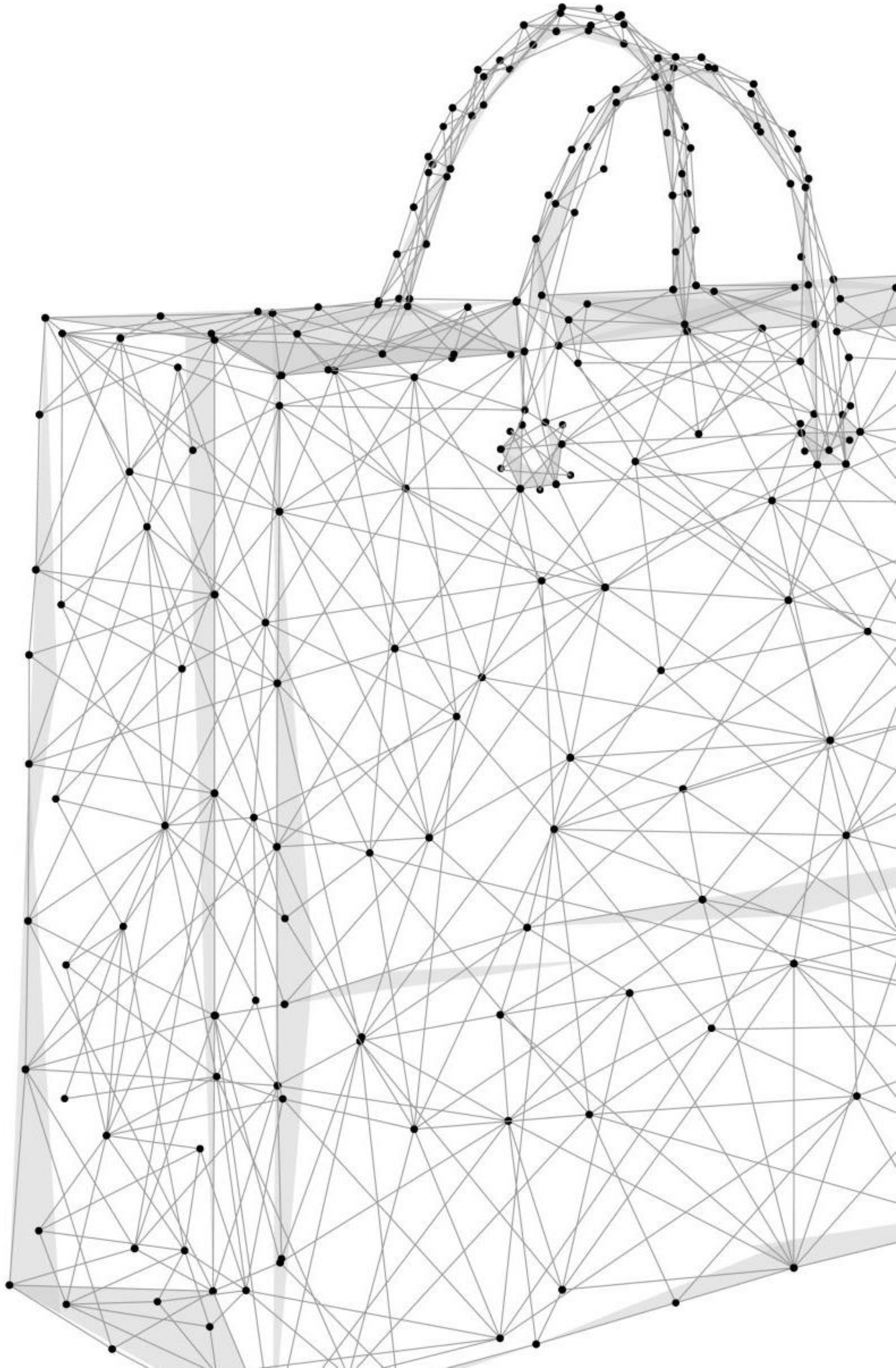
Shopping Cart GUI

CS& 141 - Group Z

Alex Luna, Shaf Mir, Nathan Willett, Stafford Winn



Overview

- Why did we choose this project?
- What are we covering today?
  - Key Features
  - Class Descriptions
  - UML Class Diagram
  - GUI Design
  - Testing & Validation

Key Features

- Bulk & non-Bulk Item Pricing
- Order Total
  - Calculation of all items in cart
- Interactive GUI
  - Scrollable list of Items
  - Button toggle for optional 10% discount
  - Item details along w/ bulk pricing if applicable
- Clear cart
  - Removes all items from the cart

Class Descriptions

- Item

  Represents an item that a user may purchase. Items contain a ‘name’ and ‘price’ and may contain a ‘bulk_quantity’ and ‘bulk_price’.

