Lab 3

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG ỨNG DỤNG WINDOWS FORMS

Một cửa hàng sách cần quản lý thông tin về các loại sách sẽ bán của mình. Một số thông tin cần quản lý là: tên sách, số lượng, giá bán. Hãy thiết kế lớp *sách bán* cho phép có những thuộc tính (có phạm vi truy cập là public) là các thông tin trên. Ngoài các thuộc tính lớp nên có phương thức tính tiền dựa vào số lượng sách và giá bán mỗi quyển.

Thí dụ: Quyển *Mật mã của Da Vinci* bán cho 1 khách hàng là *20* quyển, giá bán 1 quyển là 99000đ. Vậy số tiền phải trả là 1.980.000 đ.

1. Tạo ứng dụng loại Windows Application.

2. Thêm một lớp mới tên là clsBookSale. Lớp này có 03 dữ liệu thành viên: tên sách, số lượng, giá bán.

```
class clsBookSale
{
    string strTitle;
    int intQuantity;
    decimal decPrice;
}
```

3. Ứng với mỗi dữ liệu thành viên, một thuộc tính được tạo ra:

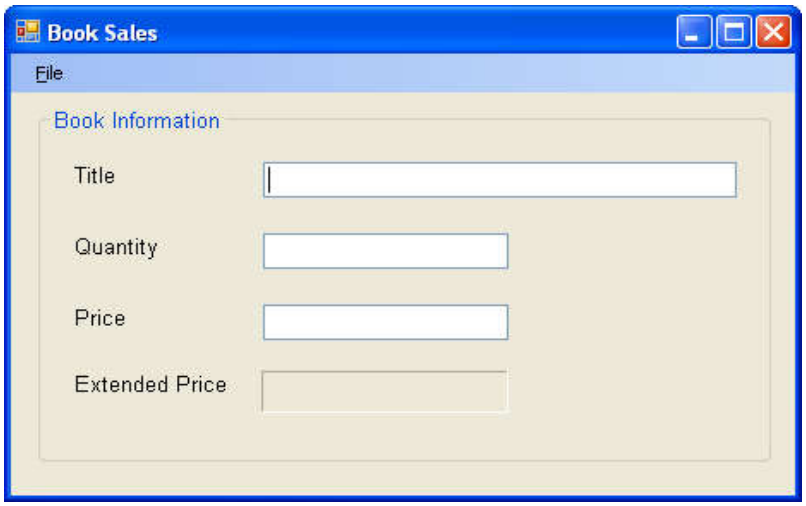
```
public string Title
{
    get
    {
        return strTitle;
    }
    set
    {
        strTitle=value;
    }
}

public int Quantity
{
    get
    {
        return intQuantity;
    }
    set
    {
        if (value>0)
            intQuantity = value;
        else throw new Exception();
    }
}
```

```
public decimal Price
{
    get
    {
        return decPrice;
    }
    set
    {
        if(value>=0)
            decPrice=value;
        else throw new Exception();
    }
}
```

4. Phương thức tính tiền cho phép tính tiền sau mỗi lần bán sách:

```
public decimal ExtendedPrice()
{
    decimal decExtendedPrice = intQuantity * decPrice;
    return decExtendedPrice;
}
```

5. Thiết kế form của ứng dụng:



Hình 3.1: Giao diện ban đầu

Các mục trên menu File như sau:

- New: cho phép xóa trắng các ô nhập.

- Calculate: Sau khi người dùng nhập các thông tin vào các ô Title, Quantity, Price; người dùng có thể chọn tính năng này để tính số tiền phải trả.

- Exit: chấm dứt ứng dụng.

