

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG MỞ RỘNG

NHÓM L04 - ĐỀ 10:

**TÌM HIỂU CÔNG DỤNG VÀ LƯỢC ĐỒ CLASS CỦA
MẪU THIẾT KẾ BUILDER
ÁP DỤNG MẪU NÀY VÀO XÂY DỰNG ỨNG DỤNG.**

GVHD: Nguyễn Văn Hiệp
SV: Võ Hoàng Hải Nam MSSV: 1810340
Lê Thành Lâm MSSV: 1810730

TP. HỒ CHÍ MINH, THÁNG 12/2019



Mục lục

1	Mẫu thiết kế Builder	2
1.1	Giới thiệu mẫu Builder	2
1.2	Lược đồ class	2
2	Ứng dụng Quản lý nhà hàng KFC sử dụng mẫu thiết kế Builder	3
2.1	Giới thiệu ứng dụng	3
2.2	Ứng dụng của mẫu Builder trong thiết kế ứng dụng	7
2.3	Code ứng dụng	8
2.3.1	Code BillConcreteBuilder	8
2.3.2	Code BillBuilder	10
2.3.3	Code Bill	11
2.3.4	Code Form1	12
3	Kết luận	17
4	Tài liệu tham khảo	19

1 Mẫu thiết kế Builder

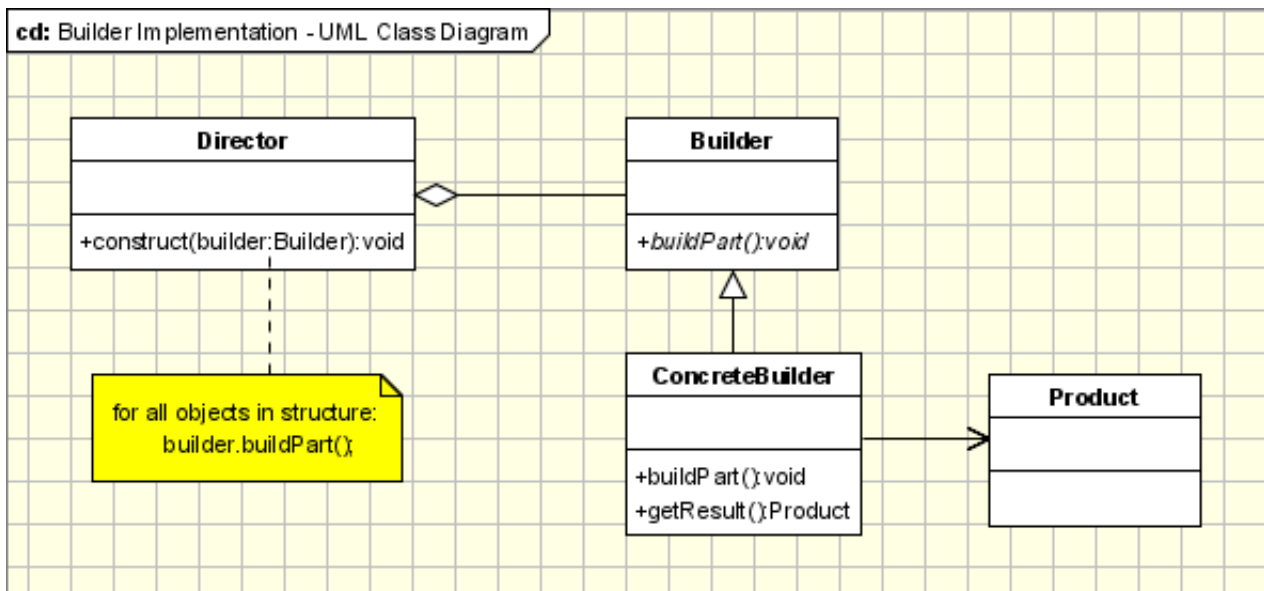
1.1 Giới thiệu mẫu Builder

Builder pattern là một mẫu thiết kế thuộc "Nhóm khởi tạo" (Creational Pattern). Mẫu thiết kế này cho phép lập trình viên tạo ra những đối tượng phức tạp nhưng chỉ cần thông qua các câu lệnh đơn giản để tác động nên các thuộc tính của nó.

Ta sử dụng Builder pattern khi:

- Ta muốn thay đổi thiết kế cho việc lồng nhau của các hàm khởi tạo (Telescoping Constructor Pattern). Vấn đề này phát sinh khi ta làm việc với một lớp mà có chứa rất nhiều các thuộc tính và cần phải tạo ra nhiều hàm khởi tạo với số lượng các thuộc tính tăng dần.
- Ta cần tạo ra một đối tượng phức tạp, một đối tượng mà thuật toán để tạo tạo lập các thuộc tính là độc lập đối với các thuộc tính khác.

