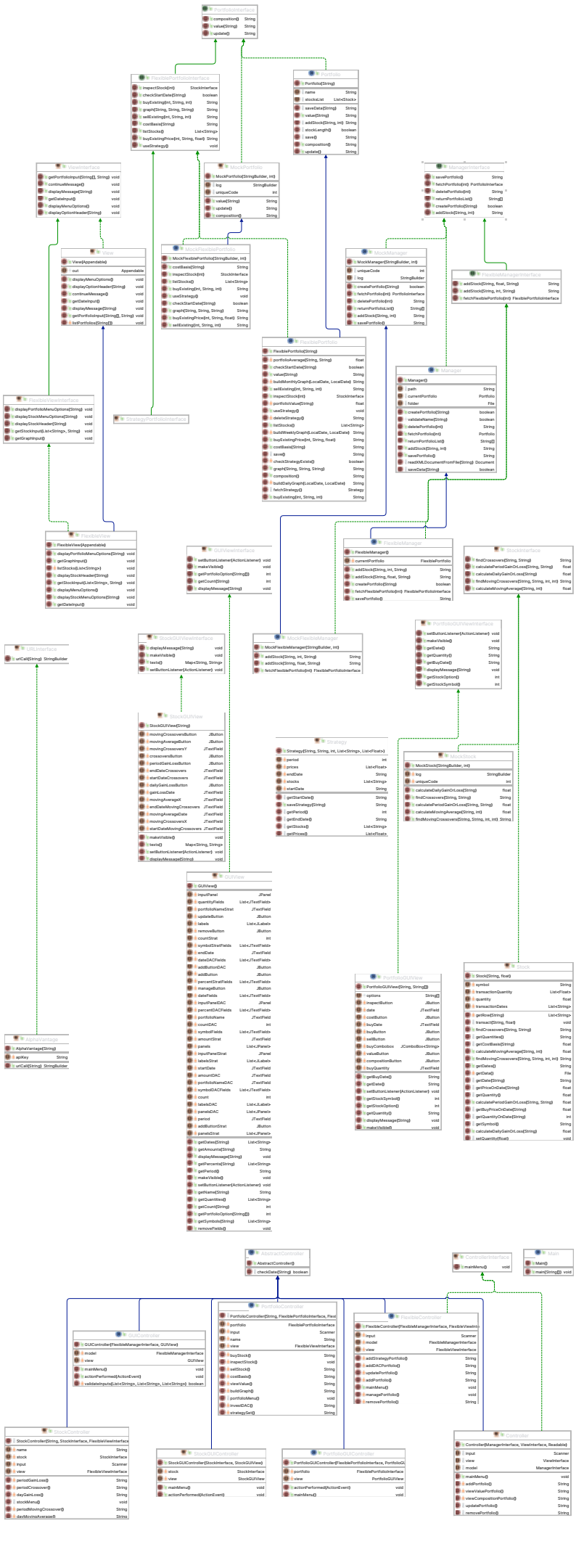
PortfolioInterface
composition() String
value(String) String
update() String
FlexiblePortfolioInterface
inspectStock(int) StockInterface
checkStartDate(String) boolean
buyExisting(int, String, int) String
graph(String, String, String) String
sellExisting(int, String, int) String
costBasis(String) String
listStocks() List<String>
buyExistingPrice(int, String, float) String
useStrategy() void
Portfolio
Portfolio(String)
name String
stockList List<Stock>
saveDate(String) String
value(String) String
addStock(String, int) String
stockLength() boolean
save() String
composition() String
update() String
ViewInterface
getPortfolioInput(String[], String) void
continueMessage() void
displayMessage(String) void
getDateInput() void
displayMenuOptions() void
displayOptionHeader(String) void
MockPortfolio
MockPortfolio(StringBuilder, int)
log StringBuilder
uniqueCode int
value(String) String
update() String
composition() String
ManagerInterface
savePortfolio() String
fetchPortfolio(int) PortfolioInterface
deletePortfolio(int) String
returnPortfolioList() String[]
createPortfolio(String) boolean
addStock(String, int) String
View
View(Appendable)
out Appendable
displayMenuOptions() void
displayOptionHeader(String) void
continueMessage() void
getDateInput() void
displayMessage(String) void
getPortfolioInput(String[], String) void
listPortfolios(String[]) void
MockFlexiblePortfolio
MockFlexiblePortfolio(StringBuilder, int)
costBasis(String) String
inspectStock(int) StockInterface
listStocks() List<String>
buyExisting(int, String, int) String
useStrategy() void
checkStartDate(String) boolean
graph(String, String, String) String
buyExistingPrice(int, String, float) String
sellExisting(int, String, int) String
MockManager
MockManager(StringBuilder, int)
uniqueCode int
log StringBuilder
createPortfolio(String) boolean
fetchPortfolio(int) PortfolioInterface
deletePortfolio(int) String
returnPortfolioList() String[]
addStock(String, int) String
savePortfolio() String
FlexibleManagerInterface
addStock(String, float, String) String
addStock(String, int, String) String
fetchFlexiblePortfolio(int) FlexiblePortfolioInterface
FlexiblePortfolio
FlexiblePortfolio(String)
portfolioAverage(String, String) float
checkStartDate(String) boolean
value(String) String
buildMonthlyGraph(LocalDate, LocalDate) String
sellExisting(int, String, int) String
inspectStock(int) StockInterface
portfolioValue(String) float
useStrategy() void
deleteStrategy() String
listStocks() List<String>
buildWeeklyGraph(LocalDate, LocalDate) String
buyExistingPrice(int, String, float) String
costBasis(String) String
save() String
checkStrategyExists() boolean
graph(String, String, String) String
composition() String
buildDailyGraph(LocalDate, LocalDate) String
fetchStrategy() Strategy
buyExisting(int, String, int) String
Manager
Manager()
path String
currentPortfolio Portfolio
folder File
createPortfolio(String) boolean
validateName(String) boolean
deletePortfolio(int) String
fetchPortfolio(int) Portfolio
returnPortfolioList() String[]
addStock(String, int) String
savePortfolio() String
readXMLDocumentFromFile(String) Document
saveDate(String) boolean
FlexibleViewInterface
displayPortfolioMenuOptions(String) void
displayStockMenuOptions(String) void
displayStockHeader(String) void
getStockInput(List<String>, String) void
getGraphInput() void
StrategyPortfolioInterface
FlexibleView
FlexibleView(Appendable)
