

Week 8 Exercise 8

Telmen Enkhbold

San Fransico Bay University

CE480 - Java and Internet Application

Dr. Chang, Henry

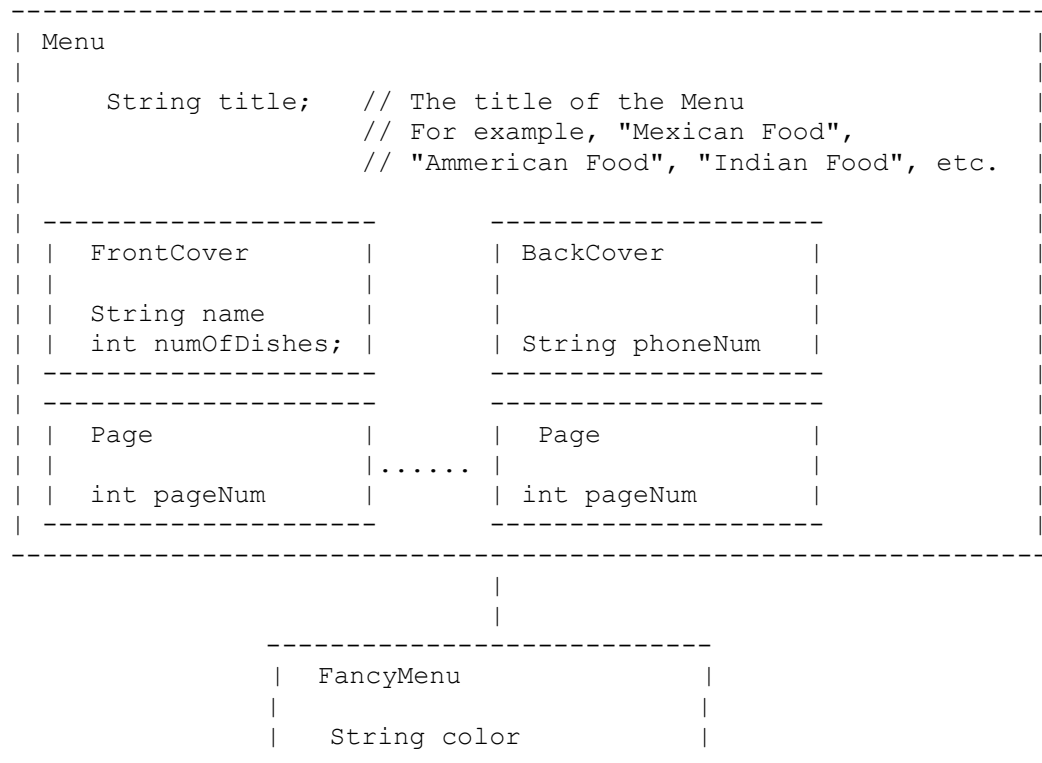
10/12/2023

Author Note

The Question

Restaurant (Vector + toString() + equals()) .

A big restaurant serves several kinds of food. For example, Japanese Food, Korean Food, Mexican food, Vietnamese Food, Indian food, Chinese food, American food, etc. Details of each kind of food is described by its corresponding menu. The front cover describes the name of the restaurant and the total number of dishes. The back cover describes the phone number to order to-go. The pages between the front cover and back cover list the page numbers. A menu is usually colorless. However, a fancy menu is printed with color.



Note:

- pageNum shows the number of each menu page For example, the first page's page number is 1, the second page's page number is 2, etc.
- The default value for numOfDishes is 10.
- The minimum features for FrontCover, BackCover, Menu and Page are:
 1. implements Cloneable.
 2. Constructor
 3. get and set.
 4. toString

```
Title: Chinese Food
Front Cover:
    name: Forbidden City
    numOfDishes: 123
Back Cover:
    phoneNum: 408-123-4356
Totoal number of pages: 3
    pageNum: 1
    pageNum: 2
    pageNum: 3
```

