**CUP**

-CupType: String (Cup,Cone,Waffle)

-BowlBasePrice: Double

-BowlMax: Int

-ConeBasePrice: Double

-ConeMax: Int

-WaffleBasePrice: Double

-WaffleMax: Int

-PricePerGram: Double

-Weight: int

-IceCreamRate: int

-ToppingRate: int

-CandyRate: int

-Price: double

+Cup() :

Creates a cup with from the generic file

+Cup(FileName: String) :

Creates a cup with from the FileName file

+GetCupType() :

Returns the cup type

+GetWeight() :

Returns the weight

+GetCupPrice() :

Returns the price of the container (cone, waffle or bowl)

+GetCupAndFillingPrice() :

Returns the price of the fillings added to the price of the cup

+IsFull() : boolean

Tells if the cup is full

+IsEmpty() : boolean

Tells if the cup is empty

+ChangeCupType(CupType: String) : boolean

Changes the cup type. Checks to see if the cup mass is more than 0 before allowing it to happen

+AddIceCream(time: int) void

Adds ice cream to the bowl by grams \* time

+AddHotTopping(time: int) void

Adds topping to the bowl by grams \* time

+AddCandyTopping(time: int) void

Adds topping to the bowl by grams \* time

**Main**

+userChoice : int

+aCup: Cup

+Kbd: Scanner

+GetUserChoice() : int

Accepts valid input from the user

+PreformUserChoice(userChoice: String, aCup : Cup ) : void

Performs the user choice

+DisplayMenu(aCup: Cup) : void

Displays the basic menu

+ChangeCupType(aCup : Cup, kbdScanner: Scanner ) : void

Performs the user choice

+AddIceCream(aCup : Cup, kbdScanner: Scanner ) : void

Adds ice cream to the cup

+AddHotTopping(aCup : Cup, kbdScanner: Scanner ) : void

Adds hot topping to the cup

+AddCandyTopping(aCup : Cup, kbdScanner: Scanner ) : void

Adds candy to the cup

+CheckOut(aCup : Cup) : void

Displays all the sales info and then resets the Cup