Menu App

Menu-driven Applications are very useful in the coding world. They allow a program to receive data directly from a user. The user is provided a menu, and then is instructed to select an option from that menu. The chosen option will be used by the program to execute a code path (or branch) specific to that option.

In the Back End, the menus we create are text based. Adding a Client, or Front End program, which reads in user data, and then communicates to a Back End Server could provide a graphical user interface for the Menu.

Menu-driven applications are used in a variety of industries, including but not limited to computing, application development, banking (ATMs), websites, tablets, self-guided machines, word-processors, gaming, and more

Menu-Driven Application:

Advantages:

- User-friendly

  - Provide guidance to the user
  - No need for a user to remember commands
- Allow a user to control how and in what order a program executes

Disadvantages:

- Difficulty finding content, especially with nested sub-menus

Menu-Driven Application -- ArrayIceCreamOrder Example:

MenuDriven Applications are really useful, and here is another example using Arrays.

ArrayIceCreamOrder.java Class:

This is the class that is used to instantiate an *ArrayIceCreamOrder* Object. Notice that this class has fields, a default constructor, additional constructors, and a describe method.

```
package com.promineotech.application;

public class ArrayIceCreamOrder {
    String user;
    String size;
    String cupOrCone;
    String[] flavors = new String[3];

    // Default Constructor
    public ArrayIceCreamOrder() {
        this.user = "";
        this.size = "";
        this.cupOrCone = "";
    }
    // Additional Constructors
    public ArrayIceCreamOrder(String user, String[] flavors) {
        this.user = user;
        this.flavors = flavors;
    }
    public ArrayIceCreamOrder(String user, String[] flavors, String size, String cupOrCone) {
        this.user = user;
        this.flavors = flavors;
        this.size = size;
        this.cupOrCone = cupOrCone;
    }

    //Getters & Setters
    public String getUser() { return this.user; }
    public void setUser(String user) { this.user = user; }
    public String[] getFlavors() { return this.flavors; }
    public void setFlavors(String[] flavors) { this.flavors = flavors; }
    public String getSize() { return this.size; }
    public void setSize(String size) { this.size = size; }
    public String getCupOrCone() { return this.cupOrCone; }
    public void setCupOrCone(String cupOrCone) { this.cupOrCone = cupOrCone; }

    // Describe Method
    public void describe(int iceCreamCount) {
        int printFlavors = 0;
        int boxSize = 40;
        char star = '*';

        for (int i = 0; i < boxSize; i++) {
            System.out.print(star);
        }
        System.out.println("\nIce Cream Order:\n----------------");
        System.out.print("Name:\n\t" + this.user + "\nOrder: \n\t");
        System.out.println(this.size + " " + this.cupOrCone);

        System.out.print("Flavors:\n\t");
        while (printFlavors < iceCreamCount) {
            System.out.print(this.flavors[printFlavors++]);
            if (printFlavors == iceCreamCount-1) {
                System.out.print (" & \n\t");
            } else if (printFlavors < iceCreamCount-1) {
                System.out.print (", \n\t");
            }
        }

        System.out.println();
        for (int i = 0; i < boxSize; i++) {
            System.out.print(star);
        }
    } // end of describe
} // end of ArrayIceCreamOrder
```

