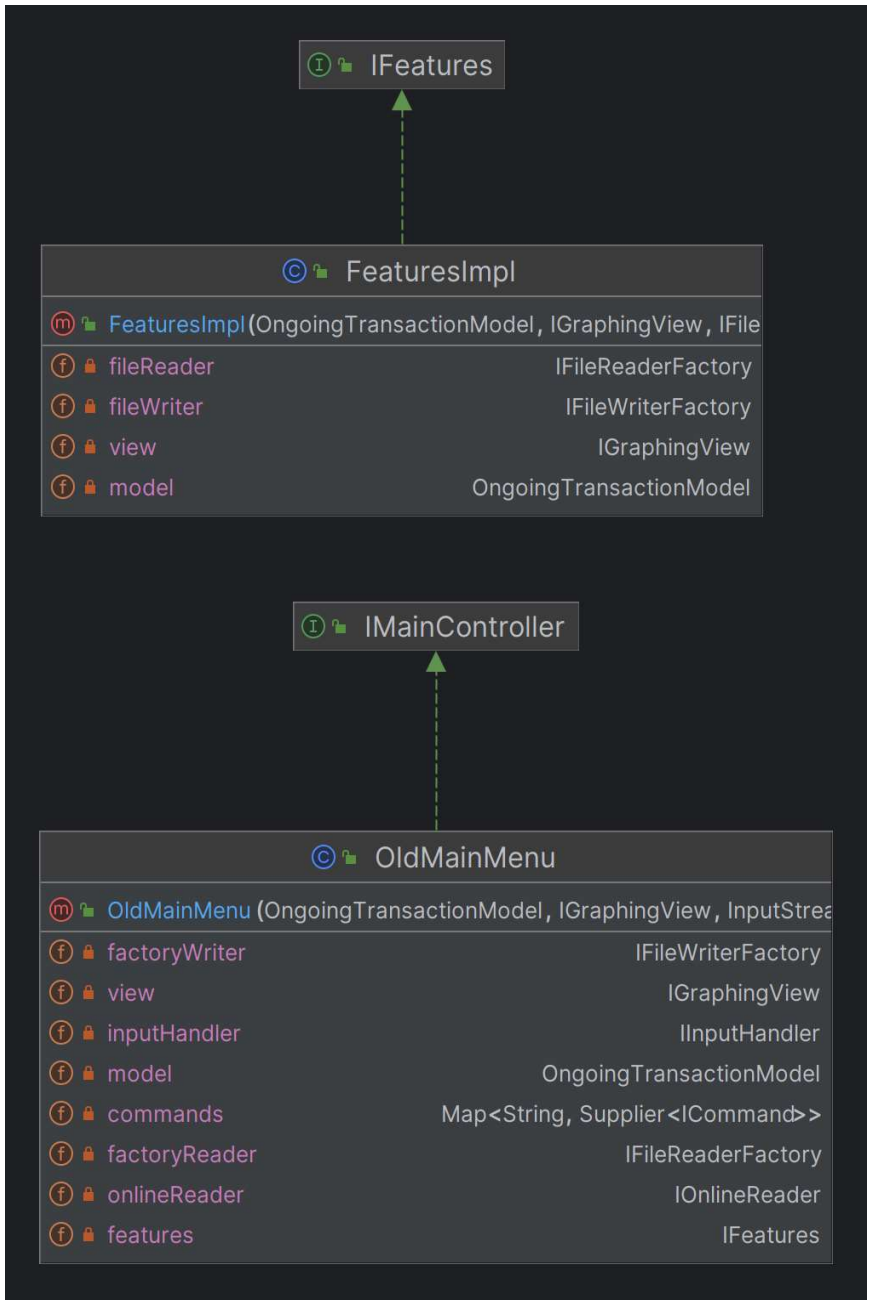
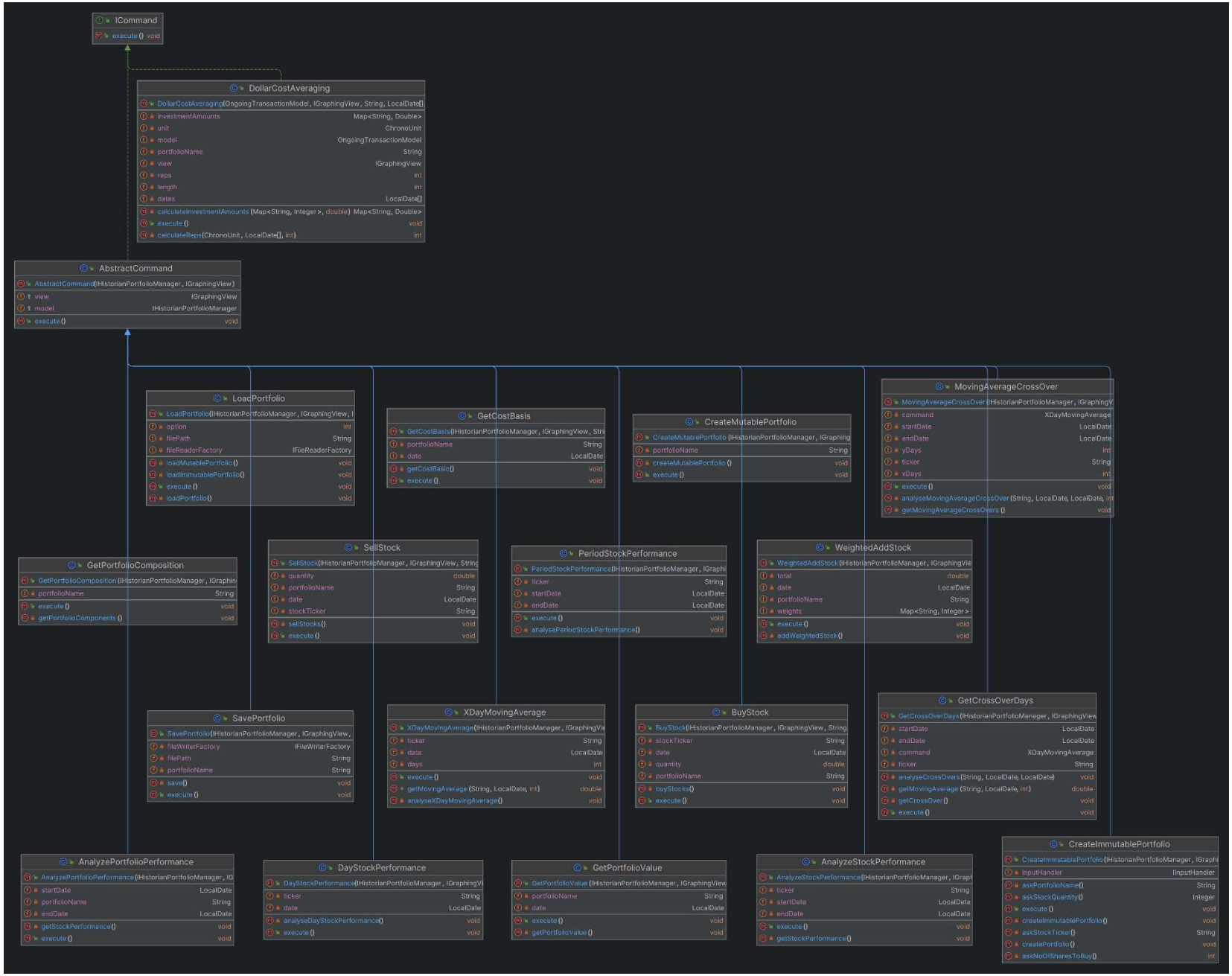
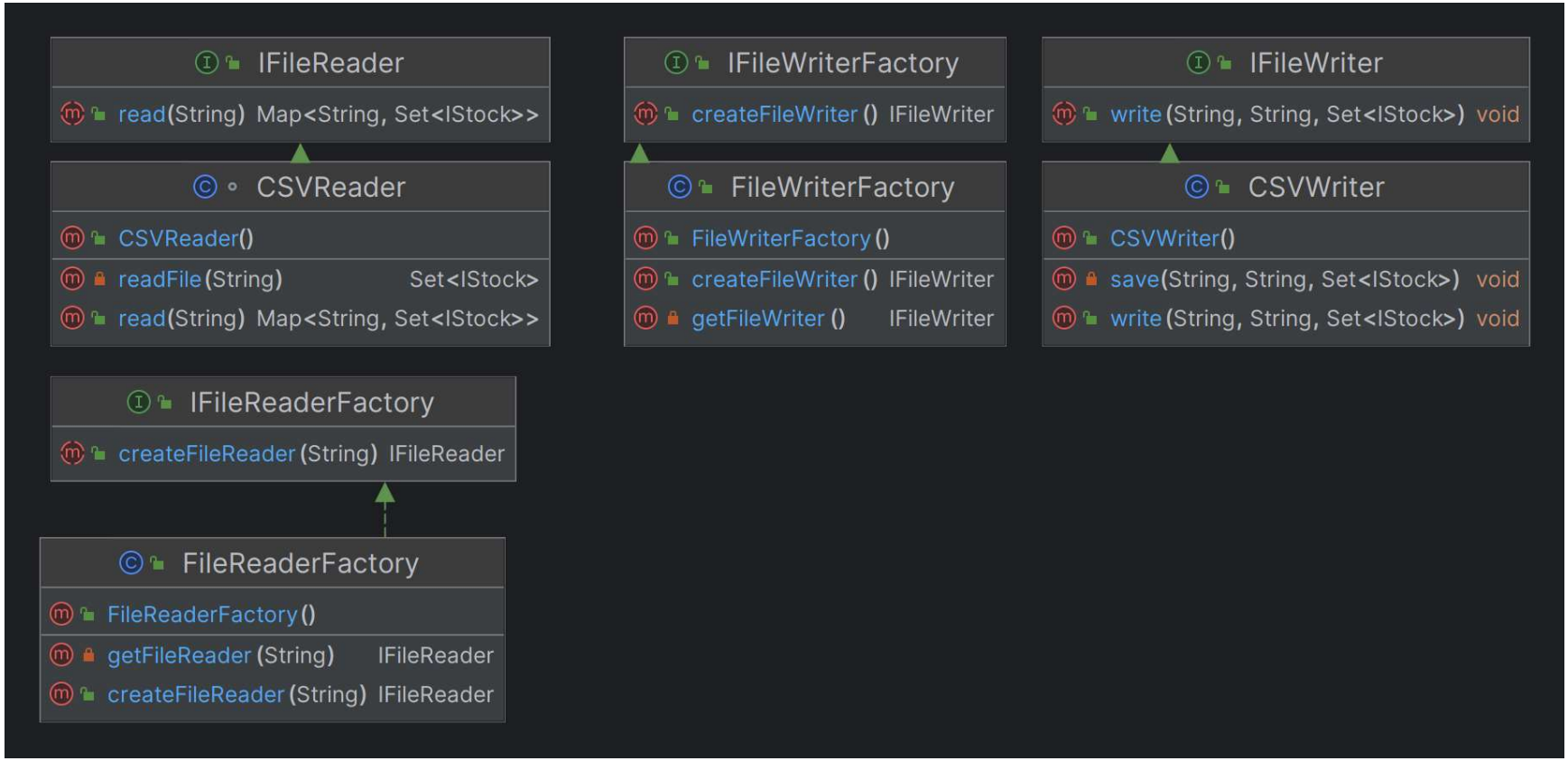
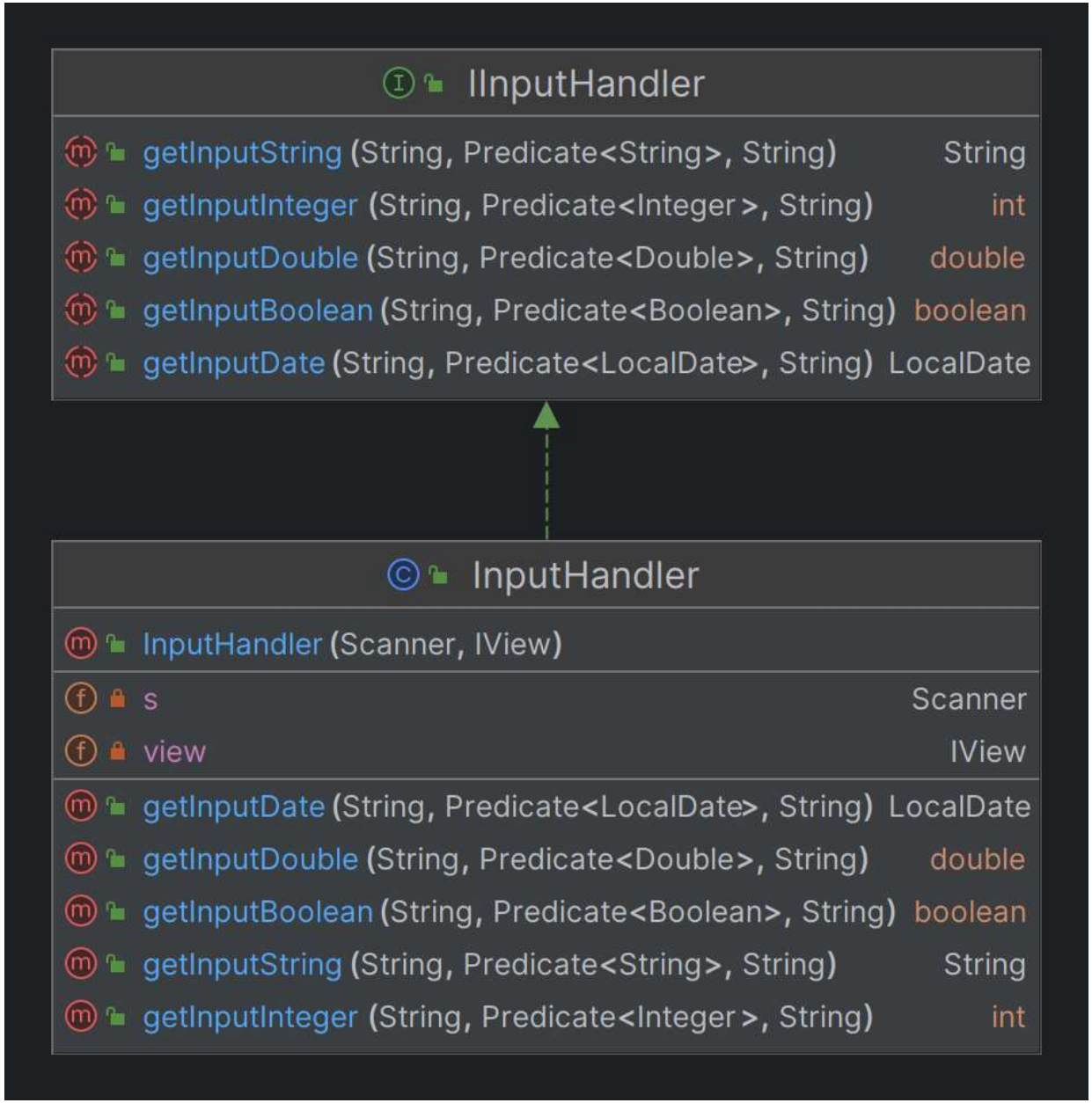
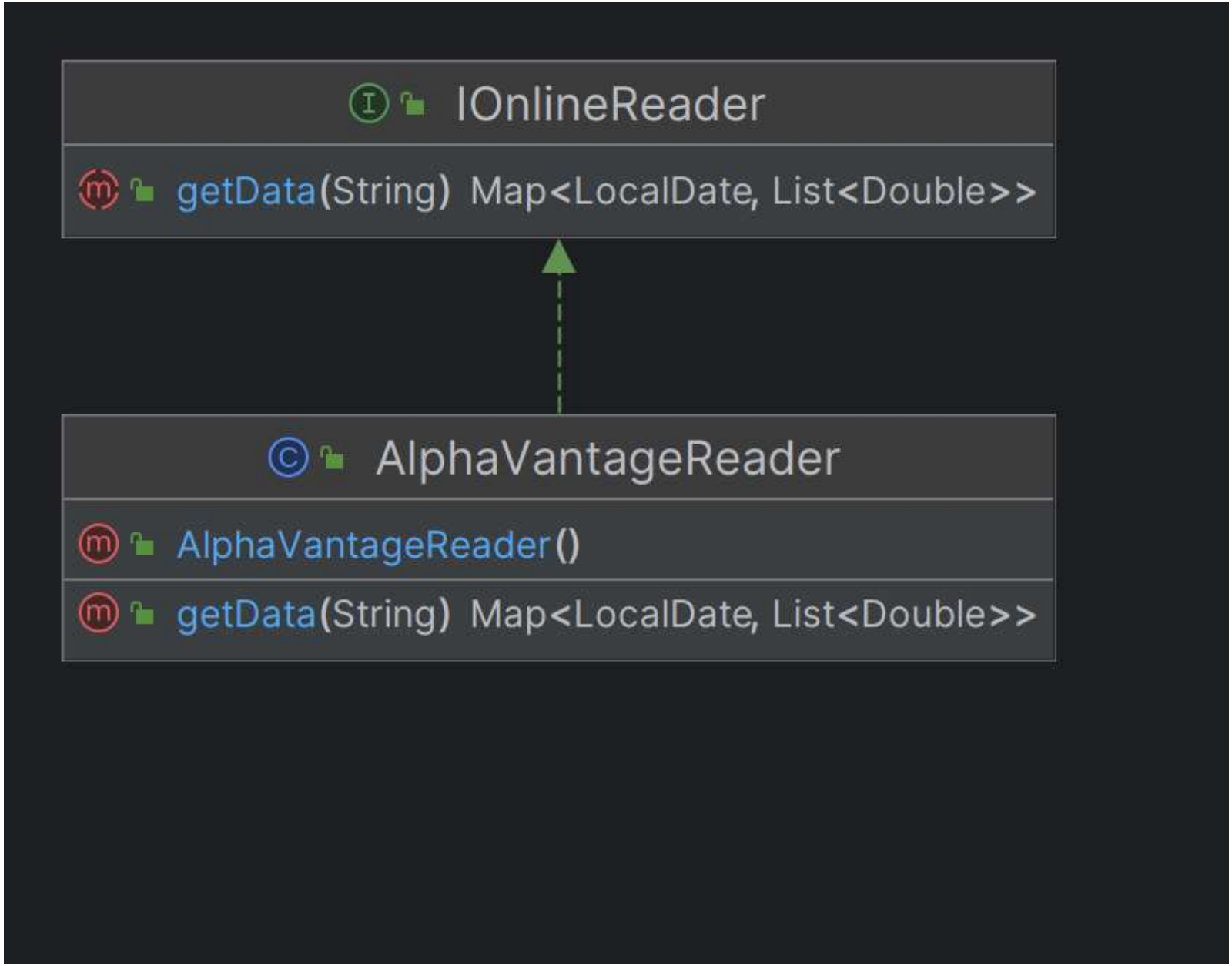
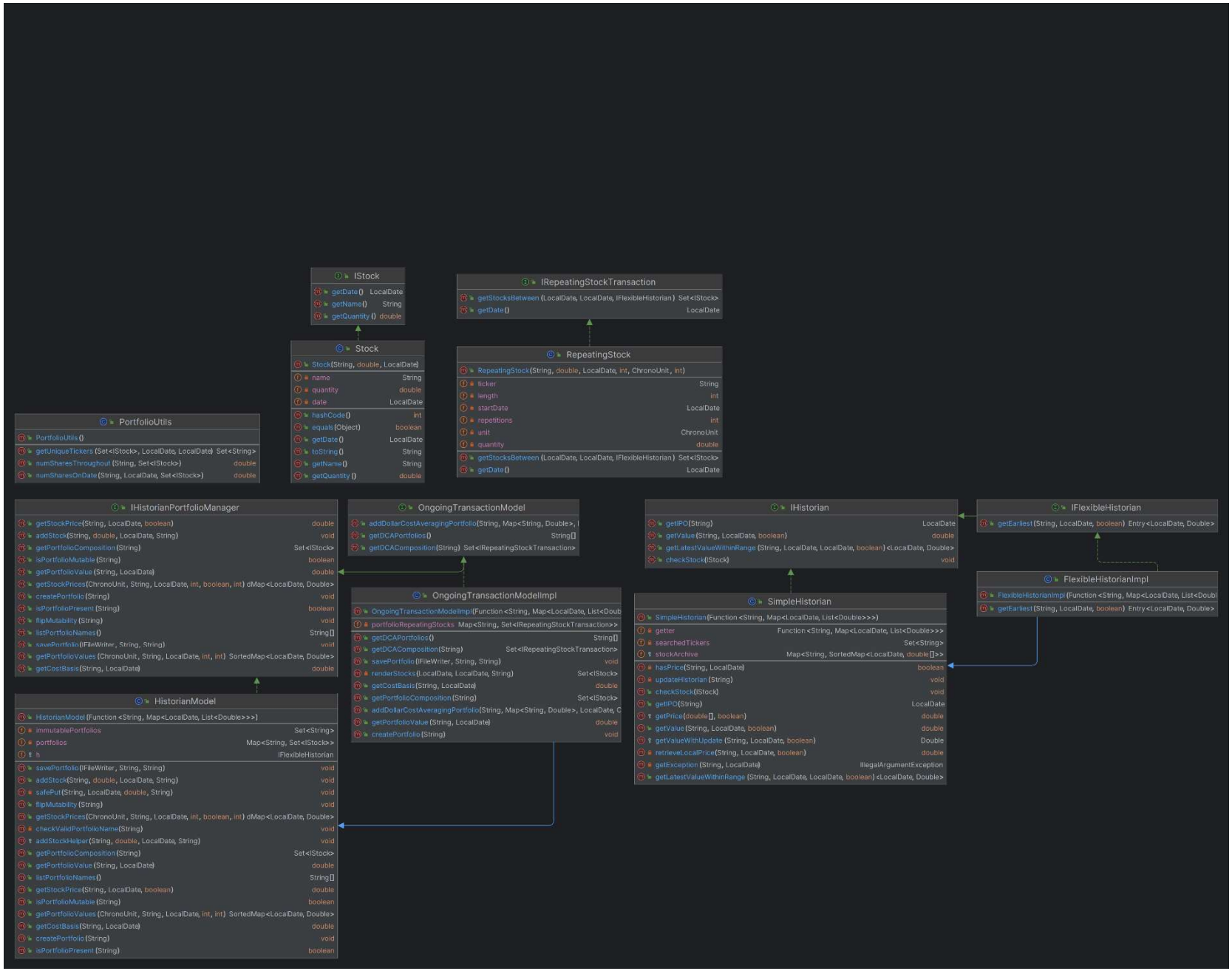

IFeatures
FeaturesImpl
FeaturesImpl(OngoingTransactionModel, IGraphingView, IFile
fileReader IFileReaderFactory
fileWriter IFileWriterFactory
view IGraphingView
model OngoingTransactionModel
IMainController
OldMainMenu
OldMainMenu (OngoingTransactionModel, IGraphingView, InputStrea
factoryWriter IFileWriterFactory
view IGraphingView
inputHandler IInputHandler
model OngoingTransactionModel
commands Map<String, Supplier<ICommand>>
factoryReader IFileReaderFactory
onlineReader IOnlineReader
features IFeatures

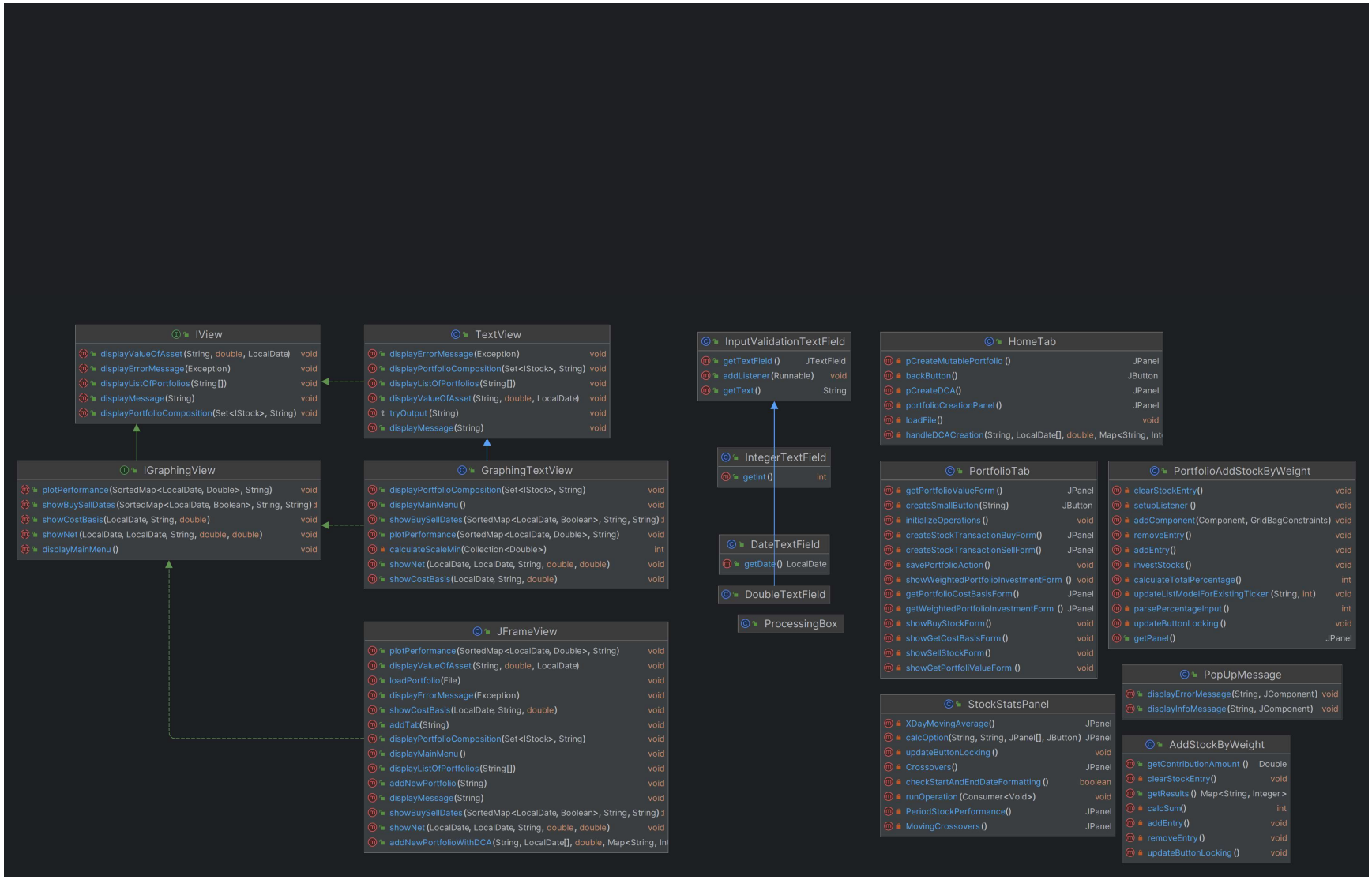

ICommand
execute() void
DollarCostAveraging
DollarCostAveraging(OngoingTransactionModel, IGraphingView, String, LocalDate[])
investmentAmounts Map<String, Double>
unit ChronoUnit
model OngoingTransactionModel
portfolioName String
view IGraphingView
reps int
length int
dates LocalDate[]
calculateInvestmentAmounts(Map<String, Integer>, double) Map<String, Double>
execute() void
calculateReps(ChronoUnit, LocalDate[], int) int
AbstractCommand
AbstractCommand(IHistorianPortfolioManager, IGraphingView)
view IGraphingView
model IHistorianPortfolioManager
execute() void
LoadPortfolio
LoadPortfolio(IHistorianPortfolioManager, IGraphingView, I
option int
filePath String
fileReaderFactory IFileReaderFactory
loadMutablePortfolio() void
loadImmutablePortfolio() void
execute() void
loadPortfolio() void
GetCostBasis
GetCostBasis(IHistorianPortfolioManager, IGraphingView, Stri
portfolioName String
date LocalDate