displayPortfolioMenuOptions(String) void
getGraphInput() void
listStocks(List<String>) void
displayStockHeader(String) void
getStockInput(List<String>, String) void
displayMenuOptions() void
displayStockMenuOptions(String) void
getDateInput() void
GUIViewInterface
setButtonListener(ActionListener) void
makeVisible() void
getPortfolioOption(String[]) int
getCount(String) int
displayMessage(String) void
FlexibleManager
FlexibleManager()
currentPortfolio FlexiblePortfolio
addStock(String, int, String) String
addStock(String, float, String) String
createPortfolio(String) boolean
fetchFlexiblePortfolio(int) FlexiblePortfolioInterface
savePortfolio() String
StockInterface
findCrossovers(String, String) String
calculatePeriodGainOrLoss(String, String) float
calculateDailyGainOrLoss(String) float
findMovingCrossovers(String, String, int, int) String
calculateMovingAverage(String, int) float
PortfolioGUIViewInterface
setButtonListener(ActionListener) void
makeVisible() void
getDate() String
getQuantity() String
getBuyDate() String
displayMessage(String) void
getStockOption() int
getStockSymbol() int
URLInterface
urlCall(String) StringBuilder
StockGUIViewInterface
displayMessage(String) void
makeVisible() void
texts() Map<String, String>
setButtonListener(ActionListener) void
MockFlexibleManager
MockFlexibleManager(StringBuilder, int)
addStock(String, int, String) String
addStock(String, float, String) String
fetchFlexiblePortfolio(int) FlexiblePortfolioInterface
StockGUIView
StockGUIView(String)
movingCrossoversButton JButton
movingAverageButton JButton
movingCrossoversY JTextField
crossoversButton JButton
periodGainLossButton JButton
endDateCrossovers JTextField
startDateCrossovers JTextField
dailyGainLossButton JButton
gainLossDate JTextField
movingAverageX JTextField
endDateMovingCrossovers JTextField
movingAverageDate JTextField
movingCrossoversX JTextField
startDateMovingCrossovers JTextField
makeVisible() void
texts() Map<String, String>
setButtonListener(ActionListener) void
displayMessage(String) void
Strategy
Strategy(String, String, int, List<String>, List<Float>)
period int
prices List<Float>
endDate String
stocks List<String>
startDate String
getStartDate() String
saveStrategy(String) String
getPeriod() int
getEndDate() String
getStocks() List<String>
getPrices() List<Float>
MockStock
MockStock(StringBuilder, int)
log StringBuilder
uniqueCode int
calculateDailyGainOrLoss(String) float
findCrossovers(String, String) String
calculatePeriodGainOrLoss(String, String) float
calculateMovingAverage(String, int) float
findMovingCrossovers(String, String, int, int) String
GUIView
GUIView()
inputPanel JPanel
quantityFields List<JTextField>
portfolioNameStrat JTextField
updateButton JButton
labels List<JLabel>
removeButton JButton
countStrat int
symbolStratFields List<JTextField>
endDate JTextField
dateDACFields List<JTextField>
addButtonDAC JButton
addButton JButton
percentStratFields List<JTextField>
manageButton JButton
dateFields List<JTextField>
inputPanelDAC JPanel
percentDACFields List<JTextField>
portfolioName JTextField
countDAC int
symbolFields List<JTextField>
amountStrat JTextField
panels List<JPanel>
inputPanelStrat JPanel
labelsStrat List<JLabel>
startDate JTextField
amountDAC JTextField
portfolioNameDAC JTextField
symbolDACFields List<JTextField>
count int
labelsDAC List<JLabel>
panelsDAC List<JPanel>
period JTextField
addButtonStrat JButton
panelsStrat List<JPanel>
getDates(String) List<String>
