

Bài 1: Tạo ứng dụng ghép tên nước với tên thành phố có giao diện như sau:

Capital of Country

Country

- ☒ France
- ☐ Japan
- ☐ Hungary
- ☐ Spain
- ☐ Turkey
- ☐ The UK
- ☐ Italia
- ☐ Argentina
- ☐ Brazil
- ☐ The USA

Capital

- ☐ Buenos Aires
- ☐ Brazil
- ☐ Tokyo
- ☐ Rome
- ☐ Washington
- ☐ Madrid
- ☐ London
- ☐ Ankara
- ☐ Budapest
- ☒ Paris

Chúc mừng bạn, thủ đô của France là Paris

Exit

Yêu cầu:

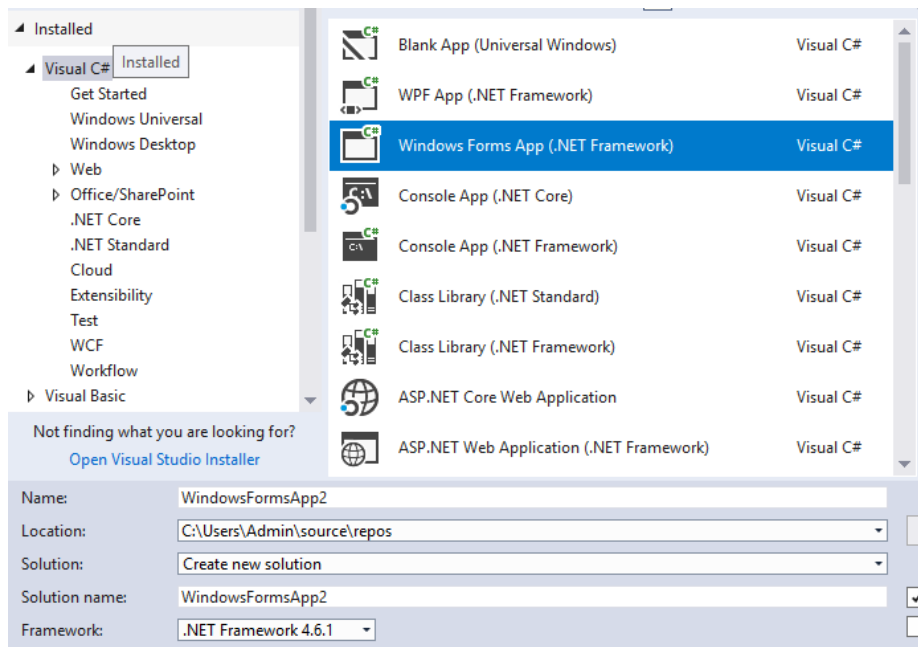
- 1.1. Viết phương thức **EmptyOption()** bỏ chọn tất cả các RadioButton tên thành phố.
- 1.2. Khi kích chọn vào một nước, giả sử France thì xuất hiện dòng thông báo: “Hãy chọn thành phố cho France” và gọi phương thức EmptyOption
- 1.3. Khi chọn một thành phố, nếu đúng là thành phố của tên nước đã chọn thì xuất hiện dòng thông báo, ví dụ: “Chúc mừng bạn, thủ đô của France là Paris”, ngược lại thông báo, ví dụ: “Bạn sai rồi, thủ đô của France không phải là London”

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, Button, RadioButton vào Form và chỉnh sửa như Hình vẽ yêu cầu đề bài, 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	Capital of country	
GroupBox	Text	Country, capital	2 groupbox, thuộc tính Text lần lượt sửa như trong ô màu vàng
Lable			Đặt lable để đưa ra thông báo tên các nước và tên thủ đô
RadioButton	Text		Các radiobutton tương ứng với các tên thủ đô và các nước
Button	Text	Exit	

Bước 5: Viết mã lệnh

1.1. phương thức **EmptyOption()**

```
private void EmptyOption()
{
    rdoAnkara.Checked = false;
    rdoBuenos.Checked = false;
    rdoCaptalBrazil.Checked = false;
    rdoTokyo.Checked = false;
    rdoRome.Checked = false;
    rdoWashington.Checked = false;
    rdoMadrid.Checked = false;
    rdoLondon.Checked = false;
    rdoBudapest.Checked = false;
    rdoParis.Checked = false;
}
```