1.2 Lược đồ class



Hình 1.1: Lược đồ mẫu Builder

- **Builder** : là thành phần định nghĩa một lớp trừu tượng (abstract class) để tạo ra một hoặc nhiều phần của đối tượng Product.

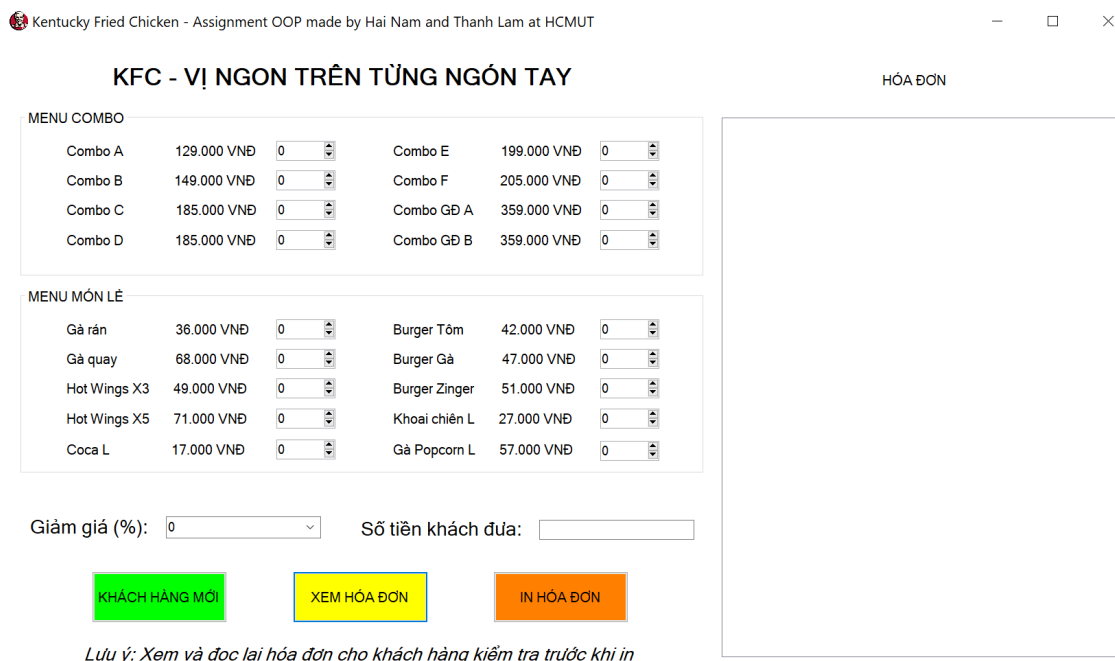
- **Concrete Builder** : là thành phần triển khai, cụ thể hóa các lớp trừu tượng cho để tạo ra các thành phần và tập hợp các thành phần đó với nhau. thành phần này sẽ xác định và nắm giữ các thể hiện mà nó tạo ra. Đồng thời nó cũng cung cấp phương thức để trả các thể hiện mà nó đã tạo ra trước đó.
- **Product** : thành phần này trong bài viết sẽ đại diện cho đối tượng phức tạp phải tạo ra.
- **Director** : thành phần này sẽ khởi tạo đối tượng Builder

2 Ứng dụng Quản lý nhà hàng KFC sử dụng mẫu thiết kế Builder

2.1 Giới thiệu ứng dụng

Đây là ứng dụng để thực hiện việc thanh toán hóa đơn trong của hàng KFC.

Trên giao diện ứng dụng trực quan, nhân viên sẽ nhập vào số lượng các món ăn mà khách hàng đã chọn sau đó nhấp để in hóa đơn.



Kentucky Fried Chicken - Assignment OOP made by Hai Nam and Thanh Lam at HCMUT

KFC - VỊ NGON TRÊN TỪNG NGÓN TAY

HÓA ĐƠN

MENU COMBO					
Combo A	129.000 VNĐ	0	Combo E	199.000 VNĐ	0
Combo B	149.000 VNĐ	0	Combo F	205.000 VNĐ	0
Combo C	185.000 VNĐ	0	Combo GĐ A	359.000 VNĐ	0
Combo D	185.000 VNĐ	0	Combo GĐ B	359.000 VNĐ	0

MENU MÓN LẺ					
Gà rán	36.000 VNĐ	0	Burger Tôm	42.000 VNĐ	0
Gà quay	68.000 VNĐ	0	Burger Gà	47.000 VNĐ	0
Hot Wings X3	49.000 VNĐ	0	Burger Zinger	51.000 VNĐ	0
Hot Wings X5	71.000 VNĐ	0	Khoai chiên L	27.000 VNĐ	0
Coca L	17.000 VNĐ	0	Gà Popcorn L	57.000 VNĐ	0