getCostBasic() void
execute() void
CreateMutablePortfolio
CreateMutablePortfolio(IHistorianPortfolioManager, IGraphing
portfolioName String
createMutablePortfolio() void
execute() void
MovingAverageCrossOver
MovingAverageCrossOver(IHistorianPortfolioManager, IGraphingV
command XDayMovingAverage
startDate LocalDate
endDate LocalDate
yDays int
ticker String
xDays int
execute() void
analyseMovingAverageCrossOver(String, LocalDate, LocalDate, int
getMovingAverageCrossOvers() void
GetPortfolioComposition
GetPortfolioComposition(IHistorianPortfolioManager, IGraphin
portfolioName String
execute() void
getPortfolioComponents() void
SellStock
SellStock(IHistorianPortfolioManager, IGraphingView, String
quantity double
portfolioName String
date LocalDate
stockTicker String
sellStocks() void
execute() void
PeriodStockPerformance
PeriodStockPerformance(IHistorianPortfolioManager, IGraphi
ticker String
startDate LocalDate
endDate LocalDate
execute() void
analysePeriodStockPerformance() void
WeightedAddStock
WeightedAddStock(IHistorianPortfolioManager, IGraphingVie
total double
date LocalDate
portfolioName String
weights Map<String, Integer>
execute() void
addWeightedStock() void
SavePortfolio
SavePortfolio(IHistorianPortfolioManager, IGraphingView,
fileWriterFactory IFileWriterFactory
filePath String
portfolioName String
save() void
execute() void
XDayMovingAverage
XDayMovingAverage(IHistorianPortfolioManager, IGraphingVie
ticker String
date LocalDate
days int
execute() void
getMovingAverage(String, LocalDate, int) double
analyseXDayMovingAverage() void
BuyStock
BuyStock(IHistorianPortfolioManager, IGraphingView, String
stockTicker String
date LocalDate
quantity double
portfolioName String
buyStocks() void
execute() void
