

## OrderFormIO contact: String date: String fname: String numOrdered: String originalRequest: String totPrice: String fillDataMembersWithUser(): void <<Abstract>>Furniture getContact(): String getDate(): String getFname(): String t: Manufacturer getNumOrdered(): String getOriginalRequest(): String getTotPrice(): String OrderformIO() ure(id: String, type: String, price: int, manu: Manufacturer) OrderFormIO(fname: String, contact: String, date: String, originalRequest: String, numOrdered: int, totPrice: int) orderFormToString(): String Manufacturer writeFormattedFile(): void Manufacturer bj: Manufacturer): void : String): void manuID: String void name: String g): void phone: String province: String getManuID(): String getName(): String getPhone(): String getProvince(): String Manufacturer(manuID: String, name: String, phone: String, province: String) Manufacturer() setManuID(id: String): void setName(name: String): void setPhone(num: String): void setProvince(prov: String): void FurnitureOrder category: String cheapestOrder: ArrayList <Furniture> = new ArrayList<F... furnitureInfo: FurnitureList orderPrice: int quantity: int FurnitureList type: String chairList: ArrayList<Chair> addParts(furniture: Chair, parts: int[]); int[] deskList: ArrayList<Desk> addParts(furniture: Desk, parts: int[]): int[] filingList: ArrayList<Filing> addParts(furniture: Lamp, parts: int[]): int[] lampList: ArrayList<Lamp> addParts(furniture: Filing, parts: int[]): int[] manuList: ArrayList<Manufacturer> buyAllFurnitureWithPart(posFurniture: Chair[], part: int): int buyAllFurnitureWithPart(posFurniture: Desk[], part: int): int getChairs(type: String): Chair[] buyAllFurnitureWithPart(posFurniture: Lamp[], part: int): int getChairs(): ArrayList<Chair> buyAllFurnitureWithPart(posFurniture: Filing[], part: int): int getChairsArray(): Chair[] calculateAndPrintCheapestOrder(): void getDesks(type: String): Desk[] calculateOrder(posFurniture: Chair[]): void getDesks(): ArrayList<Desk> calculateOrder(posFurniture: Desk[]): void getDesksArray(): Desk[] calculateOrder(posFurniture: Lamp[]): void getFilings(type: String): Filing[] calculateOrder(posFurniture: Filing[]): void getFilings(): ArrayList<Filing> calculateOrderPrice(): int getFilingsArray(): Filing[] findCheapestFurniture(posFurniture: Chair[], numParts: int[]): Chair getLamps(type: String): Lamp[] findCheapestFurniture(posFurniture: Desk[], numParts: int[]): Desk getLamps(): ArrayList<Lamp> findCheapestFurniture(posFurniture: Lamp[], numParts: int[]): Lamp getLampsArray(): Lamp[] findCheapestFurniture(posFurniture: Filing[], numParts: int[]): Filing getManufacturers(): ArrayList<Manufacturer> findPossibleParts(posFurniture: Chair[]): int[] getManufacturersArray(): Manufacturer[] findPossibleParts(posFurniture: Desk[]): int[] setChairs(chairs: Chair[]): void findPossibleParts(posFurniture: Filing[]): int[] setDesks(desks: Desk[]): void findPossibleParts(posFurniture: Lamp[]): int[] setFilings(filings: Filing[]): void FurnitureOrder(category: String, type: String, quantity: int, furnitureInfo: FurnitureList) setLamps(lamps: Lamp[]): void getCategory(): String setManufacturers(manufacturers: Manufacturer[]): void getCheapestOrder(): <<Abstract>>Furniture[] getFurnitureInfo(): FurnitureList getOrderPrice(): int getQuantity(): int getType(): String printOrder(): void

printRecommendedManufacturers(posFurniture: Furniture[]): void

setCategory(category: String): void
setFurnitureInfo(info: FurnitureList): void

+	setQuantity(quantity: int): void setType(type: String): void