Giảm giá (%): 0 Số tiền khách đưa:

KHÁCH HÀNG MỚI **XEM HÓA ĐƠN** **IN HÓA ĐƠN**

Lưu ý: Xem và đọc lại hóa đơn cho khách hàng kiểm tra trước khi in

Hình 2.1: Giao diện ứng dụng khi mở

Giới thiệu các chức năng cơ bản:

- Các group MEUNU COMBO - MENU MÓN LẺ: Phân loại các món trong menu để người dùng dễ nhập số lượng
- Các nút tăng giảm: Để chọn số lượng món bằng cách ấn tăng, giảm hoặc nhập trực tiếp số lượng vào ô
- Giảm giá: Lựa chọn các mức giảm giá nếu có (mặc định là 0)
- Số tiền khách đưa: Để nhập số tiền khách hàng đưa, đơn vị Đồng
- Xem hóa đơn: Xem hóa đơn để xem tổng tiền và kiểm tra lại các món hàng đã mua
- In hóa đơn: Kết nối với thiết bị để in hóa đơn. Chỉ in khi số tiền đủ thanh toán và số món hàng lớn hơn 0.
- Khách hàng mới: Xóa dữ liệu của hóa đơn cũ và thiết lập app về trạng thái ban đầu

Kentucky Fried Chicken - Assignment OOP made by Hai Nam and Thanh Lam at HCMUT

KFC - VỊ NGON TRÊN TỪNG NGÓN TAY

MENU COMBO

Combo A	129.000 VNĐ	0	Combo E	199.000 VNĐ	0
Combo B	149.000 VNĐ	0	Combo F	205.000 VNĐ	1
Combo C	185.000 VNĐ	0	Combo GB A	359.000 VNĐ	0
Combo D	185.000 VNĐ	0	Combo GB B	359.000 VNĐ	0

MENU MÓN LẺ

Gà rán	36.000 VNĐ	1	Burger Tôm	42.000 VNĐ	1
Gà quay	68.000 VNĐ	0	Burger Gà	47.000 VNĐ	0
Hot Wings X3	49.000 VNĐ	0	Burger Zinger	51.000 VNĐ	1
Hot Wings X5	71.000 VNĐ	0	Khoai chiên L	27.000 VNĐ	0
Coca L	17.000 VNĐ	0	Gà Popcorn L	57.000 VNĐ	0

Giảm giá (%): Số tiền khách đưa:

1,000,000 VNĐ

KHÁCH HÀNG MỚI

XEM HÓA ĐƠN

IN HÓA ĐƠN

Lưu ý: Xem và đọc lại hóa đơn cho khách hàng kiểm tra trước khi in

HÓA ĐƠN

CUA HÀNG KFC ĐẠI HỌC BACH KHOA

Thời gian lập hóa đơn: 1/2/2020 3:05:37 PM

Tên món ăn	Số lượng	Thành tiền
GaRan	1	36,000
ComboF	1	205,000
BurgerTom	1	42,000
BurgerZinger	1	51,000
Tổng hóa đơn:		334,000
Khuyến mãi:		10%
Phải thanh toán:		300,600
Khách hàng đưa:		1,000,000
Thối lại:		699,400

Hình 2.2: Giao diện khi xem hóa đơn

Hóa đơn gồm các thông tin:

- Tên chi nhánh cửa hàng
- Thời gian lập hóa đơn

- Tên món ăn, Số lượng, Thành tiền mỗi món
- Tổng tiền hóa đơn, Khuyến mãi, Tiền phải thanh toán, Tiền khách hàng đưa, Tiền thối lại

Kentucky Fried Chicken - Assignment OOP made by Hai Nam and Thanh Lam at HCMUT

KFC - VỊ NGON TRÊN TỪNG NGÓN TAY

MENU COMBO

Combo A	129.000 VNĐ	0	Combo E	199.000 VNĐ	0
Combo B	149.000 VNĐ	0	Combo F	205.000 VNĐ	1
Combo C	185.000 VNĐ	0	Combo GĐ A	359.000 VNĐ	0
Combo D	185.000 VNĐ	0	Combo GĐ B	359.000 VNĐ	0

MENU MÓN LẺ

Gà rán	36.000 VNĐ	1	Burger Tôm	42.000 VNĐ	1
Gà quay	68.000 VNĐ	0	Burger Gà	47.000 VNĐ	0
Hot Wings X3	49.000 VNĐ	0	Burger Zinger	51.000 VNĐ	0
Hot Wings X5	71.000 VNĐ	0	Khoai chiên L	27.000 VNĐ	0
Coca L	17.000 VNĐ	0	Gà Popcorn L	57.000 VNĐ	0

Giảm giá (%): 10 Số tiền khách đưa:

KHÁCH HÀNG MỚI **XEM HÓA ĐƠN** **IN HÓA ĐƠN**

Lưu ý: Xem và đọc lại hóa đơn cho khách hàng kiểm tra trước khi in

HÓA ĐƠN

CUA HÀNG KFC ĐẠI HỌC BÁCH KHOA

Thời gian lập hóa đơn: 1/2/2020 3:06:09 PM

Tên món ăn	Số lượng	Thành tiền
Gà Rán	1	36,000
Combo F	1	205,000
Burger Tom	1	42,000
Burger Zinger	1	51,000
Tổng hóa đơn:		334,000
Khuyến mãi:		10%
Phải thanh toán:		300,600
Khách hàng đưa:		1,000,000
Thối lại:		699,400

In thành công

Hình 2.3: Giao diện khi in hóa đơn



Kentucky Fried Chicken - Assignment OOP made by Hai Nam and Thanh Lam at HCMUT

KFC - VỊ NGON TRÊN TỪNG NGÓN TAY

MENU COMBO

Combo A	129.000 VNĐ	0	Combo E	199.000 VNĐ	0
Combo B	149.000 VNĐ	0	Combo F	205.000 VNĐ	1
Combo C	185.000 VNĐ	0	Combo GD A	359.000 VNĐ	0
Combo D	185.000 VNĐ	0	Combo GD B	359.000 VNĐ	0

MENU MÓN LẺ

Gà rán	36.000 VNĐ	1	Burger Tôm	42.000 VNĐ	1
Gà quay	68.000 VNĐ	0	Burger Gà	47.000 VNĐ	0
Hot Wings X3	49.000 VNĐ	0	Burger Zinger	51.000 VNĐ	0
Hot Wings X5	71.000 VNĐ	0	Khoai chiên L	27.000 VNĐ	0
Coca L	17.000 VNĐ	0	Gà Popcorn L	57.000 VNĐ	0

HÓA ĐƠN

CUA HÀNG KFC ĐẠI HỌC BACH KHOA

Thời gian lập hóa đơn: 1/2/2020 3:06:40 PM

Tên món ăn	Số lượng	Thành tiền
Gà Rán	1	36,000
Combo F	1	205,000
Burger Tôm	1	42,000
Burger Zinger	1	51,000

Tổng hóa đơn: 334,000
Khuyến mãi: 10%
Phải thanh toán: 300,600
Khách hàng đưa: 100,000
Thối lại: -200,600

Giảm giá (%): 10 Số tiền khách đưa:

KHÁCH HÀNG MỚI **XEM HÓA ĐƠN** **IN HÓA ĐƠN**

Lưu ý: Xem và đọc lại hóa đơn cho khách hàng kiểm tra trước khi in

Không đủ thanh toán

OK

Hình 2.4: Khi tiền không đủ thanh toán hóa đơn

Kentucky Fried Chicken - Assignment OOP made by Hai Nam and Thanh Lam at HCMUT

KFC - VỊ NGON TRÊN TỪNG NGÓN TAY

MENU COMBO

Combo A	129.000 VNĐ	0	Combo E	199.000 VNĐ	0
Combo B	149.000 VNĐ	0	Combo F	205.000 VNĐ	0
Combo C	185.000 VNĐ	0	Combo GD A	359.000 VNĐ	0
Combo D	185.000 VNĐ	0	Combo GD B	359.000 VNĐ	0

MENU MÓN LẺ

Gà rán	36.000 VNĐ	0	Burger Tôm	42.000 VNĐ	0
Gà quay	68.000 VNĐ	0	Burger Gà	47.000 VNĐ	0
Hot Wings X3	49.000 VNĐ	0	Burger Zinger	51.000 VNĐ	0
Hot Wings X5	71.000 VNĐ	0	Khoai chiên L	27.000 VNĐ	0
Coca L	17.000 VNĐ	0	Gà Popcorn L	57.000 VNĐ	0

HÓA ĐƠN

Giảm giá (%): 0 Số tiền khách đưa:

KHÁCH HÀNG MỚI **XEM HÓA ĐƠN** **IN HÓA ĐƠN**

Lưu ý: Xem và đọc lại hóa đơn cho khách hàng kiểm tra trước khi in

Vui lòng chọn món

OK

Hình 2.5: Khi chưa chọn món mà in hóa đơn

Kentucky Fried Chicken - Assignment OOP made by Hải Nam and Thanh Lam at HCMUT

KFC - VỊ NGON TRÊN TỪNG NGÓN TAY

HÓA ĐƠN

MENU COMBO

Combo A

129.000 VNĐ

0

Combo B

149.000 VNĐ

0

Combo C

185.000 VNĐ

0

Combo D

185.000 VNĐ

0

Combo E

199.000 VNĐ

0

Combo F

205.000 VNĐ

0

Combo GD A

359.000 VNĐ

0

Combo GD B

359.000 VNĐ

0

MENU MÓN LẺ

Gà rán

36.000 VNĐ

0

Gà quay

68.000 VNĐ

0

Hot Wings X3

49.000 VNĐ

0

Hot Wings X5

71.000 VNĐ

0

Coca L

17.000 VNĐ

0

Burger Tôm

42.000 VNĐ

0

Burger Gà

47.000 VNĐ

0

Burger Zinger

51.000 VNĐ

0

Khoai chiên L

27.000 VNĐ

0

Gà Popcorn L

57.000 VNĐ

0

Giảm giá (%): 0

Số tiền khách đưa: 0 VNĐ

KHÁCH HÀNG MỚI

XEM HÓA ĐƠN

IN HÓA ĐƠN

Lưu ý: Xem và đọc lại hóa đơn cho khách hàng kiểm tra trước khi in

Hình 2.6: Khi chọn khách hàng mới

2.2 Ứng dụng của mẫu Builder trong thiết kế ứng dụng

Để xây dựng một hóa đơn tính tiền, ta cần phải có thông tin các món mà khách hàng đã chọn (ở đây số lượng phần ăn của các món ăn chính là các thuộc tính), phần trăm giảm giá và số tiền khách hàng đã đưa, bởi vì số lượng thuộc tính nêu trên lớn và có những thuộc tính không bắt buộc (optional), bên cạnh đó việc xây dựng các thuộc tính này cũng cần độc lập với nhau (vd: việc thay đổi số phần ăn combo A không ảnh hưởng đến số phần ăn combo B). Bởi vì những lý do trên cho nên nhóm chúng em đã chọn mẫu thiết kế builder để thiết kế cấu trúc cho ứng dụng này.

Để sử dụng mẫu thiết kế này chúng em đã hiện thực 3 class chính, đó là:

- **Product:** class `Bill` bao gồm các thông tin cơ bản của hóa đơn, đó là số lượng khẩu phần ăn của từng món, phần trăm giảm giá và số tiền khách trả, ngoài ra còn có một biến lưu trữ nội dung để in ra của hóa đơn. Chỉ có một hàm constructor với 4 tham số đã nêu trên.
- **Builder:** abstract class `BillBuilder` bao gồm các thành phần như trong class `Bill`, ngoài ra còn có thêm một số cờ kiểm tra, các hàm abstract để tạo lập các thành phần

và một hàm build() để trả về đối tượng đã xây dựng. Ở đây không hiện thực hàm constructor cho class BillBuilder cho nên khi tạo đối tượng BillBuilder giá trị của các thành phần đều có giá trị khởi tạo là 0.

- ConcreteBuilder: class ConcreteBuilder kế thừa class BillBuilder hiện thực các hàm có trong class BillBuilder.

Về cách thức khởi tạo một đối tượng:

- Bước 1: Khởi tạo đối tượng PrintBill có đặc tính của class BillConcreteBuilder.
- Bước 2: Thông qua các thao tác với giao diện ứng dụng mà xây dựng các thành phần (buildPart) có trong PrintBill.
- Bước 3: Khi cần xem hóa đơn. Ta tạo một đối tượng BillDisplay thuộc class Bill nhận giá trị trả về khi gọi tác vụ build() từ PrintBill. Đối tượng BillDisplay chính là đối tượng mà ta cần khởi tạo.

Ngoài ra, đoạn code còn có thêm một số tính năng để xử lý các yêu cầu cần thiết khi xuất hóa đơn (vd: Khách hàng đã chọn món hay chưa? Số tiền khách hàng đưa có đủ để thanh toán? ...)

2.3 Code ứng dụng

2.3.1 Code BillConcreteBuilder

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace KFC_Menu
8 {
9     class BillConcreteBuilder : BillBuilder
10     {
11         public override BillBuilder set(Menu option, int val)
12         {
13             if (order.ContainsKey(option))
14                 order[option] = val;
15             else order.Add(option, val);
16             if (val == 0) order.Remove(option);
17             return this;
18         }
19     }
```