GetCrossOverDays
GetCrossOverDays(IHistorianPortfolioManager, IGraphingView
startDate LocalDate
endDate LocalDate
command XDayMovingAverage
ticker String
analyseCrossOvers(String, LocalDate, LocalDate) void
getMovingAverage(String, LocalDate, int) double
getCrossOver() void
execute() void
AnalyzePortfolioPerformance
AnalyzePortfolioPerformance(IHistorianPortfolioManager, IG
startDate LocalDate
portfolioName String
endDate LocalDate
getStockPerformance() void
execute() void
DayStockPerformance
DayStockPerformance(IHistorianPortfolioManager, IGraphingV
ticker String
date LocalDate
analyseDayStockPerformance() void
execute() void
GetPortfolioValue
GetPortfolioValue(IHistorianPortfolioManager, IGraphingView
portfolioName String
date LocalDate
execute() void
getPortfolioValue() void
AnalyzeStockPerformance
AnalyzeStockPerformance(IHistorianPortfolioManager, IGraph
ticker String
startDate LocalDate
endDate LocalDate
execute() void
getStockPerformance() void
CreateImmutablePortfolio
CreateImmutablePortfolio(IHistorianPortfolioManager, IGraphi
inputHandler InputHandler
askPortfolioName() String
askStockQuantity() Integer
execute() void
createImmutablePortfolio() void
askStockTicker() String
createPortfolio() void
askNoOfSharesToBuy() int


IFileReader
read(String) Map<String, Set<IStock>>
CSVReader
CSVReader()
readFile(String) Set<IStock>
read(String) Map<String, Set<IStock>>
IFileReaderFactory
createFileReader(String) IFileReader
FileReaderFactory
FileReaderFactory()
getFileReader(String) IFileReader
createFileReader(String) IFileReader
IFileWriterFactory
createFileWriter() IFileWriter
FileWriterFactory
FileWriterFactory()
createFileWriter() IFileWriter
getFileWriter() IFileWriter
IFileWriter
