Container

Add

Remove

<<interface>>
Command

+ execute(): void

Order
- cart: List<Food>
- totalPrice: double
- numberOfItems: int
- numberOfDrinks: int
- numberOfChips: int
- numberOfSandwiches: int

+ Order()

Food
- price: double
- size : Size
- calaries: double

+ Food()
+ getters()
+ setters()

CommandInvoker
- commands: List<Commands>

+ executeCommand(Command command): void
+ showCommandHistory(): void

Extends

Chips
None

+ Food()
+ getters()
+ setters()

Sandwich
- ingredients: List<Toppings>
- breadType: BreadType
- totalPrice: double

+ Food()
+ getters()
+ setters()

Drinks
- name: String

+ Drinks()
+ getters()
+ setters()

OrderScreen
- logger: Logger
- order: Order
- orderScreen: OrderScreen

- OrderScreen()
+ getInstance(): OrderScreen
+ run(): void
- orderScreenMenu(): void

0..*

Builder (inner Class Of Sandwich)
- ingrediants: List<Toppings>

+ setBread(Toppings breadType): Builder
+ setMeats(Toppings meatType): Builder
+ setCheeses(Toppings cheeseType): Builder
+ setVegetables(Toppings veggieType): Builder
+ setSauces(Toppings sauceType): Builder
+ setSides(Toppings sides): Builder
+ builder(): Builder
+ BLTBuilder(): SignatureSandwich
+phillyCheeseSteakSandwich(): PhillyCheeseSteakSandwich

abstract Toppings
- ingrediants: List<Toppings>

+ Food()
+ getters()
+ setters()

0..*

HomeScreen
- logger: Logger
- orderScreen: OrderScreen
- homeScreen: HomeScreen

- HomeScreen()
+ getInstance(): HomeScreen
+ run(): void
- homeScreenMenu(): void

SanwichShopApplication
- logger: Logger

+ Main(args: String[]):void

Extend

Sides
- sauceList: List<SauceType>

+ Sides()
+ getters()
+ setters()

SauceTopping
- sauceList: List<SauceType>

+ SauceTopping()
+ getters()
+ setters()

RegularTopping
- vegetableList: List<VegetableType>

+ RegularTopping()
+ getters()
+ setters()

MeatTopping
- meatList: List<MeatType>

+ MeatTopping()
+ getters()
+ setters()

CheeseTopping
- cheeseList: List<CheeseType>

+ CheeseTopping()
+ getters()
+ setters()

Sandwich Shop Application Architecture

Diagram information
This diagram illustrates the class structure and interactions within a sandwich shop application. The key components include the Builder pattern used for sandwich construction, various topping classes, drink and chip options, and application screens. The diagram also outlines the use of a CommandInvoker for handling orders.

Sandwich Shop Application SOP
1. Initialize the SandwichShopApplication.
    1.1. Configure the logger.
2. Launch the HomeScreen.
    2.1. Access the homeScreenMenu.
    2.2. Navigate to OrderScreen if ordering.
3. Use the OrderScreen to create an Order.
    3.1. Select Food items, including Sandwich, Drinks, and Chips.
    3.2. Utilize the Builder class to set sandwich options: bread, meats, cheeses, vegetables, sauces, and sides.
    3.3. Adjust order quantities for drinks and chips.
4. Use CommandInvoker to execute order commands.
    4.1. Execute the command for the current Order.
    4.2. Show the command history for review.

Record video

Summarize

Customize