Shopping List

* Add a new item to the list, something you sometimes buy.
* Remove an item from the list, something you no longer ever buy.
* Move an item to a different slot. You might allow reordering by allowing the user to assign priority/category numbers, and you keep the list sorted by priority/category. Or you keep it sorted alphabetically.
* Check an item to put it on your list the next time you go shopping.
* Uncheck an item to remove it from your list the next time you go shopping.
* Print a list of just the checked items. You can take this to the store.
* Print a list of all items. You can use this to take a manual inventory of what you are out of.
* Save different Lists
* Send an email of your shopping list

# Objects

1. InterfaceItems
   1. Items
   2. Category
   3. Shops
2. InterfaceLists
   1. Lists
3. DataContainers

# Implementations

1. InterfaceItems

Methods

* 1. setItemName
  2. getItemName
  3. toString

## Classes that implements InterfaceItems

1. Items

Private

* Name
* Category
* Quantity

Methods

* getCategory
* getQuantity
* setCategory
* setQuantity
* toString

1. Category

Private

* CategoryName
* Shop

Methods

* getCategoryName
* getShop
* setCategoryName
* setShop
* toString

1. Shop

Private

* ShopName

Methods

* getShopName
* setShopName
* toString

1. Interface Lists

Methods

* setList
* getList
* getItem
* addItem
* removeItem
* sortList
* printList

## Classes that implements InterfaceList

1. Lists

Private

* Collection<InterfaceItems> List

Methods

* The same as Interface

1. DataContainers

Private

* Collection ItemList;
* Collection CategoryList;
* Collection ShopList;
* Collection ShoppingList;

Methods

* GetMethods
* SetMethods