write(String, String, Set<IStock>) void
CSVWriter
CSVWriter()
save(String, String, Set<IStock>) void
write(String, String, Set<IStock>) void


IInputHandler
getInputString (String, Predicate<String>, String) String
getInputInteger (String, Predicate<Integer>, String) int
getInputDouble (String, Predicate<Double>, String) double
getInputBoolean (String, Predicate<Boolean>, String) boolean
getInputDate (String, Predicate<LocalDate>, String) LocalDate
InputHandler
InputHandler (Scanner, IView)
s Scanner
view IView
getInputDate (String, Predicate<LocalDate>, String) LocalDate
getInputDouble (String, Predicate<Double>, String) double
getInputBoolean (String, Predicate<Boolean>, String) boolean
getInputString (String, Predicate<String>, String) String
getInputInteger (String, Predicate<Integer>, String) int


IOnlineReader
getData(String) Map<LocalDate, List<Double>>
AlphaVantageReader
AlphaVantageReader()
getData(String) Map<LocalDate, List<Double>>


IStock
getDate() LocalDate
getName() String
getQuantity() double
IRepeatingStockTransaction
getStocksBetween(LocalDate, LocalDate, IFlexibleHistorian) Set<IStock>
getDate() LocalDate
Stock
Stock(String, double, LocalDate)
name String
quantity double
date LocalDate
hashCode() int
equals(Object) boolean
getDate() LocalDate
toString() String
getName() String
getQuantity() double
RepeatingStock
RepeatingStock(String, double, LocalDate, int, ChronoUnit, int)
ticker String
length int
startDate LocalDate
repetitions int
unit ChronoUnit
quantity double
getStocksBetween(LocalDate, LocalDate, IFlexibleHistorian) Set<IStock>
getDate() LocalDate
PortfolioUtils
PortfolioUtils()
getUniqueTickers(Set<IStock>, LocalDate, LocalDate) Set<String>