1.2. Khi kích chọn vào một nước, giả sử France thì xuất hiện dòng thông báo: “Hãy chọn thành phố cho France” và gọi phương thức EmptyOption

Trước hết, ta tạo hàm getNotification để trả về một chuỗi tương ứng với tên đất nước được chọn để dùng

```
private string getNotification(string country)
{
    EmptyOption();
    return "Hãy chọn thành phố cho " + country;
}
```

Sau đó, khi chọn radioButton đất nước nào thì ta gán cho label bằng dòng chữ “hãy chọn thành phố cho đất nước đó”

Ví dụ là đất nước Pháp, các em làm tương tự cho các radioButton khác

```
private void rdoFrance_CheckedChanged(object sender, EventArgs e)
{
    country = rdoFrance.Text;
    lbYeuCau.Text = getNotification(rdoFrance.Text);
}
```

1.3. Khi chọn một thành phố, nếu đúng là thành phố của tên nước đã chọn thì xuất hiện dòng thông báo, ví dụ: “Chúc mừng bạn, thủ đô của France là Paris”, ngược lại thông báo, ví dụ: “Bạn sai rồi, thủ đô của France không phải là London”

Để dễ dàng cho việc quản lý chương trình, ta nên tổ chức thành các đối tượng, cũng có thể xét điều kiện trực tiếp trên các RadioButton.

Tạo lớp Country_Capital với các hàm tạo và hàm set, get tương ứng

```

class Country_Capital
{
    string country;
    string capital;

    public Country_Capital(string country, string capital)
    {
        this.country = country;
        this.capital = capital;
    }

    public string Country { get => country; set => country = value; }
    public string Capital { get => capital; set => capital = value; }
}

```

Sau đó, tạo một danh sách Country_Capital:

```
List<Country_Capital> listCountry_Capitals = new List<Country_Capital>();
```

Ta viết hàm add dữ liệu và gọi ở hàm tạo của Form

```

public Form1()
{
    InitializeComponent();
    EmptyOption();
    addData();
}
private void addData()
{
    listCountry_Capitals.Add(new Country_Capital("France", "Paris"));
    listCountry_Capitals.Add(new Country_Capital("Japan", "Tokyo"));
    listCountry_Capitals.Add(new Country_Capital("Hungary", "Budapest"));
    listCountry_Capitals.Add(new Country_Capital("Spain", "Madrid"));
    listCountry_Capitals.Add(new Country_Capital("Turkey", "Ankara"));
    listCountry_Capitals.Add(new Country_Capital("The UK", "London"));
    listCountry_Capitals.Add(new Country_Capital("Italia", "Rome"));
    listCountry_Capitals.Add(new Country_Capital("Argentina", "Buenos Aires"));
    listCountry_Capitals.Add(new Country_Capital("Brazil", "Brazil"));
    listCountry_Capitals.Add(new Country_Capital("The USA", "Washington"));
}

```

Ta có 2 biến country và capital để lưu trữ tên thành phố và đất nước đang được chọn

```

string capital = "";
string country = "";

```

Mỗi một lần ấn chọn một radioButton, ta lại lưu tên đất nước và tên thủ đô tương ứng

Ví dụ: Chọn radioButton Japan, ta gán biến country bằng tên đất nước tương ứng

```
private void rdoJapan_CheckedChanged(object sender, EventArgs e)
{
    country = rdoJapan.Text;

    lbYeuCau.Text = getNotification(rdoJapan.Text);
}
```

Viết hàm kiểm tra tính đúng đắn thủ đô với nước cho trước.

```
bool check(string country, string capital)
{
    foreach (Country_Capital i in listCountry_Capitals)
    {
        if (i.Country.Equals(country) && i.Capital.Equals(capital))
        {
            return true;
        }
    }
    return false;
}
```

