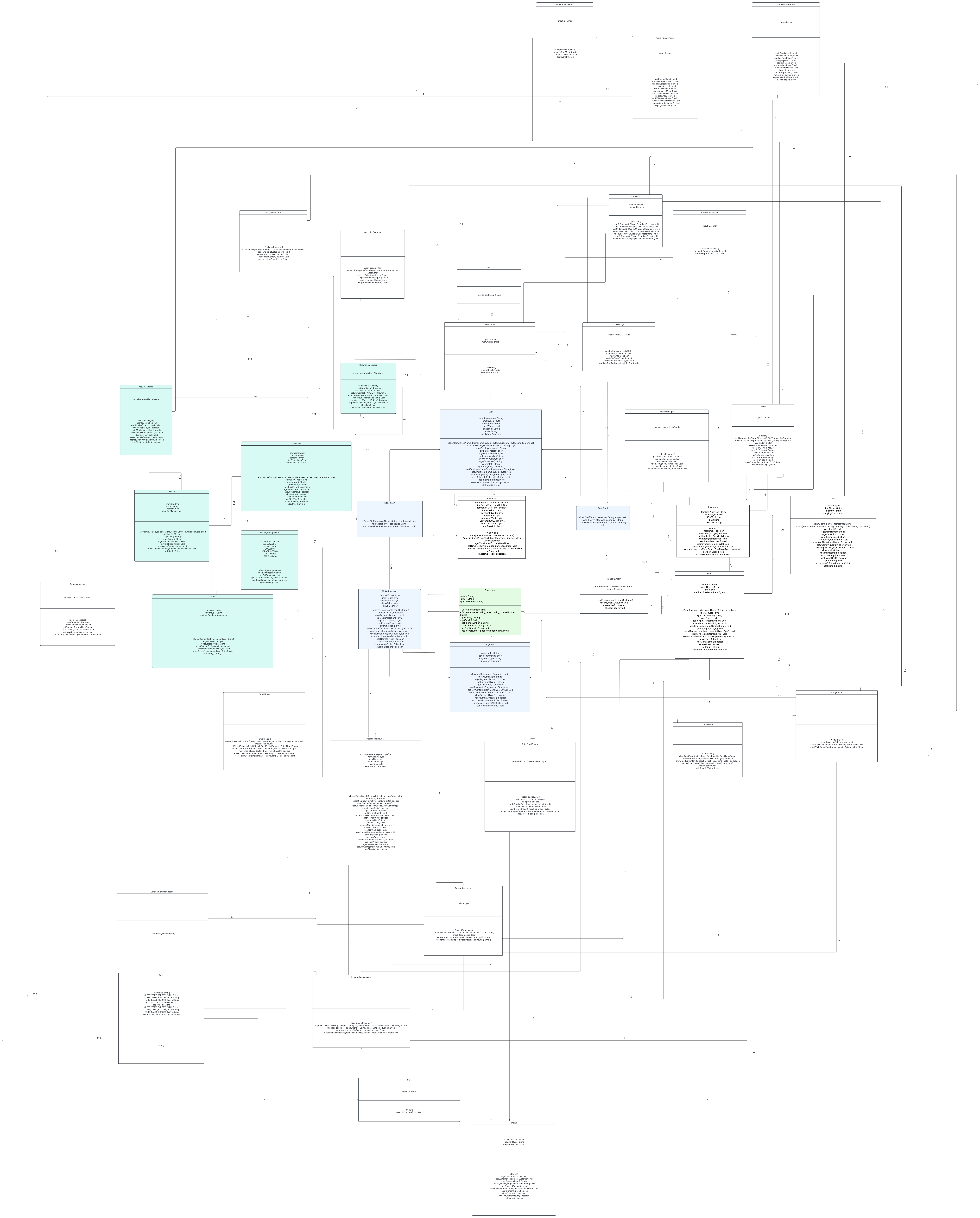
SubSubMenuStaff
-input: Scanner
+addStaffMenu(): void
+removeStaffMenu(): void
+updateStaffMenu(): void
+displayStaff(): void
SubSubMenuTicket
-input: Scanner
+addScreenMenu(): void
+removeScreenMenu(): void
+updateScreenMenu(): void
+displayScreen(): void
+addMovieMenu(): void
+removeMovieMenu(): void
+updateMovieMenu(): void
+displayMovie(): void
+addShowtimeMenu(): void
+removeShowtimeMenu(): void
+updateShowtimeMenu(): void
+displayShowtime(): void
SubSubMenuFood
-input: Scanner
+addFoodMenu(): void
+removeFoodMenu(): void
+updateFoodMenu(): void
+displayFood(): void
+addItemMenu(): void
+removeItemMenu(): void
+updateItemMenu(): void
+displayItem(): void
+addRecipeMenu(): void
+removeRecipeMenu(): void
+updateRecipeMenu(): void
+displayRecipe(): void
SubMenu
-input: Scanner
-menuWidth: short
-SubMenu()
+addOrRemoveOrDisplayOrUpdateScreen(): void
+addOrRemoveOrDisplayOrUpdateMovie(): void
+addOrRemoveOrDisplayOrUpdateShowtime(): void
+addOrRemoveOrDisplayOrUpdateRecipe(): void
+addOrRemoveOrDisplayOrUpdateItem(): void
+addOrRemoveOrDisplayOrUpdateFood(): void
+addOrRemoveOrDisplayOrUpdateFoodStaff(): void
SubMenuAnalytics
-input: Scanner
-SubMenuAnalytics()
+generateReport(staff: Staff): void
+exportReport(staff: Staff): void
AnalyticsReporter
+AnalyticsReporter()
+AnalyticsReporter(startReport: LocalDate, endReport: LocalDate)
+generateTicketSalesReport(): void
+generateFoodSalesReport(): void
+generateInventoryReport(): void
+generateItemOrderReport(): void
AnalyticsExporter
+AnalyticsExporter()
+AnalyticsExporter(startReport: LocalDate, endReport: LocalDate)
+exportTicketSalesReport(): void
+exportFoodSalesReport(): void
+exportInventoryReport(): void
+exportItemOrderReport(): void
Main
+main(args: String[]): void
MainMenu
-input: Scanner
-menuWidth: short
-MainMenu()
-createObjects():void
+showMenu(): void
StaffManager
-staffs: ArrayList<Staff>
+getStaffs(): ArrayList<Staff>
+contains(id: byte): boolean
+hasStaffs(): boolean
+addStaff(staff: Staff): void
+removeStaff(index: byte): void
+updateStaff(index: byte, staff: Staff): void
ShowtimeManager
-showtimes: ArrayList<Showtime>
+ShowtimeManager()
+hasShowtimes(): boolean
+contains(id byte): boolean
+getShowtimes(): ArrayList<Showtime>
+addShowtime(showtime: Showtime): void
+removeShowtime(index: int): void
+hasScreenID(screenID: byte); boolean
+updateShowtime(index: byte, showtime: Showtime):void
+showAllShowtimeSchedule(): void
MovieManager
-movies: ArrayList<Movie>
+MovieManager()
+hasMovies(): boolean
+getMovies(): ArrayList<Movie>
+contains(id: byte): boolean
+addMovie(movie: Movie): void
+removeMovie(movieId: byte): void