- LineItem

  Represents a line item from an order (placed by the user). Contains an ‘item’ and ‘quantity’.

- Catalog

  Represents a catalog that stores items. Contains a ‘name’ and ‘size’.

- ShoppingCart

  Handles the storage of shopping cart items, and calculation of total, including bulk pricing.

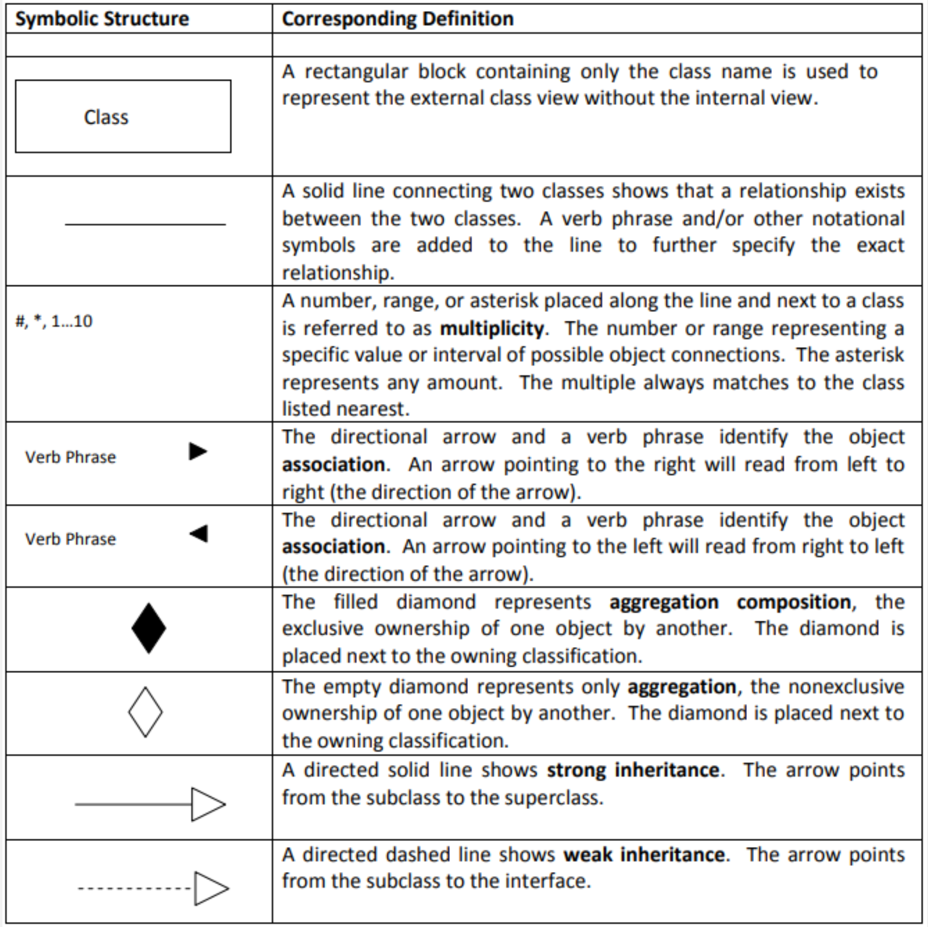
- ShoppingCartGUI

  Defines the content, layout, and behaviors of the GUI.