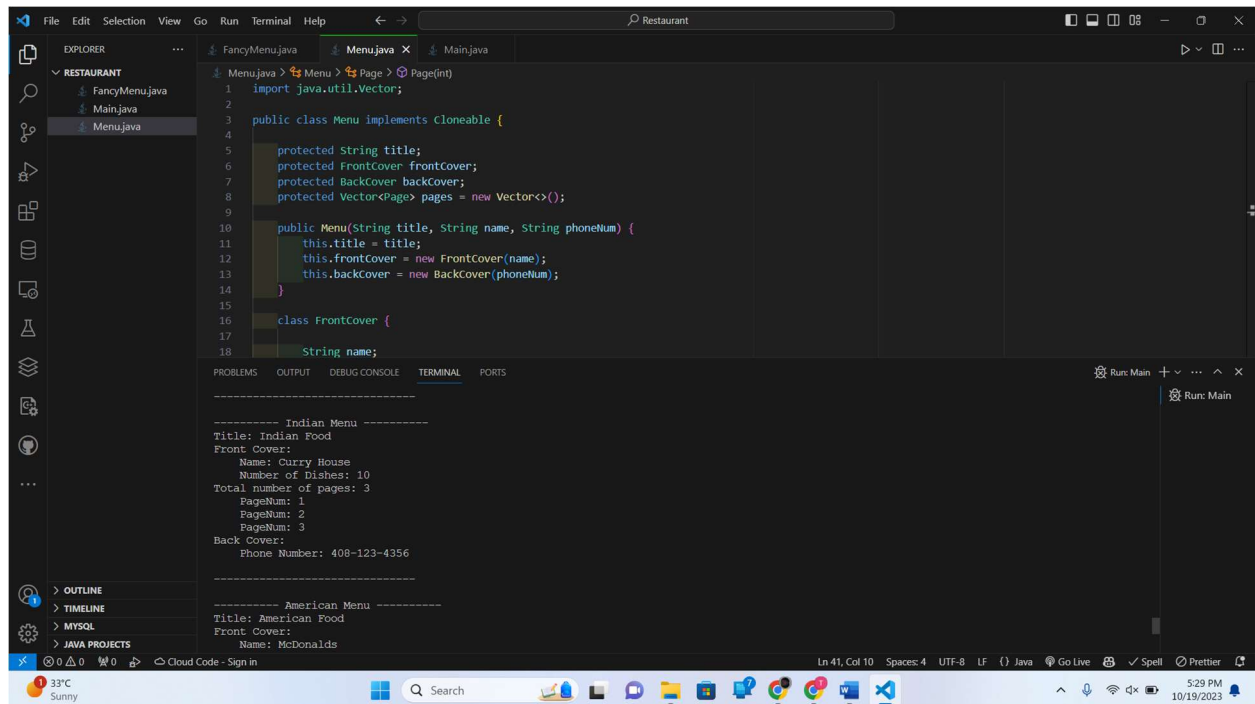
5. equals
- The minimum features for FanceyMenu are:
 1. Constructor
 2. get and set
 3. toString

Note: you can just print name-value pair of all the attributes of the FancyMenu. For example,

```
Title: Mexicon Food
Front Cover:
    name: Taco Bell
    numOfDishes: 46
Back Cover:
    phoneNum: 510-123-4567
Totoal number of pages: 4
    pageNum: 1
    pageNum: 2
    pageNum: 3
    pageNum: 4
Color: red
```

4. equals

- You should create a main function to test all your member functions of all the classes in your code.
- You should use Javadoc to document your program and post the document on Internet.
- References
 - [Class containing two layers of simple object attribute + toString\(\) + clone\(\) + equals\(\)](#)
 - [get/set](#)
 - [Inheritance Class Structure](#)
 - [4 Types of Member Functions](#)
 - [Summary of toString\(\), clone\(\), equals\(\)](#)
- Sample code `..`
 1. Vector can be used to implement array of pages?
 - Week 3, link ["class structure"](#)
 - Week 3, link ["Vector"](#)
 2. How to implement inheritance?
 - Week4, link ["Inheritance Class Structure"](#)
 3. String
 - Week3, link ["Strings"](#)
 4. toString
 - Week3, link ["toString\(\) methods \(operator<<\)"](#)
 5. equals
 - Week3, link ["The equals\(\) method \(operator==\)"](#)
 6. clone
 - Week3, link ["Clone"](#)
 7. Base Class vs. Subclass
 - [Case 7](#)