+displayAllMovies(): void
+searchMovie(movieID: byte): byte
+hasMovieID(movieID: byte): boolean
+hasTitle(title: String): boolean
Staff
-employeeName: String
-employeeId: byte
-hourlyRate: byte
-hoursWorked: byte
-schedule: String
-role: String
-analytics: Analytics
+Staff(employeeName: String, employeeId: byte, hourlyRate: byte, schedule: String)
-calculateWeeklyHours(scheduleStr: String): byte
+getEmployeeName(): String
+getEmployeeId(): short
+getHourlyRate(): byte
+getHoursWorked() byte
+getWeeklySalary(): short
+getSchedule(): String
+getRole(): String
+getAnalytics(): Analytics
+setEmployeeName(employeeName: String): void
+setEmployeeId(employeeId: byte): void
+setHourlyRate(hourlyRate: byte): void
+setSchedule(schedule: String): void
+setRole(role: String): void
+setAnalytics(analytics: Analytics): void
+toString(): String
MenuManager
-menuList: ArrayList<Food>
+MenuManager()
+getMenuList(): ArrayList<Food>
+contains(id: byte): boolean
+hasMenu(): boolean
+addMenu(menuItem: Food): void
+removeMenu(menuId: byte): void
+updateMenu(index: byte, food: Food): void
Prompt
-input: Scanner
-Prompt()
+askForAnalyticsReportTime(staff: Staff): AnalyticsReporter
+askForAnalyticsExportTime(staff: Staff): AnalyticsExporter
+askForStaff(): Staff
+askForCustomer(): Customer
+askForMovie(): Movie
+askForScreen(): Screen
+askForTime(): LocalTime
+asForDate(): LocalDate
+askStaffRole(): String
+askForFood(): Food
+askForItemRecipe(food: Food): Item
+askForItemRecipe(): Item
Showtime
- showtimeID: int
- movie: Movie
- screen: Screen
- startTime: LocalTime
-endTime: LocalTime
+ Showtime(showtimeID: int, movie: Movie, screen: Screen, startTime: LocalTime)
+getShowTimeID(): int
+ getMovie(): Movie
+ getScreen(): Screen
+ getStartTime(): LocalTime
+getEndTime(): LocalTime
+hasShowtimeID(): boolean
+hasMovie(): boolean
+hasScreen(): boolean
+hasStartTime(): boolean
+hasEndTime(): boolean
+toString(): String
Movie
- movieID: byte
- title: String
- genre: String
- durationMinutes: short
+ Movie(movieID: byte, title: String, genre: String, durationMinutes: short)
+ getMovieID(): byte
+ getTitle(): String
+ getGenre(): String
+ getDurationMinutes(): short
+ setTitle(title: String): void
+ setGenre(genre: String): void
+ setDurationMinutes(durationMinutes: short): void
+toString(): String
SeatingArrangement
-seatStatus: boolean
-capacity: short
-ROWS: byte
-COLS: byte
+RESET: STRING
+RED: String
+GREEN: String
+SeatingArrangement()
+getRowCapacity(): byte
+getColCapacity(): byte
+getSeatStatus(row: int, col: int): boolean
+setSeatStatus(row: int, col: int): void
+viewSeating(): void
ScreenManager
-screens: ArrayList<Screen>
