

Hướng dẫn thực hành Môn LTTQ

Bài tập 2: Anh/Chị hãy sử dụng tạo giao diện đề xuất như hình dưới đây (3 điểm):

Danh sách khách hàng						
000001	Lê Anh	ĐH Giao thông	1000000	1/1/2014	3 tháng	17500

Yêu cầu:

1. Khi xuất hiện form thì hộp ComboBox chứa các mục 1, 3, 6, 12 (0.5 điểm)
2. Các TextBox mã khách hàng, Số tiền gửi chỉ cho phép người dùng nhập vào các số nguyên. (1 điểm)
3. Khi người dùng chọn nút “Thêm mới” (hoặc ấn tổ hợp phím Alt+M) thì đặt tất cả các TextBox thành rỗng, các OptionButton “Loại tiết kiệm” chưa được chọn, ComboBox “Thời gian gửi” chưa có giá trị. (1 điểm)
4. Khi chọn nút Thoát (hoặc tổ hợp phím Alt+H) thì đưa ra thông báo “Bạn có muốn thoát không?” với hai nút lệnh Yes/No. Nếu người dùng chọn Yes thì thoát, chọn No thì không thoát. (1 điểm)
5. Khi người dùng chọn nút “Thêm vào DS” (hoặc ấn tổ hợp phím Alt+T) thì:
 - + Thực hiện kiểm tra: mã khách hàng có đủ 6 chữ số hay không? Họ tên, địa chỉ khách hàng có rỗng không? Ngày gửi có phải là dữ liệu dạng ngày tháng không? Nếu điều kiện nào đó không thì yêu cầu nhập lại. (1.5 điểm)
 - + Nếu các điều kiện kiểm tra được thỏa mãn thì thực hiện tính tiền lãi biết rằng: lãi suất gửi 1, 3, 6, 12 tháng tương ứng là 6%, 7%, 8%, 9% một năm với loại tiết kiệm Thường, với loại tiết kiệm Phát lộc thì được cộng thêm 1% một năm. Các thông tin của khách hàng và số tiền lãi được thêm vào danh sách khách hàng. (1.5 điểm)

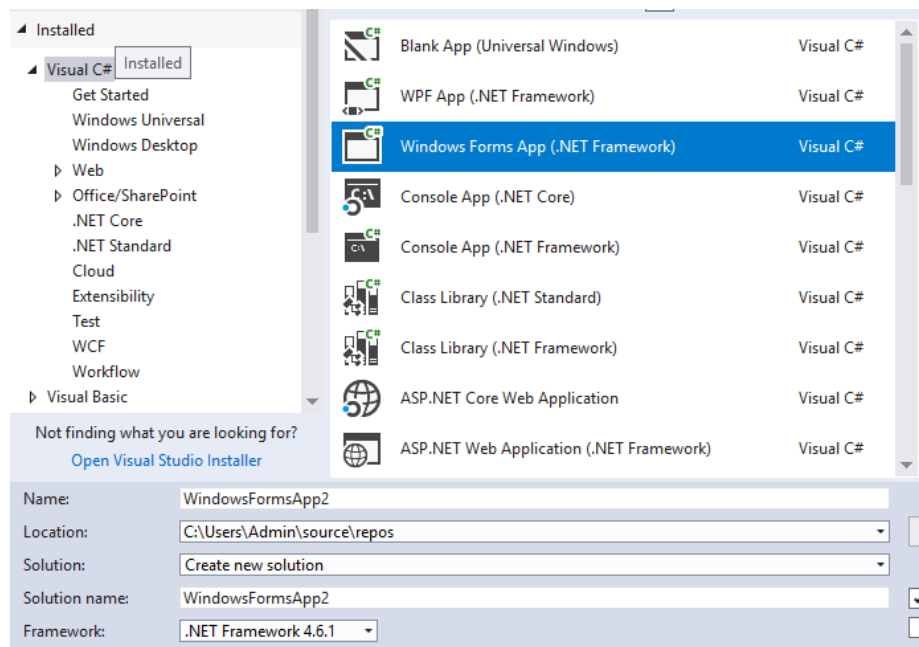
6. Khi người dùng kích chọn nút lệnh “Tìm kiếm”, ứng dụng sẽ hiển thị 1 hộp thoại cho phép người dùng nhập mã khách hàng muốn tìm kiếm. Nếu có mã tương ứng ở trong danh sách thì thông báo ra hộp thoại mã khách hàng và số tiền phải trả. Nếu chưa có thì thông báo là “Khách hàng có mã ” ... “ hiện chưa có trong danh sách”. (chỗ ... là mã của khách hàng đã nhập để tìm kiếm). (0.5 điểm)

