolioItem>, String):
+ getPortfolioFilePath(): String
+ getBuilder(): PortfolioBuilder
+ getPortfolioName(): String
- savePortfolioToFile(): void
+ getPortfolioComposition(): PortfolioItem[]
+ getPortfolioValue(): float
+ getPortfolioValueAtDate(Date): float
PortfolioListImpl
+ PortfolioListImpl():
+ getPortfolioListNames(): String[]
+ createPortfolioFromFile(String, String): Portfolio
+ getPortfolio(String): Portfolio
- loadPortfolio(String, String, Path): Portfolio
- createPortfolioImpl(String, Map<String, Float>, Path): Portfolio
- loadAllPortfolioFiles(): void
+ createPortfolio(String, Map<String, Float>): Portfolio
PortfolioBuilder
- PortfolioBuilder():
+ portfolioName(String): PortfolioBuilder
+ build(): Portfolio
+ setPortfolioFileName(Path): PortfolioBuilder
+ AddStockToPortfolio(StockObject, float): PortfolioBuilder
StockObjectImplTest
+ StockObjectImplTest():
+ testGettingCurrentPrice(): void
+ testGettingCurrentValue(): void
+ testGettingTicker(): void
+ testGettingValueAtDate(): void
+ setup(): void
+ testGettingPriceAtDate(): void
PortfolioListImplTest
+ PortfolioListImplTest():
+ testGetPortfolio(): void
+ setUp(): void
+ testCreatePortfolioFromFileNotFound(): void
+ testGetPortfolioFileNames(): void
+ testCreatePortfolio(): void
+ testGetPortfolioInvalidName(): void
+ testCreatePortfolioFromInvalidFileContent(): void
+ testCreatePortfolioFromFile(): void
+ testCreatePortfolioFromInvalidFileContentFormat(): void
PortfolioImplTest
+ PortfolioImplTest():
+ testPortfolioImplObjectCreation(): void
+ savePortfolioToFile(): void
+ testGetPortfolioValue(): void
Utils
+ Utils():
+ getNextRandomInteger(Random, int, int): int