```
19     public override BillBuilder Discount(long discount)
20     {
21         this.discount = discount;
22         return this;
23     }
24
25     public override BillBuilder Pay(long pay)
26     {
27         this.pay = pay;
28         return this;
29     }
30
31     public override void Show()
32     {
33         long total = 0;
34         show += "\n";
35         show += "\n          ";
36         show += "CUA HANG KFC DAI HOC BACH KHOA \n";
37         show += "\n";
38         show += "Thoi gian lap hoa don:";
39         show += DateTime.Now + "\n";
40         show += "\n";
41         show += String.Format("{0,-15} | {1,-5} | {2,-12}",
42             "Ten mon an", "So luong", " Thanh tien");
43         show += "\n";
44         foreach (KeyValuePair<Menu, int> kvp in order)
45         {
46             long add = Convert.ToInt64(price[(int)kvp.Key] * kvp.
Value * 1000);
47             show += String.Format("{0,-20} {1,-5} {2,12}", kvp.Key
, kvp.Value, String.Format("{0:#,###,###.##}", add));
48             show += "\n";
49             total += add;
50         }
51         show += "\n-----\n";
52         show += String.Format("{0,-30} {1,10}", "Tong hoa don:",
String.Format("{0:#,###,###.##}", total)); show += "\n";
53         show += String.Format("{0,-30} {1,10}", "Khuyen mai:",
String.Format("{0}", discount)); show += "%\n";
54         long res = Convert.ToInt64(total * (100 - discount) / 100);
55         show += String.Format("{0,-30} {1,10}", "Phai thanh toan:",
String.Format("{0:#,###,###.##}", res)); show += "\n";
56         show += String.Format("{0,-30} {1,10}", "Khach hang dua:",
String.Format("{0:#,###,###.##}", pay)); show += "\n";
```

```
57         long left = pay - res;
58         show += String.Format("{0,-30} {1,10}", "Thoi lai:", String.
Format("{0:##,###,###.##}", left)); show += "\n";
59         if (total > 0) DaChonMon = true;
60         else DaChonMon = false;
61         if (left >= 0) DuThanhToan = true;
62         else DuThanhToan = false;
63     }
64     public override Bill build()
65     {
66         Show();
67         return new Bill(order, discount, pay, show);
68     }
69 }
70 }
```

Listing 1: BillConcreteBuilder.cs

2.3.2 Code BillBuilder

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace KFC_Menu
8 {
9     abstract class BillBuilder
10     {
11         protected Dictionary<Menu, int> order = new Dictionary<Menu, int>
>();
12
13         protected long pay;
14
15         protected long discount;
16
17         protected string show;
18
19         public bool DaChonMon = false;
20
21         public bool DuThanhToan = false;
22
23         protected int[] price = { 129, 149, 185, 185, 199, 205, 359,
359, 36, 68, 49, 71, 17, 42, 47, 51, 27, 57 };
```

```
24     public abstract BillBuilder set(Menu option, int val);
25     public abstract BillBuilder Discount(long discount);
26     public abstract BillBuilder Pay(long givenMoney);
27     public abstract void Show();
28     public abstract Bill build();
29 }
30 }
```

Listing 2: BillBuilder.cs

2.3.3 Code Bill

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace KFC_Menu
8 {
9     public enum Menu
10    {
11        ComboA, ComboB, ComboC, ComboD, ComboE, ComboF,
12        ComboGDA, ComboGDB,
13        GaRan,
14        GaQuay,
15        HotWingx3,
16        HotWingx5,
17        CocaL,
18        BurgerTom,
19        BugerGa,
20        BugerZinger,
21        KhoaiChienL,
22        GaPopcornL
23    }
24    class Bill
25    {
26        private Dictionary<Menu, int> order;
27        private long pay;
28        private long discount;
29        private string show;
30
31        public Bill(Dictionary<Menu, int> order, long discount, long
givenMoney, string show)
32        {
```

