Cấu trúc bài thi Lập trình trên Windows (***Đề đóng*** thời gian 75-90 phút)

* Event
* Delegate
* Đọc file

readLine() -> string

Split(‘’ ‘’,s)

You TUBE : “tìm kiếm và liệt kê các danh sách các tỉnh thành\_\_Hoclaptrinh”

“chương trình danh bạ điện thoại thông minh”

Cấu trúc thời gian:

* + - Thiết kế form :10-15’
    - Code form :15-25’
    - Chỉnh sửa code + form : 15’

Sử dụng control: pictureBox,combobox,textbox,treeview

==========================================

pic1.Image = Image.FromFile(@"C:\Users\Administrator\Desktop\Project C#\RandomPicture\RandomPicture\bin\Debug\Images\bietthu.jpg");

===============================================================

\*Delegate:

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

/\* Form2 frm2 = new Form2();

this.Hide();

frm2.ShowDialog();

this.Close();\*/

Form2 frm2 = new Form2();

frm2.sendMsg += frm\_sendMsg;

frm2.ShowDialog();

this.Hide();

}

public void frm\_sendMsg(string value)

{

label1.Text = value;

}

}

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public delegate void Message(String value);

public event Message sendMsg;

private void Form2\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

sendMsg.Invoke("nguyen ho duy khang");

}

}

=======================================================================

\*Đặc sản vùng miền:

class DacSan

{

private string Ten;

public string TEN

{

get { return Ten; }

set { Ten = value; }

}

private string Mien;

public string MIEN

{

get { return Mien; }

set { Mien = value; }

}

System.Drawing.Image Hinh;

public System.Drawing.Image HINH

{

get { return Hinh; }

set { Hinh = value; }

}

public DacSan(string TEN,string MIEN,System.Drawing.Image HINH)

{

Ten = TEN;

Mien = MIEN;

Hinh = HINH;

}

}

List<DacSan> list = new List<DacSan>();

cbMien.Items.Add(BAC);

cbMien.Items.Add(TRUNG);

cbMien.Items.Add(NAM);

cbMien.Items.Add(TATCA);

cbMien.SelectedIndex = 3;

public void loadListbox()

{

list.Add(ds1);

list.Add(ds2);

foreach(DacSan d in list){

lbMonAn.Items.Add(d.TEN);

}

}

========================================================================

\*\*\*Hàm Trim():là hàm dùng để loại bỏ khoảng trắng

\*\*\*Trim().length:dùng để lấy độ dài chuổi

//example

String text = tb.Text;

int le = text.Trim().Length;

MessageBox.Show(Convert.ToString(le));

========================================================================

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

SolidBrush brush= new SolidBrush(Color.Red);

Rectangle rec=new Rectangle(200,200,20,20);

Graphics paper;

private void Form1\_Load(object sender, EventArgs e)

{

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

paper = e.Graphics;

draw(paper);

}

public void draw(Graphics paper)

{

paper.FillEllipse(brush, rec);

}

}

===============================================================================

Treeview

public Form1()

{

InitializeComponent();

}

String text = "";

private void Form1\_Load(object sender, EventArgs e)

{

TreeNode node1=new TreeNode("Lập trình di động");

TreeNode node2 = new TreeNode("Lập Web");

TreeNode node3 = new TreeNode("Android");

TreeNode node4 = new TreeNode("IOS");

TreeNode[] array = new TreeNode[] { node3,node4};

treeView.Nodes.Add(node1);

treeView.Nodes.Add(node2);

TreeNode node5 = new TreeNode("Programming",array);

treeView.Nodes.Add(node5);

}

//Xác định node click hiện tại

private void treeView\_AfterSelect(object sender, TreeViewEventArgs e)

{

MessageBox.Show(e.Node.Text);

}

}

=================================================================================

Abstract:

public abstract class Hinh

{

public abstract float GetDienTich();//khai báo 1 abstract method

}

public class HinhTron:Hinh // lớp con HinhTron kế thừa từ lớp cha Hinh

{

private float fltBanKinh, fltPI;//có thể khai báo thêm biến trong các lớp con kế thừa

public override float GetDienTich() // xử lý hàm từ lớp cha đã khai báo

{

return (fltPI \* (fltBanKinh\* 2));

}

}

public class HinhChuNhat:Hinh

{

private float fltDai, fltRong;

public override float GetDienTich(Hinh other)

{

return fltDai \* fltRong;

}

}

class Program

{

static void Main(string[] args)

{

Hinh ojhinhtron=new HinhTron();

ojhinhtron.fltBanKinh=30;

ojhinhtron.fltPI=3.14;

float fltdientich=ojhinhtron.GetDienTich();

System.Console.WriteLine("Diện tích hình tròn là ': {0}",fltdientich);

......

......