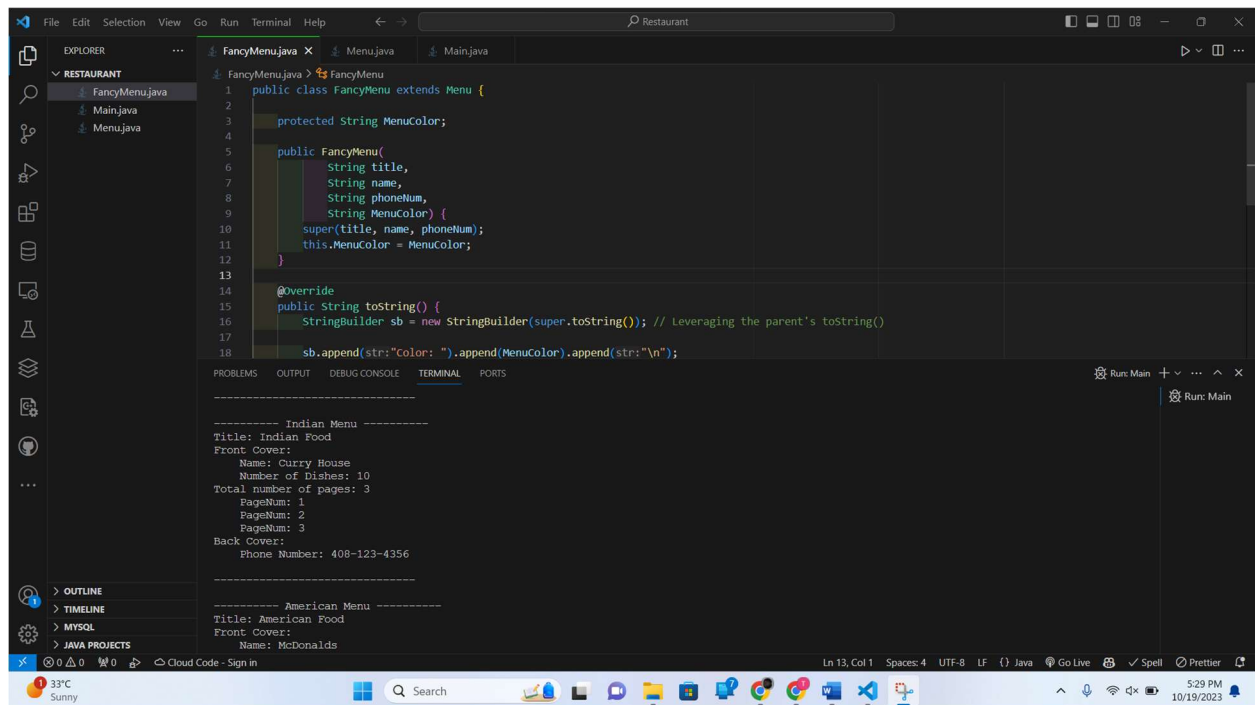


The screenshot shows an IDE window titled "Restaurant" with the "Menu.java" file open. The code defines a `Menu` class that implements `Cloneable` and contains a `FrontCover` class. The `Menu` class has attributes `title`, `frontCover`, `backCover`, and `pages` (a `Vector<Page>`). It has a constructor and a `FrontCover` class with a `name` attribute. The terminal output shows the results of running the program, displaying details for "Indian Menu" and "American Menu".

```
1 import java.util.Vector;
2
3 public class Menu implements Cloneable {
4
5     protected String title;
6     protected FrontCover frontCover;
7     protected BackCover backCover;
8     protected Vector<Page> pages = new Vector<>();
9
10    public Menu(String title, String name, String phoneNum) {
11        this.title = title;
12        this.frontCover = new FrontCover(name);
13        this.backCover = new BackCover(phoneNum);
14    }
15
16    class FrontCover {
17
18        String name;
```