Hướng dẫn

Bước 1: Khởi động Visual C#.0

Bước 2: Tạo dự án trên Window Form:

+ Tạo project mới, đặt tên project và chọn vị trí lưu phù hợp



Bước 3: Thiết kế giao diện

- Đưa các điều khiển GroupBox, Label, TextBox, Button, ComboBox, DateTimePicker, RadioButton vào Form và chỉnh sửa như Hình vẽ, đặt các ID tương ứng mang tính chất gợi nhớ.

Bước 4: Sửa thuộc tính

Điều khiển	Thuộc tính	Giá trị	Ghi chú
Form	Text	Tính lãi xuất gửi tiết kiệm	
GroupBox	Text	Nhập thông tin khách hàng gửi tiết kiệm, Loại tiết kiệm, Danh sách khách hàng	3 groupbox, thuộc tính Text lần lượt sửa như trong ô màu vàng

Lable	Text	MaKH, Tên KH, Địa chỉ, Số tiền gửi, ngày gửi, thời gian gửi	6 labels
DateTimePicker	Format	short	Nếu chọn định dạng khác chọn custom
ComboBox	DropDownStyle	DropDownList	Chỉ cho phép người dùng chọn các lựa chọn trong ComboBox, không được tự nhập vào
RadioButton	Text	Thường, Phát lộc	Thêm 2 RadioButton
Button	Text	T&hêm vào DS, Thêm &Mới, Tìm &kiem, Thoát	4 buttions
	Name	BtnThem, btnMoi, btnTimKiem, btnThoat	
TextBox	Name	txtMaKH,txtTenKH,txtDiaChi, txtSoTienGui,	4 TextBoxes
ListBox	Name	lstDanhSach	

Bước 5: Viết mã lệnh

5.1. Khi xuất hiện form thì hộp ComboBox chứa các mục 1, 3, 6, 12 (0.5 điểm)

```
private void AddComboBox()
{
    cbThoiGianGui.Items.Add("1");
    cbThoiGianGui.Items.Add("3");
    cbThoiGianGui.Items.Add("6");
    cbThoiGianGui.Items.Add("12");
}
private void Form1_Load(object sender, EventArgs e)
{
    AddComboBox();
    this.KeyPreview = true;
}
```

5.2. Các TextBox mã khách hàng, Số tiền gửi chỉ cho phép người dùng nhập vào các số nguyên. (1 điểm)

Sự kiện KeyPress của textbox mã khách hàng

```
private void txtMaKH_KeyPress(object sender, KeyPressEventArgs e)
```

```

{
    if (!Char.IsDigit(e.KeyChar) && !Char.IsControl(e.KeyChar))
    {
        e.Handled = true;
    }
}

```

5.3. Khi người dùng chọn nút “Thêm mới” (hoặc ấn tổ hợp phím Alt+M) thì đặt tất cả các TextBox thành rỗng, các OptionButton “Loại tiết kiệm” chưa được chọn, ComboBox “Thời gian gửi” chưa có giá trị. (1 điểm)