+ScreenManager()
+hasScreens(): boolean
+contains(id: byte): boolean
+getScreens(): ArrayList<Screen>
+addScreen(screen: Screen): void
+removeScreen(idx: byte): void
+updateScreen(index: byte, screen Screen): void
Screen
- screenID: byte
- screenType: String
- seating: SeatingArrangement
+ Screen(screenID: byte, screenType: String)
+ getScreenID(): byte
+ getScreenType(): String
+ getSeating(): SeatingArrangement
+ setScreenID(screenID: byte): void
+ setScreenType(screenType: String): void
+toString(): String
TicketStaff
+TicketStaff(employeName: String, employeeId: byte, hourlyRate: byte, schedule: String)
+addNewTicketPayment(customer: Customer): void
Analytics
-timePeriodStart: LocalDateTime
-timePeriodEnd: LocalDateTime
-formatter: DateTimeFormatter
-reportWidth: short
-paymentIdWidth: byte
-timeWidth: byte
-numericWidth: byte
-veryShortStrWidth: byte
-shortStrWidth: byte
-longStrWidth: byte
+Analytics()
+Analytics(timePeriodStart: LocalDateTime)
+Analytics(timePeriodStart: LocalDateTime, timePeriodEnd: LocalDateTime)
+getTimePeriod(): LocalDateTime
+setTimePeriod(timePeriodStart: LocalDate): void
+setTimePeriod(timePeriodStart: LocalDate, timePeriodEnd: LocalDate): void
+hasTimePeriod(): boolean
FoodStaff
+FoodStaff(employeeName: String, employeeId: byte, hourlyRate: byte, schedule: String)
+addNewFoodPayment(customer: Customer): void
Inventory
-itemList: ArrayList<Item>
-inventoryFile: File
-RESET: String
-RED: String
-YELLOW: String
-Inventory()
+hasItems(): boolean
+contains(id: byte): boolean
+getItemList(): ArrayList<Item>
+getItem(itemId: byte): Item
+addItem(item: Item): void
+removeItem(itemId: byte): void
+updateItem(index: byte, item Item): void
+updateInventory(foodOrder: TreeMap<Food, byte): void
-alertLowStock(): void
-orderMoreItems(item: Item): void
Item
-itemId: byte
-itemName: String
-quantity: short
-buyingCost: short
+Item(itemId: byte, itemName: String)
+Item(itemId: byte, itemName: String, quantity: short, buyingCost: short)
+getItemId(): byte
+getItemName(): String
+getQuantity(): short
+getBuyingCost(): short
+setItemId(itemId: byte): void
+setItemName(itemName: String): void
+setQuantity(quantity: short): void
+setBuyingCost(buyingCost: short): void
+hasItemId(): boolean
+hasItemName(): boolean
+hasQuantity(): boolean
+hasBuyingCost(): boolean
+aboutItem(): void
+compareTo(otherItem: Item): int
+toString(): String
TicketPayment
-normalTicket: byte
-imaxTicket: byte
-normalPrice: byte
-imaxPrice: byte
-input: Scanner
+TicketPayment(customer: Customer)
+chooseTicket(): boolean
+setPaymentAmount(): void
+getNormalTicket(): byte
+getImaxTicket(): byte
+getNormalPrice(): byte
+getImaxPrice(): byte