6. Khai báo một đối tượng thuộc lớp clsBookSale trong phần khai báo của form ở hình 3.1:

```
public partial class frmBookSale : Form
{
    private clsBookSale mBookSale;
    public frmBookSale()
    {
        InitializeComponent();
    }
```

7. Sự kiện Click của mục menu mnuCalculate được xử lý:

```
private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsBookSale();
        mBookSale.Title = txtTitle.Text;
        mBookSale.Quality = Convert.ToInt32(txtQuan.Text);
        mBookSale.Price = Convert.ToDecimal(txtPrice.Text);

        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
    }
    catch(Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

8. Sự kiện Click của mục menu New cần xử lý để cho phép xóa trắng các ô nhập.

9. Sự kiện Click của mục menuExit cho phép chấm dứt chương trình. Để thực hiện được, ta sử dụng: **Application.Exit();**

10. Lưu dự án và thực thi ứng dụng

11. Giả sử, cần ghi nhận số lần bán sách trong ngày và tổng số tiền bán ra trong ngày; lúc đó lớp clsBookSale cần được bổ sung các thành viên để cho phép ghi nhận 02 giá trị này.

Giải pháp ở đây là thêm 02 dữ liệu thành viên dạng **static**; cứ mỗi lần bán sách, số lần tăng 1 và số tiền tăng lên bằng số tiền đã bán ở lần đó.

Thêm 02 dữ liệu thành viên **static** vào lớp clsBookSale và 02 thuộc tính tương ứng:

```
static decimal decSalesTotal=0;
static int intSalesCount=0;

public static decimal SalesTotal
{
    get
    {
        return decSalesTotal;
    }
}

public static int SalesCount
{
    get
    {
        return intSalesCount;
    }
}
```

Phương thức tính tiền được viết lại:

```
public decimal ExtendedPrice()
{
    decimal decExtendedPrice = intQuantity * decPrice;
    decSalesTotal += decExtendedPrice;
    intSalesCount += 1;

    return decExtendedPrice;
}
```

12. Menu File của form hình 3.1 được bổ sung mục Summary với sự kiện Click của mục này được xử lý:

```
private void mnuSum_Click(object sender, EventArgs e)
{
    string strMessage = "Sales Total: " +
       clsBookSale.SalesTotal.ToString("C") +
       "\nSales Count: " + clsBookSale.SalesCount.ToString();

    MessageBox.Show(strMessage, "Summary Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

13. Giả sử, cửa hàng sách cần giảm giá 10% cho một số khách hàng nào đó. Việc này có thể được thể hiện bằng 1 dữ liệu thành viên **bool** trong lớp clsBookSale:

```
const float sngDISCOUNT_RATE = 0.1F;

bool blnDiscount;
public bool Discount
{
    get
    {
        return blnDiscount;
    }
    set
    {
        blnDiscount=value;
    }
}
```

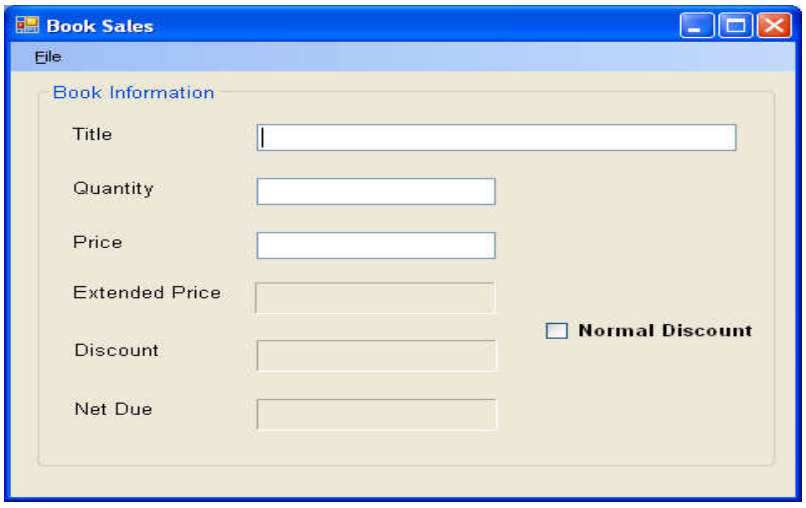
Lúc này, một phương thức tính số tiền giảm được định nghĩa thêm trong lớp clsBookSale:

```
public decimal DiscountAmount()
{
    decimal decDiscountAmount;
    if (blnDiscount)
        decDiscountAmount = intQuantity * decPrice *
            Convert.ToDecimal(sngDISCOUNT_RATE);
    else
        decDiscountAmount = 0M;

    return decDiscountAmount;
}
```