Terminal Output:

```
----- Indian Menu -----
Title: Indian Food
Front Cover:
  Name: Curry House
  Number of Dishes: 10
Total number of pages: 3
  PageNum: 1
  PageNum: 2
  PageNum: 3
Back Cover:
  Phone Number: 408-123-4356
-----
----- American Menu -----
Title: American Food
Front Cover:
  Name: McDonalds
```

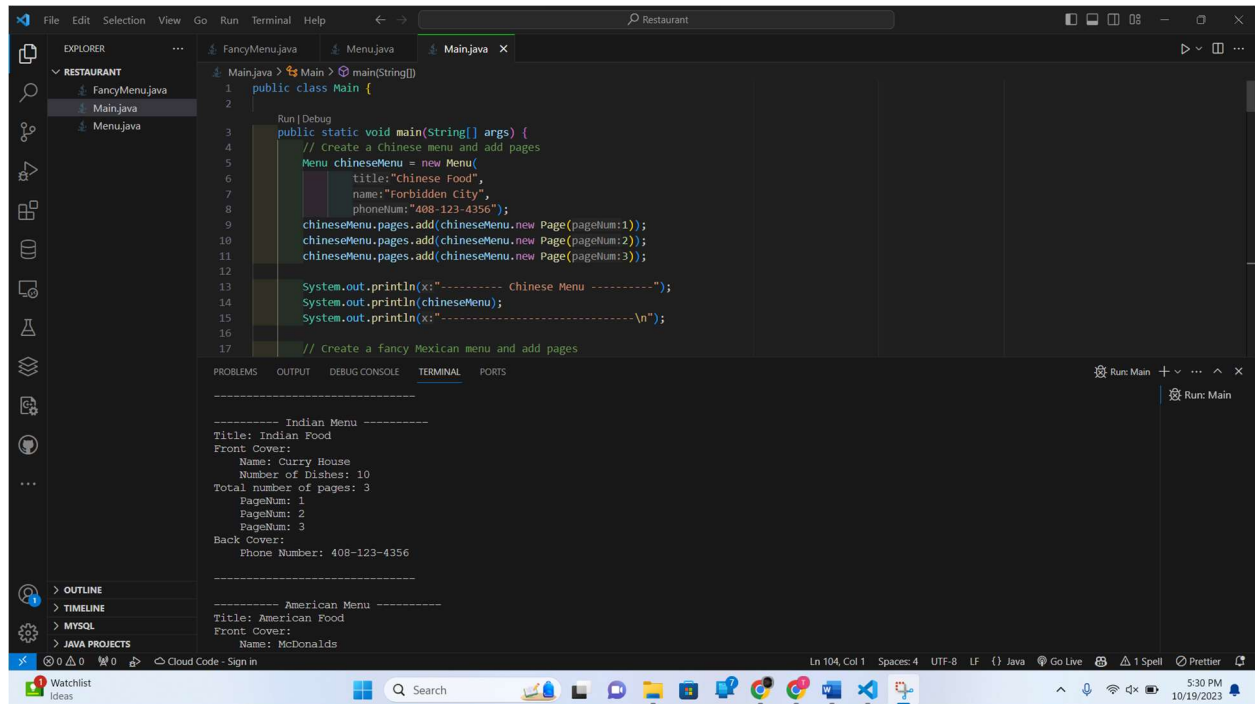


The screenshot shows the same IDE window with the "FancyMenu.java" file open. The code defines a `FancyMenu` class that extends `Menu` and adds a `MenuColor` attribute. It overrides the `toString` method to include the menu color. The terminal output is identical to the previous screenshot, showing details for "Indian Menu" and "American Menu".

```
1 public class FancyMenu extends Menu {
2
3     protected String MenuColor;
4
5     public FancyMenu(
6         String title,
7         String name,
8         String phoneNum,
9         String MenuColor) {
10        super(title, name, phoneNum);
11        this.MenuColor = MenuColor;
12    }
13
14    @Override
15    public String toString() {
16        StringBuilder sb = new StringBuilder(super.toString()); // Leveraging the parent's toString()
17        sb.append(str:"color: ").append(MenuColor).append(str:"\n");
18    }
19 }
```