+setNormalTicket(normalTicket: byte): void
+setImaxTicket(imaxTicket: byte): void
+setNormalPrice(newPrice: byte): void
+setImaxPrice(newPrice: byte): void
+hasNormalPrice(): boolean
+hasImaxPrice(): boolean
+hasNormalTicket(): boolean
+hasImaxTicket(): boolean
Customer
-name: String
-email: String
-phoneNumber: String
+Customer(name: String)
+Customer(name: String, email: String, phoneNumber: String)
+getName(): String
+getEmail(): String
+getPhoneNumber(): String
+setName(name: String): void
+setEmail(email: String): void
+setPhoneNumber(phoneNumber: String): void
FoodPayment
-orderedFood: TreeMap<Food, Byte>
-input: Scanner
+FoodPayment(customer: Customer)
+setPaymentAmount(): void
+hasOrder(): boolean
+chooseFood(): void
Food
-menuId: byte
-menuName: String
-price: byte
-recipe: TreeMap<Item: Byte>
+Food(menuId: byte, menuName: String, price: byte)
+getMenuId(): byte
+getMenuName(): String
+getPrice(): byte
+getRecipe(): TreeMap<Item, Byte>
+setMenuId(menuId: byte): void
+setMenuName(menuName: String): void
+setPrice(price: byte): void
+addRecipe(item: Item, quantityUsed: Byte): void
+removeRecipe(itemId: byte): void
+setRecipe(newRecipe: TreeMap<Item, Byte>): void
+hasMenuId(): boolean
+hasMenuName(): boolean
+hasPrice(): boolean
+toString(): String
+compareTo(otherFood: Food): int
Payment
-paymentId: String
-paymentAmount: short
-paymentType: String
-customer: Customer
+Payment(customer: Customer): void
+getPaymentId(): String
+getPaymentAmount(): short
+getPaymentType(): String
+getCustomer(): Customer
+setPaymentId(paymentId: String): void
+setPaymentType(paymentType: String): void
+setCustomer(customer: Customer): void
+hasPaymentType(): boolean
+hasPaymentAmount(): boolean
+processPaymentWithCard(): void
+processPaymentWithCash(): void
-setPaymentAmount(): void
PrettyPrinter
-PrettyPrinter()
+printDashLine(width: short): void
+writeDashLine(writer: BufferedWriter, width: short): void
+addWhiteSpace(str: String, intendedWidth: byte): String
OrderTicket
-OrderTicket()
-showTicketOptionToAdd(detail: DetailTicketBought, movieList: ArrayList<Movie>): DetailTicketBought
-askTicketQuantityToAdd(detail: DetailTicketBought): DetailTicketBought
-removeTicketOrder(detail: DetailTicketBought): DetailTicketBought
-reviewTicketOrder(detail: DetailTicketBought): boolean
+takeTicketOrder(detail DetailTicketBought): DetailTicketBought
-resetTicketSeats(detail: DetailTicketBought): DetailTicketBought
DetailTicketBought
-chosenSeats: ArrayList<byte[]>
-normalNum: byte
-imaxNum: byte
-normalPrice: byte
-imaxPrice: byte
-showtime: Showtime
+DetailTicketBought(normalPrice: byte, imaxPrice: byte)
+isEmpty(): boolean