numSharesThroughout(String, Set<IStock>) double
numSharesOnDate(String, LocalDate, Set<IStock>) double
IHistorianPortfolioManager
getStockPrice(String, LocalDate, boolean) double
addStock(String, double, LocalDate, String) void
getPortfolioComposition(String) Set<IStock>
isPortfolioMutable(String) boolean
getPortfolioValue(String, LocalDate) double
getStockPrices(ChronoUnit, String, LocalDate, int, boolean, int) dMap<LocalDate, Double>
createPortfolio(String) void
isPortfolioPresent(String) boolean
flipMutability(String) void
listPortfolioNames() String[]
savePortfolio(IFileWriter, String, String) void
getPortfolioValues(ChronoUnit, String, LocalDate, int, int) SortedMap<LocalDate, Double>
getCostBasis(String, LocalDate) double
OngoingTransactionModel
addDollarCostAveragingPortfolio(String, Map<String, Double>,
getDCAPortfolios() String[]
getDCAComposition(String) Set<IRepeatingStockTransaction>
IHistorian
getIPO(String) LocalDate
getValue(String, LocalDate, boolean) double
getLatestValueWithinRange(String, LocalDate, LocalDate, boolean) <LocalDate, Double>
checkStock(IStock) void
IFlexibleHistorian
getEarliest(String, LocalDate, boolean) Entry<LocalDate, Double>
FlexibleHistorianImpl
FlexibleHistorianImpl(Function<String, Map<LocalDate, List<Doubl
getEarliest(String, LocalDate, boolean) Entry<LocalDate, Double>
OngoingTransactionModelImpl
OngoingTransactionModelImpl(Function<String, Map<LocalDate, List<Doub
portfolioRepeatingStocks Map<String, Set<IRepeatingStockTransaction>>
getDCAPortfolios() String[]
getDCAComposition(String) Set<IRepeatingStockTransaction>
savePortfolio(IFileWriter, String, String) void
renderStocks(LocalDate, LocalDate, String) Set<IStock>
getCostBasis(String, LocalDate) double