Phương thức NetDue() cho phép tính số tiền thực sự phải trả, số tiền này = Tổng số tiền – Số tiền giảm. Lưu ý là khi tính số tiền thực sự phải trả ta cần cập nhật lại các thuộc tính dùng chung (**static**) cho các đối tượng của lớp.

```
public decimal NetDue()
{
    decimal decNetDue = ExtendedPrice() - DiscountAmount();

    decSalesTotal += decNetDue;
    intSalesCount += 1;
    return decNetDue;
}

public decimal ExtendedPrice()
{
    decimal decExtendedPrice = intQuantity * decPrice;
    return decExtendedPrice;
}
```

14. Thêm một CheckBox vào form:

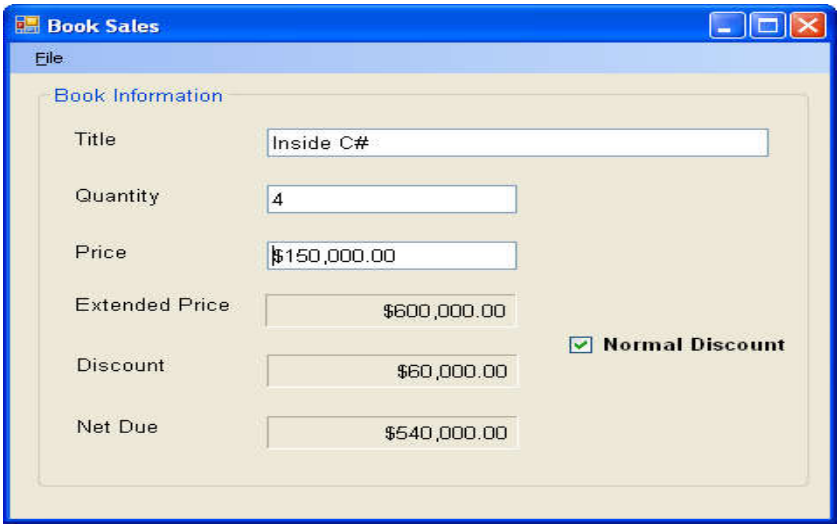


Hình 3.2: Form sau khi thêm CheckBox Normal Discount

15. Sự kiện mnuCalculate_Click được xử lý lại:

```
private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsBookSale();
        mBookSale.Title = txtTitle.Text;
        mBookSale.Quality = Convert.ToInt32(txtQuan.Text);
        mBookSale.Price = Convert.ToDecimal(txtPrice.Text);
        mBookSale.Discount=chkDis.Checked;

        txtPrice.Text = mBookSale.Price.ToString("C");
        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
        lblDis.Text = mBookSale.DiscountAmount().ToString("C");
        lblNetDue.Text = mBookSale.NetDue().ToString("C");
    }
    catch(Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

16. Một kết quả thực thi chương trình:



Hình 3.3: Một kết quả thực hiện chương trình

17. Cải tiến lớp clsBookSale bằng cách định nghĩa 1 phương thức xây dựng có tham số để gán trị cho các dữ liệu thành viên:

```
public clsBookSale(string title, int quality, decimal price, bool discount)
{
    this.Title=title;
    this.Quality=quality;
    this.Price=price;
    this.Discount=discount;
}
```

18. Sự kiện mnuCalculate_Click được viết lại:

```
private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsBookSale(txtTitle.Text,Convert.ToInt32(txtQuan.Text),
                                    Convert.ToDecimal(txtPrice.Text), chkDis.Checked);

        txtPrice.Text = mBookSale.Price.ToString("C");
        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
        lblDis.Text = mBookSale.DiscountAmount().ToString("C");
        lblNetDue.Text = mBookSale.NetDue().ToString("C");
    }
    catch(Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

19. Thực thi chương trình và kiểm tra lại kết quả.

20. Đầu mỗi năm học (tháng 8, 9); nhà sách bán giảm giá 15% cho sinh viên giỏi. Điều đó dẫn tới giao diện của chương trình được thiết kế lại. Bên cạnh đó, lớp clsBookSale cũng được sửa đổi lại. Tuy nhiên ta có thể định nghĩa lớp mới clsStudentBookSale thừa kế từ lớp clsBookSale. Lớp mới này có thêm thành viên mới kiểu bool là StudentDiscount (true: sinh viên giỏi).