Terminal Output:

```
----- Indian Menu -----
Title: Indian Food
Front Cover:
  Name: Curry House
  Number of Dishes: 10
Total number of pages: 3
  PageNum: 1
  PageNum: 2
  PageNum: 3
Back Cover:
  Phone Number: 408-123-4356
-----
----- American Menu -----
Title: American Food
Front Cover:
  Name: McDonalds
```



Source Code

Main.java

```
public class Main {  
  
    public static void main(String[] args) {  
        // Create a Chinese menu and add pages  
        Menu chineseMenu = new Menu(  
            "Chinese Food",  
            "Forbidden City",  
            "408-123-4356");  
        chineseMenu.pages.add(chineseMenu.new Page(1));  
        chineseMenu.pages.add(chineseMenu.new Page(2));  
        chineseMenu.pages.add(chineseMenu.new Page(3));  
  
        System.out.println("----- Chinese Menu -----");  
        System.out.println(chineseMenu);  
        System.out.println("-----\n");  
  
        // Create a fancy Mexican menu and add pages  
        FancyMenu mexicanMenu = new FancyMenu(  
            "Mexican Food",  
            "McDonalds",  
            "408-123-4356");  
        mexicanMenu.pages.add(mexicanMenu.new Page(1));  
        mexicanMenu.pages.add(mexicanMenu.new Page(2));  
        mexicanMenu.pages.add(mexicanMenu.new Page(3));  
  
        System.out.println("----- Mexican Menu -----");  
        System.out.println(mexicanMenu);  
        System.out.println("-----\n");  
    }  
}
```

```
        "Mexican Food",
        "Taco Bell",
        "510-123-4567",
        "red");
mexicanMenu.pages.add(mexicanMenu.new Page(1));
mexicanMenu.pages.add(mexicanMenu.new Page(2));
mexicanMenu.pages.add(mexicanMenu.new Page(3));
mexicanMenu.pages.add(mexicanMenu.new Page(4));

// Print Fancy Mexican menu details
System.out.println("----- Fancy Mexican Menu -----");
System.out.println(mexicanMenu);
System.out.println("-----");
// Japanese menu
Menu japaneseMenu = new Menu(
    "Japanese Food",
    "Sushi Joe",
    "408-123-4356");
japaneseMenu.pages.add(japaneseMenu.new Page(1));
japaneseMenu.pages.add(japaneseMenu.new Page(2));
japaneseMenu.pages.add(japaneseMenu.new Page(3));

System.out.println("----- Japanese Menu -----");
System.out.println(japaneseMenu);
System.out.println("-----\n");

// Korean menu
Menu koreanMenu = new Menu(
    "Korean Food",
    "Korean BBQ",
    "408-123-4356");
koreanMenu.pages.add(koreanMenu.new Page(1));
koreanMenu.pages.add(koreanMenu.new Page(2));
koreanMenu.pages.add(koreanMenu.new Page(3));

System.out.println("----- Korean Menu -----");
System.out.println(koreanMenu);
System.out.println("-----\n");
// Vietnamese menu
Menu vietnameseMenu = new Menu(
    "Vietnamese Food",
    "Pho King",
    "408-123-4356");
vietnameseMenu.pages.add(vietnameseMenu.new Page(1));
vietnameseMenu.pages.add(vietnameseMenu.new Page(2));
```

```
vietnameseMenu.pages.add(vietnameseMenu.new Page(3));

System.out.println("----- Vietnamese Menu -----");
System.out.println(vietnameseMenu);
System.out.println("-----\n");
// Indian menu
Menu indianMenu = new Menu(
    "Indian Food",
    "Curry House",
    "408-123-4356");
indianMenu.pages.add(indianMenu.new Page(1));
indianMenu.pages.add(indianMenu.new Page(2));
indianMenu.pages.add(indianMenu.new Page(3));

System.out.println("----- Indian Menu -----");
System.out.println(indianMenu);
System.out.println("-----\n");

// American menu
Menu americanMenu = new Menu(
    "American Food",
    "McDonalds",
    "408-123-4356");
americanMenu.pages.add(americanMenu.new Page(1));
americanMenu.pages.add(americanMenu.new Page(2));
americanMenu.pages.add(americanMenu.new Page(3));

System.out.println("----- American Menu -----");
System.out.println(americanMenu);
System.out.println("-----\n");

// Italian menu

Menu italianMenu = new Menu(
    "Italian Food",
    "Olive Garden",
    "408-123-4356");
italianMenu.pages.add(italianMenu.new Page(1));
italianMenu.pages.add(italianMenu.new Page(2));
italianMenu.pages.add(italianMenu.new Page(3));

System.out.println("----- Italian Menu -----");
System.out.println(italianMenu);
System.out.println("-----\n");
```



```

// French menu
FancyMenu frenchMenu = new FancyMenu(
    "French Food",
    "Le Bistro",
    "408-123-4356",
    "blue");
frenchMenu.pages.add(frenchMenu.new Page(1));
frenchMenu.pages.add(frenchMenu.new Page(2));
frenchMenu.pages.add(frenchMenu.new Page(3));
frenchMenu.pages.add(frenchMenu.new Page(4));

// Print Fancy Mexican menu details
System.out.println("----- Fancy French Menu -----");
System.out.println(frenchMenu);
System.out.println("-----");

// Thai menu
Menu thaiMenu = new Menu(
    "Thai Food",
    "Thai Express",
    "408-123-4356");
thaiMenu.pages.add(thaiMenu.new Page(1));
thaiMenu.pages.add(thaiMenu.new Page(2));
thaiMenu.pages.add(thaiMenu.new Page(3));

System.out.println("----- Thai Menu -----");
System.out.println(thaiMenu);
System.out.println("-----\n");
}
}