getPortfolioComposition(String) Set<IStock>
addDollarCostAveragingPortfolio(String, Map<String, Double>, LocalDate, C
getPortfolioValue(String, LocalDate) double
createPortfolio(String) void
SimpleHistorian
SimpleHistorian(Function<String, Map<LocalDate, List<Double>>>)
getter Function<String, Map<LocalDate, List<Double>>>
searchedTickers Set<String>
stockArchive Map<String, SortedMap<LocalDate, double[]>>
hasPrice(String, LocalDate) boolean
updateHistorian(String) void
checkStock(IStock) void
getIPO(String) LocalDate
getPrice(double[], boolean) double
getValue(String, LocalDate, boolean) double
getValueWithUpdate(String, LocalDate, boolean) Double
retrieveLocalPrice(String, LocalDate, boolean) double
getException(String, LocalDate) IllegalArgumentException
getLatestValueWithinRange(String, LocalDate, LocalDate, boolean) <LocalDate, Double>
HistorianModel
HistorianModel(Function<String, Map<LocalDate, List<Double>>>)
immutablePortfolios Set<String>
portfolios Map<String, Set<IStock>>
h IFlexibleHistorian
savePortfolio(IFileWriter, String, String) void
addStock(String, double, LocalDate, String) void
safePut(String, LocalDate, double, String) void
flipMutability(String) void
getStockPrices(ChronoUnit, String, LocalDate, int, boolean, int) dMap<LocalDate, Double>
checkValidPortfolioName(String) void
addStockHelper(String, double, LocalDate, String) void
getPortfolioComposition(String) Set<IStock>
getPortfolioValue(String, LocalDate) double
listPortfolioNames() String[]
getStockPrice(String, LocalDate, boolean) double
isPortfolioMutable(String) boolean
getPortfolioValues(ChronoUnit, String, LocalDate, int, int) SortedMap<LocalDate, Double>
getCostBasis(String, LocalDate) double
createPortfolio(String) void