Mỗi khi chọn radioButton thủ đô, ta gán tên thủ đô vừa chọn vào biến capital ở trên, và sử dụng hàm check trên để kiểm tra tính đúng đắn. Dưới đây là ví dụ chọn thủ đô Brazil. Các thủ đô khác làm tương tự.

```
private void rdoCapitalBrazil_CheckedChanged(object sender, EventArgs e)
{
    capital = rdoCapitalBrazil.Text;
    if (check(country, capital))
    {
        lbYeuCau.Text = ("Chúc mừng bạn, thủ đô của " + country + " là " + capital);
    }
    else
    {
        lbYeuCau.Text = ("Bạn sai rồi, thủ đô của " + country + " không phải là " + capital);
    }
}
```

Bài 2:

Form bán hàng

Yêu cầu:

- 2.1. Danh sách các mặt hàng được hiển thị sẵn trong hộp Listbox hoặc CheckedListBox “Danh sách các mặt hàng”
- 2.2. Để mua hàng người dùng nháy đúp vào mặt hàng cần mua trong “Danh sách các mặt hàng”, mặt hàng được chọn sẽ được hiển thị vào trong “Hàng đặt mua”.
- 2.3. Khi mua hàng phải kiểm tra nếu mặt hàng này đã được mua thì dùng hộp thoại thông báo đã chọn mặt hàng đó và không được mua mặt hàng đó nữa.
- 2.4. Người dùng có thể xóa mặt hàng trong số các mặt hàng đã chọn bằng cách kích đúp vào mặt hàng cần xóa, trước khi xóa phải hỏi lại người dùng có muốn xóa hay không?
- 2.5. Hiển thị hộp thông báo tất cả các thông tin khách hàng đã chọn (Họ tên, địa chỉ, các mặt hàng...) khi nhấn nút Đồng ý

Hướng dẫn

Tạo project như đã hướng dẫn phía trên

Bước 1: Thiết kế giao diện

Đưa các điều khiển Label, ListBox, TextBox, Button, ComboBox, RadioButton vào Form và chỉnh sửa như Hình vẽ theo yêu cầu đề bài, đặt các ID tương ứng mang tính chất gợi nhớ.

Bước 2: Sửa thuộc tính

Điều khiển	Thuộc tính	Giá trị	Ghi chú
Form	Text	Bán sách qua mạng	
ListBox	Name	lstDanhSachMatHang, lstHangDatMua	2 ListBox, thuộc tính Name lần lượt sửa như trong ô màu vàng
Lable	Text	Họ tên khách, địa chỉ	2 labels
GroupBox	Text	Phương thức thanh toán, hình thức liên lạc	

RadioButton	Name	rdbTienMat, rdbSec, rdbTheTinDung	Các name tương ứng
	Text	Tiền mặt, sec, thẻ tín dụng	Thêm 3 RadioButton
TextBox	Name	txtHoTen, txtDienThoai	3 TextBoxes
CheckBox	Name	chbDienThoai, chbFax, chbMail	các name tương ứng
	Text	Điện thoại, Fax, Mail	3 checkbox

Bước 5: Mã lệnh

2.1. Danh sách các mặt hàng được hiển thị sẵn trong hộp Listbox hoặc CheckedListBox “Danh sách các mặt hàng”

Xây dựng hàm addData để thêm thông tin vào listBox

```
private void Form1_Load(object sender, EventArgs e)
{
    addData();
}

private void addData()
{
    lstDanhSachMathang.Items.Add("Kỹ thuật lập trình C#");
    lstDanhSachMathang.Items.Add("Tự học Visual C# trong 21 ngày");
    lstDanhSachMathang.Items.Add("Bài tập C#");
    lstDanhSachMathang.Items.Add(".NET toàn tập-tập 1");
    lstDanhSachMathang.Items.Add(".NET toàn tập-tập 2");
    lstDanhSachMathang.Items.Add(".NET toàn tập-tập 3");
    lstDanhSachMathang.Items.Add(".NET toàn tập-tập 4");
    lstDanhSachMathang.Items.Add("Tin học cơ bản");
    lstDanhSachMathang.Items.Add("SQL server");
    lstDanhSachMathang.Items.Add("Cơ bản về XML");
    lstDanhSachMathang.Items.Add("Phân tích thiết kế hệ thống");
    lstDanhSachMathang.Items.Add("Sử dụng Word");
    lstDanhSachMathang.Items.Add("Đến với Word 2003");
}
```

2.2. Để mua hàng người dùng nháy đúp vào mặt hàng cần mua trong “Danh sách các mặt hàng”, mặt hàng được chọn sẽ được hiển thị vào trong “Hàng đặt mua”.