Sự kiện Click nút thêm mới

```

private void btnThemMoi_Click(object sender, EventArgs e)
{
    txtMaKH.Text = "";
    txtTenKH.Text = "";
    txtDiaChi.Text = "";
    txtSoTienGui.Text = "";
    dtpNgayGui.Text = "";
    cbThoiGianGui.Text = "";
    cbThoiGianGui.SelectedIndex = -1;
    rdoThuong.Checked = false;
    rdoPhatLoc.Checked = false;
}

```

5.4. Khi chọn nút Thoát (hoặc tổ hợp phím Alt+H) thì đưa ra thông báo “Bạn có muốn thoát không?” với hai nút lệnh Yes/No. Nếu người dùng chọn Yes thì thoát, chọn No thì không thoát. (1 điểm)

Sự kiện Click nút thoát

```

private void btnThoat_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Bạn có muốn thoát không?", "Thông báo",
        MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
        this.Close();
}

```

5.5. Viết mã lệnh cho nút Thêm vào DS

```

private void btnThemVaoDS_Click(object sender, EventArgs e)
{
    int kt = 1;
    if (txtMaKH.TextLength < 6)
    {
        MessageBox.Show("Nhập lại vì mã < 6");
        kt = 0;
    }
    if (txtDiaChi.TextLength == 0 || tbTenKH.TextLength == 0)

```

```

{
    MessageBox.Show("Nhập lại vì tên hoặc địa chỉ rỗng");
    kt = 0;
}

double tienlai=0;
if (kt == 1)
{
    if (rdoThuong.Checked == true)
    {
        if (cbThoiGianGui.SelectedItem == "1")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.06;
        }
        if (cbThoiGianGui.SelectedItem == "3")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.07;
        }
        if (cbThoiGianGui.SelectedItem == "6")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.08;
        }
        if (cbThoiGianGui.SelectedItem == "12")
        {
            tienlai = Convert.ToDouble(txtSoTienGui.Text) * 0.09;
        }
    }
    else if (rdoPhatLoc.Checked == true)
    {
        if (cbThoiGianGui.SelectedItem == "1")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.07;
        }
        if (cbThoiGianGui.SelectedItem == "3")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.08;
        }
        if (cbThoiGianGui.SelectedItem == "6")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.09;
        }
        if (cbThoiGianGui.SelectedItem == "12")
        {
            tienlai = Convert.ToInt32(txtSoTienGui.Text) * 0.1;
        }
    }
}
lbDanhSach.Items.Add(txtMaKH.Text + " | " + tbTenKH.Text + " | " +
    txtDiaChi.Text
    + " | " + dtpNgayGui.Text + " | " + txtSoTienGui.Text + " | "
    + cbThoiGianGui.Text + " tháng | " + tienlai);

List<NguoiGui> listNguoiGuis = new List<NguoiGui>();

```

```

        listNguoiGui.Add(new NguoiGui(Convert.ToInt32(txtMaKH.Text),
tbTenKH.Text, txtDiaChi.Text, Convert.ToInt32(txtSoTienGui.Text),
       .dtpNgayGui.Text, cbThoiGianGui.Text, tienlai
        ));

        StaticData._NguoiGui = listNguoiGui;
    }
}

```

5.6. Khi người dùng kích chọn nút lệnh “Tìm kiếm”, ứng dụng sẽ hiển thị 1 hộp thoại cho phép người dùng nhập mã khách hàng muốn tìm kiếm. Nếu có mã tương ứng ở trong danh sách thì thông báo ra hộp thoại mã khách hàng và số tiền phải trả. Nếu chưa có thì thông báo là “Khách hàng có mã ” ... “ hiện chưa có trong danh sách”. (chỗ ... là mã của khách hàng đã nhập để tìm kiếm). (0.5 điểm)

Ý tưởng:

Tạo một Form mới (Tạm gọi là Form2) để người dùng nhập mã khách hàng và tìm kiếm ở bên Form2. Bên Form2 có các điều khiển, TextBox để nhập mã KH cần tìm, Button để tìm kiếm và một Lable để Thông báo kết quả tìm kiếm.