getAmounts(String) String
displayMessage(String) void
getPercents(String) List<String>
getPeriod() String
makeVisible() void
setButtonListener(ActionListener) void
getName(String) String
getQuantities() List<String>
getCount(String) int
getPortfolioOption(String[]) int
getSymbols(String) List<String>
removeFields() void
PortfolioGUIView
PortfolioGUIView(String, String[])
options String[]
inspectButton JButton
date JTextField
costButton JButton
buyDate JTextField
buyButton JButton
sellButton JButton
buyComboBox JComboBox<String>
valueButton JButton
compositionButton JButton
buyQuantity JTextField
getBuyDate() String
getDate() String
setButtonListener(ActionListener) void
getStockSymbol() int
getStockOption() int
getQuantity() String
displayMessage(String) void
makeVisible() void
Stock
Stock(String, float)
symbol String
transactionQuantity List<Float>
quantity float
transactionDates List<String>
getRow(String) List<String>
transact(String, float) void
findCrossovers(String, String) String
getQuantities() String
getCostBasis(String) float
calculateMovingAverage(String, int) float
findMovingCrossovers(String, String, int, int) String
getDates() String
getDate() File
getDate(String) String
getPriceOnDate(String) float
getQuantity() float
calculatePeriodGainOrLoss(String, String) float
getBuyPriceOnDate(String) float
getQuantityOnDate(String) int
getSymbol() String
calculateDailyGainOrLoss(String) float
setQuantity(float) void
AlphaVantage
AlphaVantage(String)
apiKey String
urlCall(String) StringBuilder
AbstractController
AbstractController()
checkDate(String) boolean
ControllerInterface
mainMenu() void
Main
Main()
main(String[]) void
GUIController
GUIController(FlexibleManagerInterface, GUIView)
model FlexibleManagerInterface
view GUIView
mainMenu() void
actionPerformed(ActionEvent) void
validateInputs(List<String>, List<String>, List<String>) boolean
PortfolioController
PortfolioController(String, FlexiblePortfolioInterface, Flex
portfolio FlexiblePortfolioInterface
input Scanner
name String
view FlexibleViewInterface
buyStock() String
inspectStock() void
sellStock() String
costBasis() String
viewValue() String
buildGraph() String
portfolioMenu() void
investDAC() String
strategySet() String
FlexibleController
FlexibleController(FlexibleManagerInterface, FlexibleViewIn
input Scanner
model FlexibleManagerInterface
view FlexibleViewInterface
addStrategyPortfolio() String
addDACPortfolio() String
updatePortfolio() String
addPortfolio() String
mainMenu() void
managePortfolio() void
removePortfolio() String
StockController
StockController(String, StockInterface, FlexibleViewInterface
name String
stock StockInterface
input Scanner
view FlexibleViewInterface
periodGainLoss() String
periodCrossover() String
dayGainLoss() String
stockMenu() void
periodMovingCrossover() String
dayMovingAverage() String
StockGUIController
StockGUIController(StockInterface, StockGUIView)
stock StockInterface
view StockGUIView
mainMenu() void
actionPerformed(ActionEvent) void
PortfolioGUIController
PortfolioGUIController(FlexiblePortfolioInterface, PortfolioGu
portfolio FlexiblePortfolioInterface
view PortfolioGUIView
actionPerformed(ActionEvent) void
mainMenu() void
Controller
Controller(ManagerInterface, ViewInterface, Readable)
input Scanner
view ViewInterface
model ManagerInterface
mainMenu() void
addPortfolio() String
viewValuePortfolio() String
viewCompositionPortfolio() String
updatePortfolio() String
removePortfolio() String