isPortfolioPresent(String) boolean


IView
displayValueOfAsset(String, double, LocalDate) void
displayErrorMessage(Exception) void
displayListOfPortfolios(String[]) void
displayMessage(String) void
displayPortfolioComposition(Set<IStock>, String) void
TextView
displayErrorMessage(Exception) void
displayPortfolioComposition(Set<IStock>, String) void
displayListOfPortfolios(String[]) void
displayValueOfAsset(String, double, LocalDate) void
tryOutput(String) void
displayMessage(String) void
InputValidationTextField
getTextField() JTextField
addListener(Runnable) void
getText() String
HomeTab
pCreateMutablePortfolio() JPanel
backButton() JButton
pCreateDCA() JPanel
portfolioCreationPanel() JPanel
loadFile() void
handleDCACreation(String, LocalDate[], double, Map<String, Int
IGraphingView
plotPerformance(SortedMap<LocalDate, Double>, String) void
showBuySellDates(SortedMap<LocalDate, Boolean>, String, String)
showCostBasis(LocalDate, String, double) void
showNet(LocalDate, LocalDate, String, double, double) void
displayMainMenu() void
GraphingTextView
displayPortfolioComposition(Set<IStock>, String) void
displayMainMenu() void
showBuySellDates(SortedMap<LocalDate, Boolean>, String, String)
plotPerformance(SortedMap<LocalDate, Double>, String) void
calculateScaleMin(Collection<Double>) int
showNet(LocalDate, LocalDate, String, double, double) void
showCostBasis(LocalDate, String, double) void
IntegerTextField
getInt() int
PortfolioTab
getPortfolioValueForm() JPanel
createSmallButton(String) JButton
initializeOperations() void
createStockTransactionBuyForm() JPanel
createStockTransactionSellForm() JPanel
savePortfolioAction() void
showWeightedPortfolioInvestmentForm() void
getPortfolioCostBasisForm() JPanel
