**LAB1**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace lab01

{

class Student

{

private int codeid;

private string name;

private string faculty;

private float gpa;

public Student()

{

codeid = 1;

name = " Le Ngoc Hao";

faculty = "CNTT";

gpa = 9;

}

public Student(Student stu)

{

codeid = stu.codeid;

name = stu.name;

faculty = stu.faculty;

gpa = stu.gpa;

}

public Student(int id, string ten, string kh, float dtb)

{

codeid = id;

name = ten;

faculty = kh;

gpa = dtb;

}

public int Codeid

{

get { return codeid; }

set { codeid = value; }

}

public String Name

{

get { return name; }

set { name = value; }

}

public String Faculty

{

get { return faculty; }

set { faculty = value; }

}

public float GPA

{

get { return gpa; }

set { gpa = value; }

}

public void Show()

{

Console.WriteLine("Code ID:{0}", codeid);

Console.WriteLine("Name:{0}", name);

Console.WriteLine("Faculty:{0}", faculty);

Console.WriteLine("GPA:{0}", gpa);

}

class Tester

{

public static void Main()

{

Student[] DSSV; int n;

Console.Write("Number of students:");

n = Convert.ToInt32(Console.ReadLine()); DSSV = new Student[n];

Console.WriteLine("\n ====Enter student list====");

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

{

DSSV[i] = new Student();

Console.Write("Input Code ID {0}:", i + 1);

DSSV[i].Codeid = int.Parse(Console.ReadLine());

Console.Write("Input Name:");

DSSV[i].Name = Console.ReadLine();

Console.Write("Input Faculty:");

DSSV[i].Faculty = Console.ReadLine();

Console.Write("Input GPA:");

DSSV[i].GPA = float.Parse(Console.ReadLine());

}

Console.WriteLine("\n====Output student list====");

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

{

DSSV[i].Show();

}

Console.ReadLine();

}

}

}

}

using lab01;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lap1

{

class Student2

{

private int codeid;

private string name;

private string faculty;

private float gpa;

public int GetCodeid()

{

return codeid;

}

public void SetCodeid(int value)

{

this.codeid = value;

}

public string GetName()

{

return name;

}

public void SetName(string value)

{

this.name = value;

}

public string GetFaculty()

{

return faculty;

}

public void SetFaculty(string value)

{

this.faculty = value;

}

public float GetGPA()

{

return gpa;

}

public void SetGPA(float value)

{

this.gpa = value;

}

public void Show()

{

Console.WriteLine("Code ID:{0}", codeid);

Console.WriteLine("Name:{0}", name);

Console.WriteLine("Faculty:{0}", faculty);

Console.WriteLine("GPA:{0}", gpa);

}

class Tester

{

static ArrayList listSV = new ArrayList();

public static void InputSV()

{

Console.WriteLine("\n ====Enter student====");

Student2 sv = new Student2();

Console.Write("Input Code ID :");

sv.SetCodeid(int.Parse(Console.ReadLine()));

Console.Write("Input Name:");

sv.SetName(Console.ReadLine());

Console.Write("Input Faculty:");

sv.SetFaculty(Console.ReadLine());

Console.Write("Input GPA:");

sv.SetGPA(float.Parse(Console.ReadLine()));

listSV.Add(sv);

}

public static void InputList()

{

int n;

Console.Write("Number of students:");

n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("\n ====Enter student list====");

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

{

InputSV();

}

}

public static void OutputList()

{

Console.WriteLine("\n====Output student list====");

foreach (Student2 sv in listSV)

{

sv.Show();

}

}

public static void Main()

{

InputList();

OutputList();

Console.ReadLine();

}

}

}

}

using lab01;

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lap1

{

class People

{

private string name;

private string address;

public string Name

{

get { return name; }

set { name = value; }

}

public string Address

{

get { return address; }

set { address = value; }

}

}

class Student3 : People

{

private int codeid;

private string faculty;

private float gpa;

public int Codeid

{

get { return codeid; }

set { codeid = value; }

}

public String Faculty

{

get { return faculty; }

set { faculty = value; }

}

public float GPA

{

get { return gpa; }

set { gpa = value; }

}

public void Show()

{

Console.WriteLine("Code ID:{0}", codeid);