Lớp clsStudentBookSale:

```
class clsStudentBookSale:clsBookSale
{
    bool blnStudentDiscount;
    const float sngSTUDENT_DISCOUNT_RATE = 0.15F;

    public bool StudentDiscount
    {
        get
        {
            return blnStudentDiscount;
        }
        set
        {
            blnStudentDiscount=value;
        }
    }
}
```

Lớp clsStudentBookSale có phương thức xây dựng gọi lại phương thức xây dựng của lớp clsBookSale:

```
public clsStudentBookSale(string title, int quality, decimal price,
                          bool discount, bool student)
    :base(title,quality,price,discount)

{
    blnStudentDiscount=student;
}
```

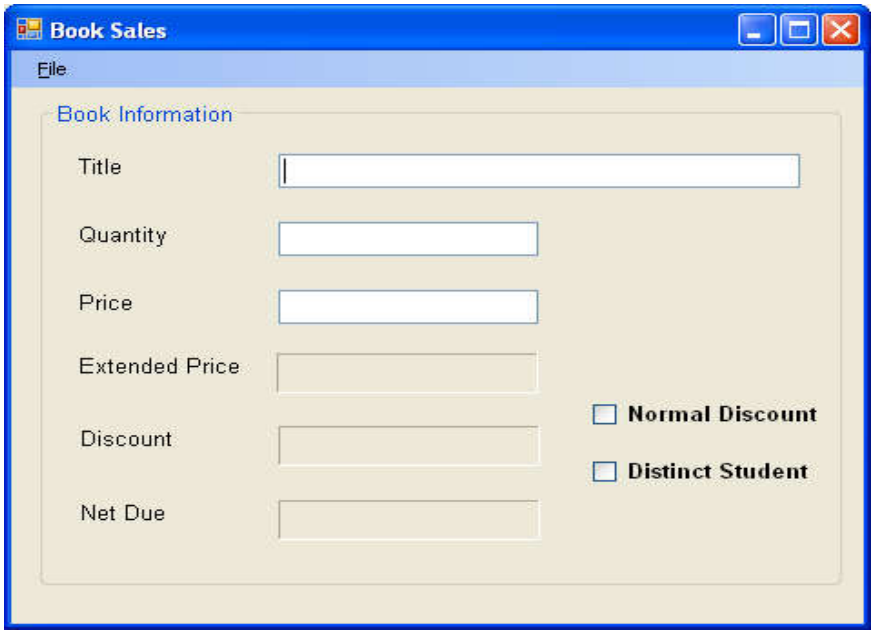
21. Phương thức tính tiền giảm của lớp clsBookSale sẽ bị ghi đè:

```
public virtual decimal DiscountAmount()
```

22. Trong lớp clsStudentBookSale, phương thức DiscountAmount() được định nghĩa:

```
public override decimal DiscountAmount()
{
    decimal decDiscountAmount;

    decDiscountAmount=base.DiscountAmount();

    if (StudentDiscount)
        decDiscountAmount += ExtendedPrice() * Convert.ToDecimal(sngSTUDENT_DISCOUNT_RATE);

    return decDiscountAmount;
}
```

23. Giao diện chương trình được thiết kế lại:



Hình 3.4: Giao diện cuối cùng

24. Sự kiện mnuCalculate_Click được định nghĩa lại:

```
private void mnuCal_Click(object sender, EventArgs e)
{
    try
    {
        mBookSale = new clsStudentBookSale(
                            txtTitle.Text,
                            Convert.ToInt32(txtQuan.Text),
                            Convert.ToDecimal(txtPrice.Text),
                            chkDis.Checked,chkDisStudent.Checked
                    );

        txtPrice.Text = mBookSale.Price.ToString("C");
        lblEPrice.Text = mBookSale.ExtendedPrice().ToString("C");
        lblDis.Text = mBookSale.DiscountAmount().ToString("C");
        lblNetDue.Text = mBookSale.NetDue().ToString("C");
    }
    catch(Exception ex)
    {
        MessageBox.Show("Error in Quantity or Price. " + ex.Message, "Data Entry Error",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}
```

25. Lưu dự án và thực thi ứng dụng.