getWeightedPortfolioInvestmentForm() JPanel
showBuyStockForm() void
showGetCostBasisForm() void
showSellStockForm() void
showGetPortfoliValueForm() void
PortfolioAddStockByWeight
clearStockEntry() void
setupListener() void
addComponent(Component, GridBagConstraints) void
removeEntry() void
addEntry() void
investStocks() void
calculateTotalPercentage() int
updateListModelForExistingTicker(String, int) void
parsePercentageInput() int
updateButtonLocking() void
getPanel() JPanel
DateTextField
getDate() LocalDate
DoubleTextField
ProcessingBox
JFrameView
plotPerformance(SortedMap<LocalDate, Double>, String) void
displayValueOfAsset(String, double, LocalDate) void
loadPortfolio(File) void
displayErrorMessage(Exception) void
showCostBasis(LocalDate, String, double) void
addTab(String) void
displayPortfolioComposition(Set<IStock>, String) void
displayMainMenu() void
displayListOfPortfolios(String[]) void
addNewPortfolio(String) void
displayMessage(String) void
showBuySellDates(SortedMap<LocalDate, Boolean>, String, String)
showNet(LocalDate, LocalDate, String, double, double) void
addNewPortfolioWithDCA(String, LocalDate[], double, Map<String, In
StockStatsPanel
XDayMovingAverage() JPanel
calcOption(String, String, JPanel[], JButton) JPanel
updateButtonLocking() void
Crossovers() JPanel
checkStartAndEndDateFormatting() boolean
runOperation(Consumer<Void>) void
PeriodStockPerformance() JPanel
MovingCrossovers() JPanel
PopUpMessage
displayErrorMessage(String, JComponent) void
displayInfoMessage(String, JComponent) void
AddStockByWeight
getContributionAmount() Double
clearStockEntry() void
getResults() Map<String, Integer>
calcSum() int
addEntry() void
removeEntry() void
updateButtonLocking() void