Ở bên Form ban đầu, mỗi khi ta thêm dữ liệu khách hàng vào listbox(danh sách khách hàng), thì ta cũng thêm đồng thời dữ liệu khách hàng đó vào List các đối tượng khách hàng. Khi gọi Form 2 để tìm kiếm, ta truyền List đối tượng khách hàng đó từ Form1 sang bên Form2, và cuối cùng, ta chỉ cần thao tác tìm kiếm đơn giản ở List trên Form2.

Bước 1: Tạo Lớp NguoiGui (Khách hàng) bằng cách kích phải chuột tại tên project rồi chọn add , tiếp theo chọn class và đặt tên class là NguoiGui

```

class NguoiGui
{
    private int MaKH;
    private string TenKH;
    private string DiaChi;
    private int SoTienGui;
    private string NgayGui;
    private string ThoiGianGui;
    private double Tien;

    public NguoiGui(int maKH)
    {
        MaKH = maKH;
    }

    public NguoiGui(int maKH, string tenKH, string diaChi, int soTienGui, string ngayGui, string thoiGianGui, double tien)
    {
        MaKH = maKH;
        TenKH = tenKH;
        DiaChi = diaChi;
        SoTienGui = soTienGui;
        NgayGui = ngayGui;
    }
}

```

```

        ThoiGianGui = thoiGianGui;
        Tien = tien;
    }

    public int MaKH1 { get => MaKH; set => MaKH = value; }
    public string TenKH1 { get => TenKH; set => TenKH = value; }
    public double Tien1 { get => Tien; set => Tien = value; }
}

```

Bước 2: Tạo một lớp StaticData để chuyển dữ liệu giữa 2 Form

```

class StaticData
{
    public static List<NguoiGui> _NguoiGui = new List<NguoiGui>();
}

```

Bước 3: Ở bên Form1, mỗi khi thêm dữ liệu vào listbox, ta cũng thêm dữ liệu đó vào listNguoiGuis, và gán List đó vào List ở lớp StaticData ở lớp ta đã tạo ở bước 2.

```

List<NguoiGui> listNguoiGuis = new List<NguoiGui>();
listNguoiGuis.Add(new NguoiGui(Convert.ToInt32(txtMaKH.Text),
tbTenKH.Text, txtDiaChi.Text, Convert.ToInt32(txtSoTienGui.Text),
dtpNgayGui.Text, cbThoiGianGui.Text, tienlai
));

StaticData._NguoiGui = listNguoiGuis;

```

Bước 4: Ở Form2, ta khai báo một List mới.

```

List<NguoiGui> listNguoiGuis = new List<NguoiGui>();

```

Khi Form2 được hiển thị, ta gán List ở lớp Static đã tạo ở bước 2 vào List ta mới tạo ra ở Form2.

```

public Form2()
{
    InitializeComponent();
    listNguoiGuis = StaticData._NguoiGui;
}

```

Viết mã lệnh cho nút Tìm Kiếm ở form 2 như sau:

```

private void btnTim_Click(object sender, EventArgs e)
{
    int timthay = 0;
    foreach (NguoiGui i in listNguoiGuis)
    {
        if(i.MaKH1 == Convert.ToInt32(tbTimKiem.Text))
        {
            timthay = 1;
            lbThongTinTimKiem.Text = "Khách hàng " + i.TenKH1 + "phải trả"
                + i.Tien1 + " nghìn đồng";
        }
    }
}

```

```
        if (timthay == 0)
        {
            lbThongTinTimKiem.Text = "Khách hàng " + tbTimKiem.Text + " không có  
trong danh sách";
        }
    }
```

Bước 5: Ở nút lệnh Tìm kiếm của Form1 ta viết mã lệnh như sau:

```
private void btnTimKiem_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    form2.Show();
}
```