This is the *ArrayMenuInput* Application that uses the above Object.

```
package com.promineotech.application;
import java.util.Scanner;
```

```
public class ArrayMenuInput {
    static String[] iceCreamList = new String[3];
    static int iceCreamCounter = 0;
    static Scanner input = new Scanner(System.in);
    private static final int maxScoops = 3;
    private static String[] options = {"Vanilla", "Chocolate", "Mint Chocolate Chip", "Strawberry", "Black Raspberry",
    "Chocolate Peanut Butter Cup", "Bubble Gum", "Banana Fudge","End of Ice Cream Selection"};

    public static void main(String[] args) {
        // **********************************************************
        // Future enhancement -- Create a List of IceCreamOrders,
        // and then add the order to the list, and prompt the User
        // and ask whether their order is complete or not.
        // **********************************************************

        ArrayIceCreamOrder userOrder = new ArrayIceCreamOrder();
        boolean finished = false;
        System.out.println("\n\n-------------------------\nThis Example Uses Arrays!\n-------------------------\n\n\n");
        System.out.println( "------------------------------\nWelcome to The Ice Cream Stand\n---------------------------
        ---\n");

        while (!finished) {
            System.out.print("Enter a name for your order: ");
            userOrder.setUser(input.nextLine());
            int counter = 0;
            boolean willOrderIceCream = true;
            int userInput = 0;
            System.out.println("\n\n" + userOrder.getUser() + ", you may choose up to 3 flavors of ice cream!\n\n");
            do {
                if (counter == 0) {
                    printFlavorOptions();
                }
                if (counter >= 2) {
                    System.out.print("Enter your selection: ");
                    userInput = input.nextInt();
                    switch (userInput) {
                        case (1):
                        case (2):
                        case (3):
                        case (4):
                        case (5):
                        case (6):
                        case (7):
                        case (8):
                            printChoice(options[userInput - 1], counter);
                            break;
                        case (9):
                            if (counter == 0) {
                                System.out.println("No selection has been made! Do you want to order ice cream?");
                                if (willOrderIceCream = checkOrderCancellation()) {
                                    userInput = 0;
                                    counter--;
                                    break;
                                }
                            }
                            System.out.println("End of Ice Cream Selection!\n");
                            break;
                        default:
```

```
                        System.out.println("Invalid Selection Choice");
                        counter--;
                    }
                }
            counter++;
        } while (userInput != 9 && counter <= 4); // ExitCriteria == 9

        /* Set field in userOrder */
        userOrder.setFlavors(iceCreamList);
        if (willOrderIceCream) {
            // Choose the Size of the Ice Cream: Small, Medium or Large
            int userInputSize = 0;
            int size = iceCreamCounter;
            if (size == maxScoops) {
                System.out.println("With 3 Ice Cream flavors, the size you get is: LARGE");
                userOrder.setSize("Large");
                userInputSize = size;
            } else {
                printSizeOptions();
                userInputSize = input.nextInt();
                while ((userInputSize != 1) && (userInputSize != 2) && (userInputSize != 3)) {
                    System.out.print("Please enter a valid ice cream size selection: ");
                    userInputSize = input.nextInt();
                }
                System.out.print("The size you have chosen is: ");
                switch (userInputSize) {
                    case (1):
                        System.out.println("SMALL");
                        userOrder.setSize("Small");
                        break;
                    case (2):
                        System.out.println("MEDIUM");
                        userOrder.setSize("Medium");
                        break;
                    case (3):
                        System.out.println("LARGE");
                        userOrder.setSize("Large");
                        break;
                }
            }

            System.out.println("\n");
            // Choose cup or cone
            int order = 0;
            printCupOrConeOptions();
            order = input.nextInt();
            while ((order != 1) && (order != 2)) {
                System.out.print("Please enter a valid entry: ");
                order = input.nextInt();
            }
            System.out.print("You have chosen a: ");
            switch (order) {
            case (1):
                System.out.println("CUP");
                userOrder.setCupOrCone("Cup");
                break;
            case (2):
                System.out.println("CONE");
```

```
                    userOrder.setCupOrCone("Cone");
                    break;
                }
                System.out.println("\n");
                userOrder.describe(iceCreamCounter);
                System.out.println("\n");
            } else {
                System.out.println("Order Cancelled!");
            }
            // **********************************************************
            // Future enhancement -- Add a menu here to ask the
            // user if they have another request to add to this order.
            // **********************************************************
            finished = true;
        } // end of while
        System.out.println("Goodbye -- End of Order!");
    } // end of main

    public static void printCupOrConeOptions() {
        System.out.println("Choose either a CUP or a CONE for your order: ");
        System.out.println("---------------------------------------------");
        System.out.println("\t1) CUP");
        System.out.println("\t2) CONE");
        System.out.println("---------------------------------------------");
        System.out.print("Enter your selection: ");
    }

    public static void printChoice(String flavor, int counter) {
        System.out.println("Ice Cream Choice #" + (counter + 1) + ": " + flavor);
        System.out.println("------------------------------------------------");
        iceCreamList[iceCreamCounter++] = flavor;
    }

    public static void printFlavorOptions() {
        System.out.println();
        System.out.println("Choose ice cream flavors:");
        System.out.println("-------------------------");
        for (int i = 0; i < options.length; i++) {
            System.out.println(" " + (i + 1) + ") " + options[i]);
        }
        System.out.println("------------------------------------------------");
    }

    public static boolean checkOrderCancellation() {
        int check = 0;
        System.out.println();
        System.out.println("Would you like Ice cream?");
        System.out.println("-------------------------");
        System.out.println("\t1) Yes, continue the order!");
        System.out.println("\t2) No, CANCEL my order");
        System.out.print("Enter your selection: ");
        check = input.nextInt();
        while ((check != 1) && (check != 2)) {
            System.out.print("Please enter a valid entry: ");
            check = input.nextInt();
        }
        System.out.print("You have chosen: ");
        if (check == 1) {
            System.out.println("CONTINUE Your Order");
            System.out.println();
            return true;
        } else {
            System.out.println("CANCEL Your Order");
            System.out.println();
            return false;
        }
    }

    public static void printSizeOptions() {
        System.out.println("Choose size for your ice cream: ");
        System.out.println("--------------------------------");
        System.out.println("\t1) SMALL -- 1 scoop");
        System.out.println("\t2) MEDIUM -- 2 scoops");
        System.out.println("\t3) LARGE -- 3 scoops");
        System.out.print("Enter your selection: ");
    }

} // End of ArrayMenuInput
```