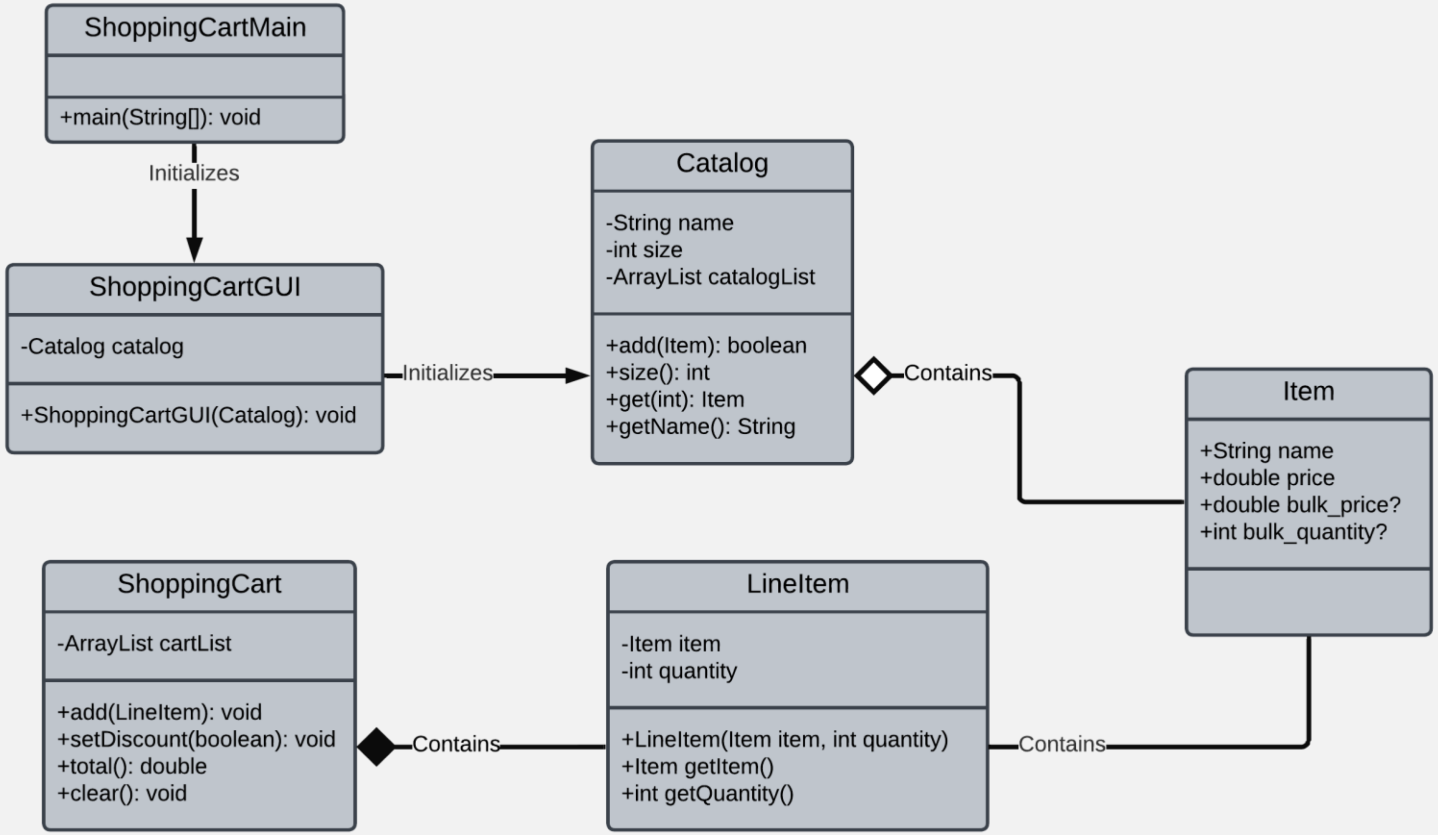
- ShoppingCartMain

  Starts the program. Instantiates the Catalog with Items and passes them to the GUI to be displayed.

UML Class Diagram



Symbolic Structure	Corresponding Definition
Class	A rectangular block containing only the class name is used to represent the external class view without the internal view.
————	A solid line connecting two classes shows that a relationship exists between the two classes. A verb phrase and/or other notational symbols are added to the line to further specify the exact relationship.
#, *, 1…10	A number, range, or asterisk placed along the line and next to a class is referred to as **multiplicity**. The number or range representing a specific value or interval of possible object connections. The asterisk represents any amount. The multiple always matches to the class listed nearest.
Verb Phrase ▶	The directional arrow and a verb phrase identify the object **association**. An arrow pointing to the right will read from left to right (the direction of the arrow).
Verb Phrase ◀	The directional arrow and a verb phrase identify the object **association**. An arrow pointing to the left will read from right to left (the direction of the arrow).
◆	The filled diamond represents **aggregation composition**, the exclusive ownership of one object by another. The diamond is placed next to the owning classification.
◇	The empty diamond represents only **aggregation**, the nonexclusive ownership of one object by another. The diamond is placed next to the owning classification.
————▷	A directed solid line shows **strong inheritance**. The arrow points from the subclass to the superclass.
- - - - - - - -▷	A directed dashed line shows **weak inheritance**. The arrow points from the subclass to the interface.