```
33         this.order = order;
34         this.discount = discount;
35         this.pay = givenMoney;
36         this.show = show;
37     }
38     public string toString()
39     {
40         return show;
41     }
42 }
43 }
```

Listing 3: Bill.cs

2.3.4 Code Form1

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.Text.RegularExpressions;
11
12 namespace KFC_Menu
13 {
14     public partial class Form1 : Form
15     {
16         BillBuilder PrintBill = new BillConcreteBuilder();
17         Bill BillDisplay;
18         public void ViewBill()
19         {
20             BillDisplay = PrintBill.build();
21             if (!PrintBill.DaChonMon) { MessageBox.Show("Vui long chon
mon"); goto E; }
22             lbBill.Items.Clear();
23             string BillContent = BillDisplay.toString();
24             foreach (string s in Regex.Split(BillContent, "\n"))
25                 lbBill.Items.Add(s);
26             E: { };
27         }
28     public Form1()
```

```
29     {
30         InitializeComponent();
31         comboBoxMaGiamGia.SelectedIndex = 0;
32         textBoxTienKhachDua.Text = "";
33     }
34
35     private void NumericUpDownComboA_ValueChanged(object sender,
EventArgs e)
36     {
37         PrintBill.set(KFC_Menu.Menu.ComboA, Decimal.ToInt32(
numericUpDownComboA.Value));
38     }
39
40     private void NumericUpDownComboB_ValueChanged(object sender,
EventArgs e)
41     {
42         PrintBill.set(KFC_Menu.Menu.ComboB, Decimal.ToInt32(
numericUpDownComboB.Value));
43     }
44
45     private void NumericUpDownComboDGA_ValueChanged(object sender,
EventArgs e)
46     {
47         PrintBill.set(KFC_Menu.Menu.ComboGDA, Decimal.ToInt32(
numericUpDownComboGDA.Value));
48     }
49     private void NumericUpDownComboC_ValueChanged(object sender,
EventArgs e)
50     {
51         PrintBill.set(KFC_Menu.Menu.ComboC, Decimal.ToInt32(
numericUpDownComboC.Value));
52     }
53
54     private void NumericUpDownComboD_ValueChanged(object sender,
EventArgs e)
55     {
56         PrintBill.set(KFC_Menu.Menu.ComboD, Decimal.ToInt32(
numericUpDownComboD.Value));
57     }
58
59     private void NumericUpDownComboE_ValueChanged(object sender,
EventArgs e)
60     {
61         PrintBill.set(KFC_Menu.Menu.ComboE, Decimal.ToInt32(
```

```
numericUpDownComboE.Value));
62     }
63
64     private void NumericUpDownComboF_ValueChanged(object sender,
EventArgs e)
65     {
66         PrintBill.set(KFC_Menu.Menu.ComboF, Decimal.ToInt32(
numericUpDownComboF.Value));
67     }
68
69     private void NumericUpDownComboGDB_ValueChanged(object sender,
EventArgs e)
70     {
71         PrintBill.set(KFC_Menu.Menu.ComboGDB, Decimal.ToInt32(
numericUpDownComboGDB.Value));
72     }
73
74     private void NumericUpDownGaRan_ValueChanged(object sender,
EventArgs e)
75     {
76         PrintBill.set(KFC_Menu.Menu.GaRan, Decimal.ToInt32(
numericUpDownGaRan.Value));
77     }
78
79     private void NumericUpDownGaQuay_ValueChanged(object sender,
EventArgs e)
80     {
81         PrintBill.set(KFC_Menu.Menu.GaQuay, Decimal.ToInt32(
numericUpDownGaQuay.Value));
82     }
83
84     private void NumericUpDownHotWingsX3_ValueChanged(object sender,
EventArgs e)
85     {
86         PrintBill.set(KFC_Menu.Menu.HotWingx3, Decimal.ToInt32(
numericUpDownHotWingsX3.Value));
87     }
88
89     private void NumericUpDownHotWingsX5_ValueChanged(object sender,
EventArgs e)
90     {
91         PrintBill.set(KFC_Menu.Menu.HotWingx5, Decimal.ToInt32(
numericUpDownHotWingsX5.Value));
92     }
```

```
93
94     private void NumericUpDownCocaL_ValueChanged(object sender,
EventArgs e)
95     {
96         PrintBill.set(KFC_Menu.Menu.CocaL, Decimal.ToInt32(
numericUpDownCocaL.Value));
97     }
98
99     private void NumericUpDownBgTom_ValueChanged(object sender,
EventArgs e)
100    {
101        PrintBill.set(KFC_Menu.Menu.BurgerTom, Decimal.ToInt32(
numericUpDownBgTom.Value));
102    }
103
104    private void NumericUpDownBgGa_ValueChanged(object sender,
EventArgs e)
105    {
106        PrintBill.set(KFC_Menu.Menu.BugerGa, Decimal.ToInt32(
numericUpDownBgGa.Value));
107    }
108
109    private void NumericUpDownBgZinger_ValueChanged(object sender,
EventArgs e)
110    {
111        PrintBill.set(KFC_Menu.Menu.BugerZinger, Decimal.ToInt32(
numericUpDownBgZinger.Value));
112    }
113
114    private void NumericUpDownKhoaiChienL_ValueChanged(object sender
, EventArgs e)
115    {
116        PrintBill.set(KFC_Menu.Menu.KhoaiChienL, Decimal.ToInt32(
numericUpDownKhoaiChienL.Value));
117    }
118
119    private void NumericUpDownGaPcL_ValueChanged(object sender,
EventArgs e)
120    {
121        PrintBill.set(KFC_Menu.Menu.GaPopcornL, Decimal.ToInt32(
numericUpDownGaPcL.Value));
122    }
123
124    private void ComboBoxMaGiamGia_SelectedIndexChanged(object
```



```
sender, EventArgs e)
{
    if (comboBoxMaGiamGia.SelectedItem != null)
    {
        long x = Convert.ToInt64(comboBoxMaGiamGia.SelectedItem.
ToString());
        PrintBill.Discount(x);
    }
    else PrintBill.Discount(0);
}

private void TextBoxTienKhachDua_TextChanged(object sender,
EventArgs e)
{
    if (textBoxTienKhachDua.Text != "")
    {
        PrintBill.Pay(Convert.ToInt64(textBoxTienKhachDua.Text))
;
        lblSoTien.Text = String.Format("{0:#,###,###.##}",
Convert.ToInt64(textBoxTienKhachDua.Text)) + " VND";
    }
    else
    {
        PrintBill.Pay(0);
        lblSoTien.Text = "0 VND";
    }
}

private void ButtonXem_Click(object sender, EventArgs e)
{
    ViewBill();
}

private void ButtonPrint_Click(object sender, EventArgs e)
{
    ViewBill();
    if (!PrintBill.DuThanhToan) MessageBox.Show("Khong du thanh
toan");
    else if (PrintBill.DaChonMon) MessageBox.Show("In thanh cong
");
}

private void ButtonRefresh_Click(object sender, EventArgs e)
{
    numericUpDownBgGa.Value = 0;
}
```

```
162         numericUpDownBgTom.Value = 0;
163         numericUpDownBgZinger.Value = 0;
164         numericUpDownCocaL.Value = 0;
165         numericUpDownComboA.Value = 0;
166         numericUpDownComboB.Value = 0;
167         numericUpDownComboC.Value = 0;
168         numericUpDownComboD.Value = 0;
169         numericUpDownComboE.Value = 0;
170         numericUpDownComboF.Value = 0;
171         numericUpDownComboGDA.Value = 0;
172         numericUpDownComboGDB.Value = 0;
173         numericUpDownGaPcL.Value = 0;
174         numericUpDownGaQuay.Value = 0;
175         numericUpDownGaRan.Value = 0;
176         numericUpDownHotWingsX3.Value = 0;
177         numericUpDownHotWingsX5.Value = 0;
178         numericUpDownKhoaiChienL.Value = 0;
179         comboBoxMaGiamGia.SelectedIndex = 0;
180         textBoxTienKhachDua.Text = "";
181         PrintBill.DaChonMon = false;
182         PrintBill.DuThanhToan = false;
183         lbBill.Items.Clear();
184     }
185
186     private void TextBoxTienKhachDua_KeyPress(object sender,
187     KeyPressEventArgs e)
188     {
189         if (!Char.IsDigit(e.KeyChar) && !Char.IsControl(e.KeyChar))
190             e.Handled = true;
191     }
192 }
```