}

}

========================================================================\* Polymorphim:

namespace Virtual\_Override

{

class Person

{

public String name;

public Int16 age;

public virtual String show()

{

String text="I am person";

return text;

}

}

}

/\*Class Student.cs\*/

namespace Virtual\_Override

{

class Student:Person

{

public override String show()

{

String text = "I am student";

return text;

}

}

}

/\*Form.cs\*/

namespace Virtual\_Override

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

Person p = new Person();

//MessageBox.Show(p.show());

Person s = new Student();

MessageBox.Show(s.show());

}

}

}

========================================================================Danh sách các tỉnh thành:

public partial class frmDSTT : Form

{

const string FILENAME = "DSTT.csv";

SortedList<string, City> cities;

//tên, mã vùng,mã bưu chính, số xe, dân số(người), diện tích(km2),thuộc, loại

class City

{

public static char[] separator= new char[]{','};

public City(string line)

{

// TODO: Complete member initialization

string[] array = line.Split(separator);

Name = array[0];

AreaCode = int.Parse(array[1]);

PostCode = int.Parse(array[2]);

VehicleCode = array[3].Replace(';', ',');

Population = int.Parse(array[4]);

Area = float.Parse(array[5]);

RegionName = array[6];

CityType = array[7];

}

/\*

\* Thuộc:Tây Nguyên

\* Mã vùng:59

\* Mã bưu chính:54

\* Số xe:81

\* Dân số:1,322,000 người

\* Diện tích: 15,536.9 km2

\*/

public override string ToString()

{

return string.Format("{0} {1}{8}Thuộc: {2}{8}Mã vùng: {3}{8}Mã bưu chính: {4}{8}Số xe: {5}{8}Dân số: {6:#,##0}người{8}Diện tích: {7:#,##0.0} km\u00B2",

CityType,Name,RegionName,AreaCode,PostCode,VehicleCode,Population,Area,Environment.NewLine);

}

public string Name

{ get; set; }

public int AreaCode { get; set; }

public int PostCode { get; set; }

public string VehicleCode { get; set; }

public int Population { get; set; }

public float Area { get; set; }

public string RegionName { get; set; }

public string CityType { get; set; }

}

public frmDSTT()

{

InitializeComponent();

cities= new SortedList<string,City>();

}

private void frmDSTT\_Load(object sender, EventArgs e)

{

ReadDataFile();

LoadToListBox();

}

private void ReadDataFile()

{

using (StreamReader sr = new StreamReader(FILENAME))

{

sr.ReadLine();

while (!sr.EndOfStream)

{

City c = new City(sr.ReadLine());

cities.Add(c.Name.ToLower(), c);

}

}

}

private void LoadToListBox()

{

lstCities.Items.Clear();

foreach (City city in cities.Values)

{

lstCities.Items.Add(city.Name);

}

}

private void lstCities\_SelectedIndexChanged(object sender, EventArgs e)

{

if (lstCities.SelectedIndex < 0) return;

City c;

if(!cities.TryGetValue(lstCities.Text.ToLower(),out c))

{

MessageBox.Show("Không tìm thấy dữ liệu tỉnh thành này","Lỗi");

return;

}

txtResult.Text=c.ToString();

}

private void txtNameToSearch\_TextChanged(object sender, EventArgs e)

{

string keyword = txtNameToSearch.Text.Trim().ToLower() ;

StringBuilder stringBuilder = new StringBuilder();

foreach (KeyValuePair<string, City> pair in cities)

{

if (pair.Key.IndexOf(keyword) >= 0)

stringBuilder.AppendFormat("{0} {1}//{1}", pair.Value, Environment.NewLine);

txtResult.Text = stringBuilder.ToString();

}

}

}

=====================================================================

Đọc file

const string FileName = "data.txt";//nằm ở thư mục bin/debug

List<string> list = new List<string>();

public void ReadFile()

{

using (StreamReader sr = new StreamReader(FileName))

{

//Đọc nhiều dòng

string line = "";

while ((line = sr.ReadLine()) != null)

{

list.Add(line);

}

}

}

========================================================================

Hàm write file

public void writeFile(string content,string FileName)

{

using (StreamWriter Writer = new StreamWriter(FileName))

{

Writer.WriteLine(content);

}

}

============================================================================

Split:

string s = "there is a cat";

// Split string on spaces.

// ... This will separate all the words.

string[] words = s.**Split**(' ');

foreach (string word in words)

{

Console.WriteLine(word);

}

================================================================================== public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

const string FileName = "data.txt";

SortedList<string,Test> sortList;

List<string> list = new List<string>();

string[,] a = new string[4,2];

int count = 0, count1 = 0;

class Test

{

}

private void Form1\_Load(object sender, EventArgs e)

{

ReadFile();

getList();

LoadDataListBox();

LoadComboBox();

}

public void ReadFile()

{

using (StreamReader sr = new StreamReader(FileName))

{

//Đọc nhiều dòng

string line = "";

while ((line = sr.ReadLine()) != null)

{

list.Add(line);

}

}

}

public void getList()

{

foreach (string item in list)

{

string[] values = item.Split(',');

foreach (string v in values)

{

a[count, count1] = v;

count1++;

}

count++;

count1 = 0;

}

}

public void LoadDataListBox()

{

for(int i=0;i<4;i++)

{

lb.Items.Add(a[i,0]);

}

}

public void LoadComboBox()

{

cb.Items.Add("Khang");

cb.Items.Add("huy");

cb.Items.Add("Lâm");

cb.SelectedIndex = 0;

}

private void lb\_SelectedIndexChanged(object sender, EventArgs e)

{

int index = lb.SelectedIndex;

MessageBox.Show(a[index,1]);

}

}