ShoppingCartMain

+main(String[]): void

Initializes

ShoppingCartGUI

-Catalog catalog

+ShoppingCartGUI(Catalog): void

Initializes

Catalog

-String name
-int size
-ArrayList catalogList

+add(Item): boolean
+size(): int
+get(int): Item
+getName(): String

Contains

Item

+String name
+double price
+double bulk_price?
+int bulk_quantity?

ShoppingCart

-ArrayList cartList

+add(LineItem): void
+setDiscount(boolean): void
+total(): double
+clear(): void

Contains

LineItem

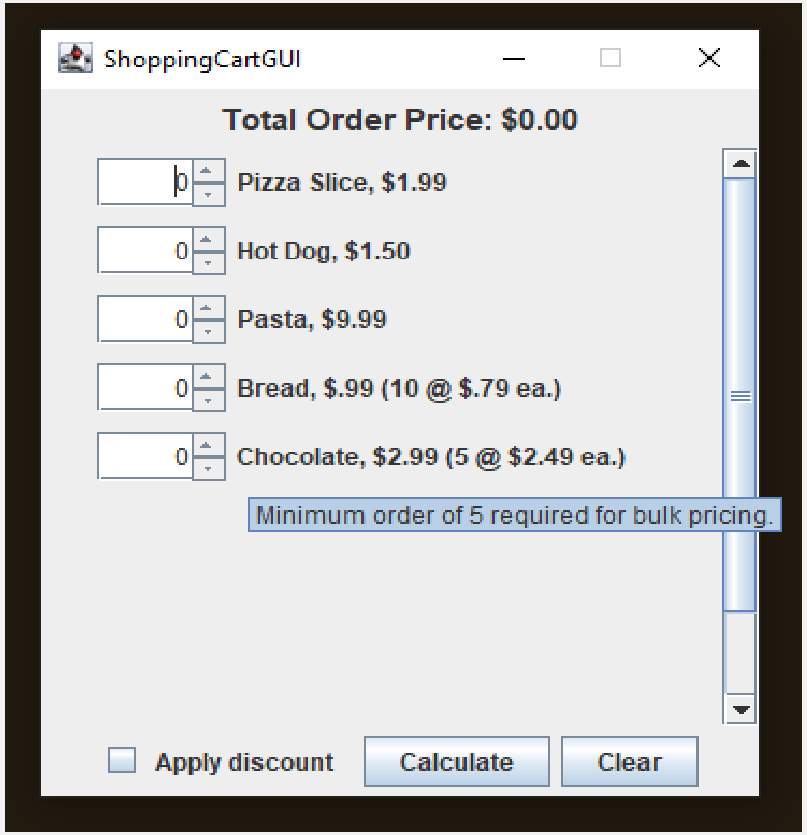
-Item item
-int quantity

+LineItem(Item item, int quantity)
+Item getItem()
+int getQuantity()

Contains

GUI Design

- JSpinners instead of textboxes
- Context sensitive tooltips and JLabels
- Properly sized GUI components
- Scroll wheel to accommodate larger catalog sets



ShoppingCartGUI

Total Order Price: $0.00

0 Pizza Slice, $1.99

0 Hot Dog, $1.50

0 Pasta, $9.99

0 Bread, $.99 (10 @ $.79 ea.)

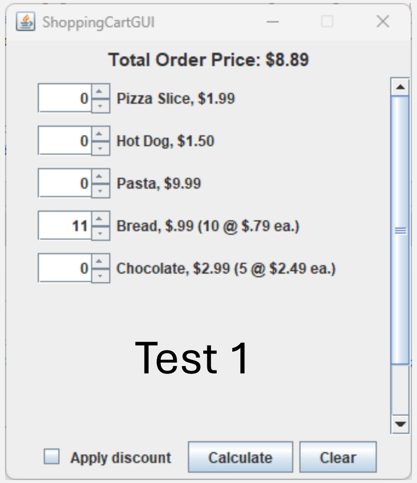
0 Chocolate, $2.99 (5 @ $2.49 ea.)