2.3. Khi mua hàng phải kiểm tra nếu mặt hàng này đã được mua thì dùng hộp thoại thông báo đã chọn mặt hàng đó và không được mua mặt hàng đó nữa.

Ta bắt sự kiện MouseDoubleClick của listBox Danh sách mặt hàng

```

private void lstDanhSachMatHang_MouseDoubleClick(object sender, MouseEventArgs e)
{
    string curItem = lstDanhSachMatHang.SelectedItem.ToString();

    //tìm kiếm xem có trong hàng đặt mua
    int index = lstHangDatMua.FindString(curItem);
    // nếu mà k có thì in thông báo và thêm vào hàng đặt mua ngược lại báo đã có
    if (index == -1)
    {
        lstHangDatMua.Items.Add(curItem);
    }
    else
        MessageBox.Show("Hàng đặt mua đã có rồi ");
}

```

2.4. Người dùng có thể xóa mặt hàng trong số các mặt hàng đã chọn bằng cách kích đúp vào mặt hàng cần xóa, trước khi xóa phải hỏi lại người dùng có muốn xóa hay không?

Tương tự, ta cũng bắt sự kiện click đúp chuột và đưa ra messagebox hỏi xem có chắc chắn muốn xóa không

```

private void lstHangDatMua_MouseDoubleClick(object sender, MouseEventArgs e)
{
    int index = lstHangDatMua.SelectedIndex;
    if (MessageBox.Show("Bạn có muốn xóa không?", "Thông báo", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
        lstHangDatMua.Items.RemoveAt(index);
}

```

2.5. Hiện thị hộp thông báo tất cả các thông tin khách hàng đã chọn (Họ tên, địa chỉ, các mặt hàng...) khi nhấn nút Đồng ý

Ở đây, ta cần lấy ra được tất cả các thông tin của khách hàng. Lần lượt xây dựng các hàm lấy các hình thức liên lạc, phương thức thanh toán, danh sách các mục đã đặt mua được lưu trong biến sb khi kích nút đồng ý:

```

private string HinhThucLL()
{
    string a = "";
    if (chbDienThoai.Checked == true)
    {
        a += "    " + chbDienThoai.Text;
    }
    if (chbEmail.Checked == true)
    {
        a += "    " + chbEmail.Text;
    }
    if (chbFax.Checked == true)
    {
        a += "    " + chbFax.Text;
    }
    return a;
}

```



```

private string ThanhToan()
{
    string s = "";
    if (rdbTienMat.Checked == true)
    {
        s += rdbTienMat.Text;
    }
    else if (rdbSec.Checked == true)
    {
        s += rdbSec.Text;
    }
    else if (rdbTheTinDung.Checked == true)
    {
        s += rdbTheTinDung.Text;
    }
    return s;
}

private void btnDongY_Click(object sender, EventArgs e)
{
    if (txtHoTen.Text.Equals("") || txtDienThoai.Text.Equals(""))
    {
        MessageBox.Show("Bạn cần nhập đủ thông tin", "Thông báo");
        txtHoTen.Focus();
    }
    else
    {
        string sb = "";
        foreach (object item in lstHangDatMua.Items)
        {
            sb += (item.ToString());
            sb += ("\n");
        }
    }
}

```

Tiếp theo, khi ấn nút đồng ý, ta cần kiểm tra họ tên và điện thoại đã nhập đủ chưa, ta có thể kiểm tra bằng sự kiện Leave như những bài trước

```

private void btnDongY_Click(object sender, EventArgs e)
{
    if (txtHoTen.Text.Equals("") || txtDienThoai.Text.Equals(""))
    {
        MessageBox.Show("Bạn cần nhập đủ thông tin", "Thông báo");
        txtHoTen.Focus();
    }
    else ...
}

```