```

Menu.java

```

import java.util.Vector;

public class Menu implements Cloneable {

    protected String title;
    protected FrontCover frontCover;
    protected BackCover backCover;
    protected Vector<Page> pages = new Vector<>();
}

```

```
public Menu(String title, String name, String phoneNum) {
    this.title = title;
    this.frontCover = new FrontCover(name);
    this.backCover = new BackCover(phoneNum);
}

class FrontCover {

    String name;
    int numOfDishes = 10;

    FrontCover(String name) {
        this.name = name;
    }
}

class BackCover {

    String phoneNum;

    BackCover(String phoneNum) {
        this.phoneNum = phoneNum;
    }
}

class Page {

    int pageNum;

    Page(int pageNum) {
        this.pageNum = pageNum;
    }
}

@Override
public String toString() {
    StringBuilder sb = new StringBuilder();

    sb.append("Title: ").append(title).append("\n");
    sb.append("Front Cover:\n");
    sb.append("    Name: ").append(frontCover.name).append("\n");
    sb
        .append("    Number of Dishes: ")
        .append(frontCover.numOfDishes)
```

```

        .append("\n");

        sb.append("Total number of pages:
").append(pages.size()).append("\n");

        for (Page page : pages) {
            sb.append("    PageNum: ").append(page.pageNum).append("\n");
        }
        sb.append("Back Cover:\n");
        sb.append("    Phone Number:
").append(backCover.phoneNum).append("\n");

        return sb.toString();
    }
}

```

FancyMenu.java

```

public class FancyMenu extends Menu {

    protected String MenuColor;

    public FancyMenu(
        String title,
        String name,
        String phoneNum,
        String MenuColor) {
        super(title, name, phoneNum);
        this.MenuColor = MenuColor;
    }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder(super.toString()); //
        Leveraging the parent's toString()

        sb.append("Color: ").append(MenuColor).append("\n");

        return sb.toString();
    }
}

```

Github - https://github.com/Georgycas/CE350-Java-and-Internet-Applications/tree/main/Code/Lab_Code/Week8/Restaurant