Minimum order of 5 required for bulk pricing.

□ Apply discount

Testing & Validation

- Test 1
  - Bulk pricing calculation check
- Test 2
  - Discount feature check
- Test 3
  - Input of Non-numerical characters
  - Negative integers

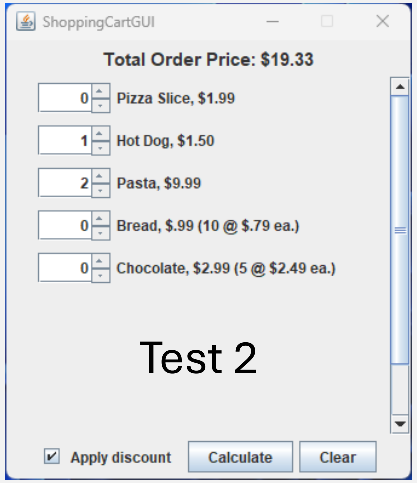


Total Order Price: $8.89

0 Pizza Slice, $1.99
0 Hot Dog, $1.50
0 Pasta, $9.99
11 Bread, $.99 (10 @ $.79 ea.)
0 Chocolate, $2.99 (5 @ $2.49 ea.)

Test 1

□ Apply discount

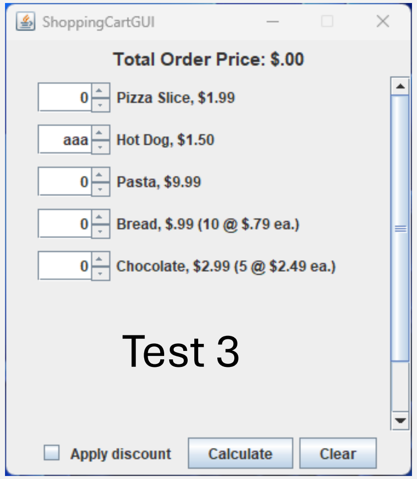


Total Order Price: $19.33

0 Pizza Slice, $1.99
1 Hot Dog, $1.50
2 Pasta, $9.99
0 Bread, $.99 (10 @ $.79 ea.)
0 Chocolate, $2.99 (5 @ $2.49 ea.)

Test 2

☑ Apply discount

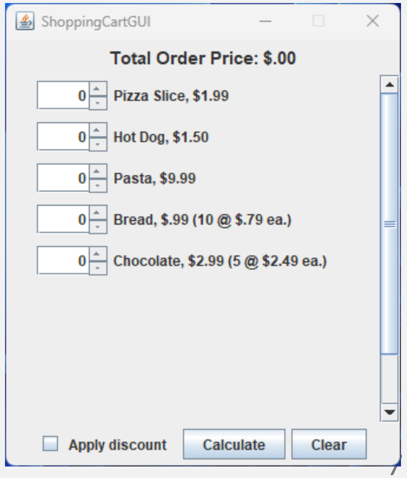


Total Order Price: $.00

0 Pizza Slice, $1.99
aaa Hot Dog, $1.50
0 Pasta, $9.99
0 Bread, $.99 (10 @ $.79 ea.)
0 Chocolate, $2.99 (5 @ $2.49 ea.)

Test 3

□ Apply discount



Total Order Price: $.00

0 Pizza Slice, $1.99
0 Hot Dog, $1.50
0 Pasta, $9.99
0 Bread, $.99 (10 @ $.79 ea.)
0 Chocolate, $2.99 (5 @ $2.49 ea.)

□ Apply discount

Resources + Documentation

- Class Diagram Legend
  - https://www.cise.ufl.edu/~pjd/courses/3502/references/UMLExternalClassView.pdf
- Javadocs
  - Group-Z/app/docs/index.html at main · nathan-willett/Group-Z (github.com)
- Source
  - Group-Z/app/src/main/java at main · nathan-willett/Group-Z (github.com)

Q&A

Thank you.