Listing 4: Form1.cs

3 Kết luận

Nhìn chung, mẫu thiết kế Builder tương đối giống với mẫu Abstract Factory. Tuy nhiên mẫu thiết kế Abstract Factory bộc lộ nhược điểm rõ rệt khi Object cần tạo có quá nhiều thuộc tính. Chúng ta có thể khắc phục điều này với một số lượng lớn các tham số bằng việc cung cấp một hàm khởi tạo với những tham số bắt buộc và các method getter/setter để cài đặt các tham số tùy chọn. Vấn đề với hướng tiếp cận này là trạng thái của Object sẽ không nhất quán cho tới khi tất cả các thuộc tính được cài đặt một cách rõ ràng.

Nếu cần xây dựng một đối tượng cố định (không được thay đổi thuộc tính) thì cách này cũng không thể thực hiện được. Trong khi đó mẫu thiết kế Builder xử lý các vấn đề này bằng việc cung cấp một cách xây dựng đối tượng từng bước một và cung cấp một method để trả về đối tượng cuối cùng.

Tuy nhiên, mẫu thiết kế Builder cũng có nhược điểm là lượng code trùng lặp khá nhiều do cần phải copy tất cả các thuộc tính từ class Product sang class Builder do đó làm tăng độ phức tạp của code (tổng thể) do số lượng class tăng lên.

Thông qua việc nghiên cứu và hoàn thành đề tài “Tìm hiểu công dụng và lược đồ class của mẫu thiết kế Builder. Áp dụng mẫu này vào xây dựng ứng dụng”, nhóm chúng em đã có cơ hội tìm hiểu về nội dung và ứng dụng của mẫu thiết kế Builder cũng như phân tích ưu nhược điểm của mẫu thiết kế này so với các mẫu thiết kế khác. Bên cạnh đó chúng em cũng đã áp dụng mẫu thiết kế Builder để xây dựng một ứng dụng xuất hóa đơn trên nền tảng ngôn ngữ lập trình VC#.



4 Tài liệu tham khảo

Tài liệu

- [1] Đặng Quyết Tiến (2016). *Hiểu biết cơ bản về Builder pattern*. Truy cập tại <https://viblo.asia/p/hieu-biet-co-ban-ve-builder-pattern-5y8Rr7L9Mob3>.
- [2] Nguyễn Văn hiệp (2018). *Lập trình hướng đối tượng bằng VC#*. Nhà xuất bản Đại học Quốc gia TP Hồ Chí Minh.