+checkSeat(rowNum: byte, colNum: byte): boolean
+getChosenSeats(): ArrayList<byte[]>
+setChosenSeats(chosenSeats: ArrayList<byte[]>
+hasChosenSeats(): boolean
+getNormalNum(): byte
+addNormalNum(): void
+setNormalNum(normalNum: byte): void
+hasNormalNum(): boolean
+getImaxNum(): byte
+addImaxNum(): void
+setImaxNum(imaxNum: byte): void
+hasImaxNum(): boolean
+getNormalPrice(): byte
+setNormalPrice(normalPrice: byte): void
+hasNormalPrice(): boolean
+getImaxPrice(): byte
+setImaxPrice(imaxPrice: byte): void
+hasImaxPrice(): boolean
+getShowtime(): Showtime
+setShowtime(showtime: Showtime): void
+hasShowtime(): boolean
DetailFoodBought
-orderedFood: TreeMap<Food, byte>
+DetailFoodBought()
+isExist(aFood: Food): boolean
+isEmpty(): boolean
+addFood(aFood: Food, quantity: byte): void
+removeFood(aFood: Food): void
+getOrderedFood(): TreeMap<Food, Byte>
+setOrderedFood(orderedFood: TreeMap<Food, Byte>): void
+hasOrderedFood(): boolean
OrderFood
-OrderFood()
+takeFoodOrder(detail: DetailFoodBought): DetailFoodBought
-reviewFoodOrder(detail DetailFoodBought): boolean
-showFoodOptionToAdd(detail: DetailFoodBought): DetailFoodBought
-showFoodOptionToRemove(detail: DetailFoodBought): DetailFoodBought
-askQuantityToAdd(): byte
DateAndPaymentTracker
-DateAndPaymentTracker()
ReceiptGenerator
-width: byte
-ReceiptGenerator()
-createPaymentID(date: LocalDate, customerCount: short): String
-checkDate(): LocalDate
+generateFoodReceipt(detail: DetailFoodBought): String
+generateTicketReceipt(detail: DetailTicketBought): String
Path
-reportPath:String
+INVENTORY_REPORT_PATH: String
+ITEM_ORDER_REPORT_PATH: String
+FOOD_SALES_REPORT_PATH: String
+TICKET_SALES_REPORT_PATH
-exportPath: String
+INVENTORY_EXPORT_PATH: String
+ITEM_ORDER_EXPORT_PATH: String
+FOOD_SALES_EXPORT_PATH: String
+TICKET_SALES_EXPORT_PATH: String
-Path()
FilesUpdateManager
-FilesUpdateManager()
+updateTicketSalesFile(paymentId: String, paymentAmount: short, detail: DetailTicketBought): void
+updateFoodSalesFile(paymentId, String, detail: DetailFoodBought): void
+updateInventoryFile(itemList: ArrayList<Item>): void
+ updateItemOrderFile(item: Item, buyingQuantity: short, totalPrice: short): void
Order
-input: Scanner
-Order()
-askStillContinue(): boolean
Detail
-customer: Customer
-paymentType: String
-paymentAmount: short
+Detail()
+getCustomer(): Customer
+setCustomer(customer: Customer): void
+getPaymentType(): String
+setPaymentType(paymentType: String): void
+getPaymentAmount(): short
+setPaymentAmount(paymentAmount: short): void
+hasPaymentType(): boolean
+hasCustomer(): boolean
+hasPaymentAmount(): boolean
+isEmpty(): boolean
1-1
M-1
1-M
1...M
M...1
1...1