**Program 1: Array Bag**

Jordon Paynter

Department of Computer Science, Colorado State University Global

CSC 400: Data Structures and Algorithms

Professor Santosh Kumar Gottipamula

September 24, 2023

**Program 1: Array Bag**

Figures start on page 11.

# **Pseudocode for CTA2 Program 1:**

**Class MyType**:

Private Integer id

Private String name

Private Integer calories

Constructor MyType(id, name, calories):

Set this.id to id

Set this.name to name

Set this.calories to calories

Function getId():

Return id

End Method

Function getName():

Return name

End Method

Function getCalories():

Return calories

End Method

Function setCalories(calories):

Set this.calories to calories

End Method

Function toString():

Return "Item{" + "id=" + id + ", name='" + name + "', calories=" + calories + '}'

End Method

**Interface BagInterfaceMyType** {

// Add an item to the end of the bag

Method add(item: MyType)

// Remove the first occurrence of the item from the bag

Method removeFirst()

// Remove a specific item from the bag

Method removeItem(item: MyType)

// Check if the bag contains a specific item

Method contains(item: MyType): Boolean

// Get the number of times a specific item is in the bag

Method getFrequencyOf(item: MyType): Integer

// Get the current number of items in the bag

Method getCurrentSize(): Integer

// Check if the bag is empty

Method isEmpty(): Boolean

// Remove all items from the bag

Method clear()

// Check if the bag is full

Method isFull(): Boolean

// Return a copy of the bag as an array

Method toArray(): MyType[]

// Display the items in the bag

Method display()

}

End BagInterfaceMyType

**Class ShoppingCart:**

// Fields

Private MyType[] myBag

Private int numberOfItems

Private final int Default\_Capacity = 10

// Constructor

Public ShoppingCart():

// Initialize the bag

myBag = New MyType[Default\_Capacity]

numberOfItems = 0

End Constructor

// Method to add an item to the bag

Public Void add(MyType item):

If isFull():

Print "Bag is full"

Else:

// Add the item to the bag

myBag[numberOfItems] = item

Increment numberOfItems

End Method

// Method to remove the first item from the bag

Public Void removeFirst():

If isEmpty():

Print "Bag is empty"

Else:

// Shift all items to the left

For i = 0 To numberOfItems - 2:

myBag[i] = myBag[i + 1]

End For

Set myBag[numberOfItems - 1] = Null

Decrement numberOfItems

End Method

// Method to remove a specific item from the bag

Public Void removeItem(MyType item):

If isEmpty():

Print "Bag is empty"

Else:

// Iterate through the bag to find and remove the item

For i = 0 To numberOfItems - 1:

If myBag[i].getId() == item.getId():

// Shift items after the found item to the left

For j = i To numberOfItems - 2:

myBag[j] = myBag[j + 1]

End For

Set myBag[numberOfItems - 1] = Null

Decrement numberOfItems

End If

End For

End Method

// Method to check if a specific item is in the bag

Public Boolean contains(MyType item):

For i = 0 To numberOfItems - 1:

If myBag[i].getId() == item.getId():

Return True

End For

Return False

End Method

// Method to get the frequency of a specific item in the bag

Public Int getFrequencyOf(MyType item):

Int count = 0

For i = 0 To numberOfItems - 1:

If myBag[i].getId() == item.getId():

Increment count

End For

Return count

End Method

// Method to get the current number of items in the bag

Public Int getCurrentSize():

Return numberOfItems

End Method

// Method to check if the bag is empty

Public Boolean isEmpty():

Return numberOfItems == 0

End Method

// Method to clear all items from the bag

Public Void clear():

For i = 0 To numberOfItems - 1:

Set myBag[i] = Null

End For

Set numberOfItems = 0

End Method

// Method to check if the bag is full

Public Boolean isFull():

Return numberOfItems == Default\_Capacity

End Method

// Method to create and return a copy of the bag as an array

Public MyType[] toArray():

// Create a copy of the bag using Arrays.copyOf

MyType[] copy = Arrays.copyOf(myBag, numberOfItems)

Return copy

End Method

// Method to display the items in the bag

Public Void display():

For i = 0 To numberOfItems - 1:

Print myBag[i]

End For

End Method

**End ShoppingCart Class**

**class BagDemo**

shoppingCart: ShoppingCart

userContinue: boolean

items: MyType[]

method main(args: String[])

initialize shoppingCart

initialize userContinue as true

initialize items as empty array

// Test adding items into the bag

loop i from 0 to 9

item = generateMyTypeForCart()

shoppingCart.add(item)

end loop

// Display the shopping cart after generating 9 items

shoppingCart.display()

loop while userContinue is true

shoppingCart.display()

displayMenu()

userChoice = validateUserMenuChoice()

switch userChoice

case "1"

addNewItem(shoppingCart)

case "2"

shoppingCart.removeFirst()

case "3"

removeSpecificItem(shoppingCart)

case "4"

checkIfItemInCart(shoppingCart)

case "5"

getFrequencyOfItem(shoppingCart)

case "6"

displayNumberOfItems(shoppingCart)

case "7"

displayCartIsEmpty(shoppingCart)

case "8"

shoppingCart.clear()

case "9"

shoppingCart.display()

case "10"

exitApplication()

end switch

end loop

end method

// Method to generate a random MyType object

function generateMyTypeForCart() returns MyType

id = random integer between 0 and 11

name = "Product" + id

stock = random integer between 0 and 59

return new MyType(id, name, stock)

End Method

// Method to validate user's menu choice

function validateUserMenuChoice(scnr2: Scanner) returns String

userInput = scnr2.nextLine()

while userInput is not in ["1", "2", "3", "4", "5", "6", "7", "8", "9", "10",”11”]

print "Invalid input. Please enter a number between 1 and 11"

userInput = scnr2.nextLine()

end while

return userInput

End Method

// Method displayMenu(){

Display all possible options and ask for user input

Return void

End Method

// Method to validate user input for integers

function validateUserInputOnlyNumbers(scnr3: Scanner) returns int

int userInput

while true

if scnr3.hasNextInt()

userInput = scnr3.nextInt()

scnr3.nextLine() // Consume the newline character

break // Exit the loop if valid input is provided

else

print "Invalid input. Please enter a number"

scnr3.nextLine() // Consume the invalid input line

end if

end while

return userInput

End Method

End DemoBag class

**Figure 1.**

Generated Values inserted into shoppingCart.

A screenshot of a computer program

Description automatically generated

**Figure 2.**

Add a Single Item into Cart

A screenshot of a computer program

Description automatically generated

**Figure 3.**

Is the Cart Empty?

A screenshot of a computer program

Description automatically generated

**Figure 4.**

Is the Cart Full?

A screenshot of a computer program

Description automatically generated

**Figure 5.**

Remove First Item from Shopping Cart.

**A screenshot of a computer program

Description automatically generated**

**Figure 6.**

Remove a specific item from Shopping Cart.

A screenshot of a computer program

Description automatically generated

**Figure 7.**

toArray()

A screenshot of a computer program

Description automatically generated

**Figure 8.**

How many same Item in the cart?

A screenshot of a computer program

Description automatically generated

**Figure 9.**

Number of Items in the cart

A screenshot of a computer program

Description automatically generated

**Figure 10.**

Clear Shopping Cart.

A screenshot of a computer program

Description automatically generated