Console.WriteLine("Name:{0}", Name);

Console.WriteLine("Address:{0}", Address);

Console.WriteLine("Faculty:{0}", faculty);

Console.WriteLine("GPA:{0}", gpa);

}

class Tester

{

static ArrayList listSV = new ArrayList();

public static void InputSV()

{

Console.WriteLine("\n ====Enter student====");

Student3 sv = new Student3();

Console.Write("Input Code ID :");

sv.codeid = int.Parse(Console.ReadLine());

Console.Write("Input Name:");

sv.Name = Console.ReadLine();

Console.Write("Input Address:");

sv.Address = Console.ReadLine();

Console.Write("Input Faculty:");

sv.faculty = Console.ReadLine();

Console.Write("Input GPA:");

sv.gpa = float.Parse(Console.ReadLine());

listSV.Add(sv);

}

public static void InputList()

{

int n;

Console.Write("Number of students:");

n = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("\n ====Enter student list====");

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

{

InputSV();

}

}

public static void OutputList()

{

Console.WriteLine("\n====Output student list====");

foreach (Student3 sv in listSV)

{

sv.Show();

}

}

public static void Main()

{

InputList();

OutputList();

Console.ReadLine();

}

}

}

}

**LAB2**

using System;

using System.Collections;

using System.Collections.Generic;

using System.Dynamic;

using System.Linq;

using System.Text;

namespace Lab\_2

{

interface IBook

{

string this[int index]

{

get;

set;

}

string Title

{

get;

set;

}

string Author

{

get;

set;

}

string Publisher

{

get;

set;

}

string ISBN

{

get;

set;

}

int Year

{

get;

set;

}

void Show();

}

class Book : IBook

{

private string isbn;

private string title;

private string author;

private string publisher;

private int year;

private ArrayList chapter = new ArrayList();

public string this[int index]

{

get

{

if (index >= 0 && index < chapter.Count)

return (string)chapter[index];

else

throw new NotImplementedException();

}

set

{

if (index >= 0 && index < chapter.Count)

chapter[index] = value;

else if (index == chapter.Count)

chapter.Add(value);

else

throw new InternalBufferOverflowException();

}

}

public string ISBN

{

get { return isbn; }

set { isbn = value; }

}

public string Title

{

get { return title; }

set { title = value; }

}

public string Author

{

get { return author; }

set { author = value; }

}

public string Publisher

{

get { return publisher; }

set { publisher = value; }

}

public int Year

{

get { return year; }

set { year = value; }

}

public void Show()

{

Console.WriteLine("--------------------------");

Console.WriteLine("Title:" + title);

Console.WriteLine("Author:" + author);

Console.WriteLine("Publisher:" + publisher);

Console.WriteLine("Year:" + year);

Console.WriteLine("ISBN:" + isbn);

Console.WriteLine("chapter:");

for (int i = 0; i < chapter.Count; i++)

Console.WriteLine("\t{0}: {1}", i + 1, chapter[i]);

Console.WriteLine("--------------------------");

}

public void Input()

{

Console.WriteLine("Title: ");

title = Console.ReadLine();

Console.WriteLine("Author: ");

author = Console.ReadLine();

Console.WriteLine("Publisher: ");

publisher = Console.ReadLine();

Console.WriteLine("ISBN: ");

isbn = Console.ReadLine();

Console.WriteLine("Year: ");

year = int.Parse(Console.ReadLine());

Console.WriteLine("Input chapter (finished with empty string)");

string str;

do

{

str = Console.ReadLine();

if (str.Length > 0)

chapter.Add(str);

} while (str.Length > 0);

}

}

class BookList

{

private ArrayList list = new ArrayList();

public void AddBook()

{

Book b = new Book();

b.Input();

list.Add(b);

}

public void ShowList()

{

foreach (Book b in list)

{

b.Show();

}

}

public void InputList()

{

int n;

Console.Write("Amount of books: ");

n = int.Parse(Console.ReadLine());

while (n > 0)

{

AddBook();

n--;

}

}

}

class Program

{

static void Main(string[] args)

{

BookList bl = new BookList();

bl.InputList();

bl.ShowList();

Console.ReadLine();

}

}

}

using System;

using System.Collections;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_2

{

public class Book2 : IComparable

{

private string isbn;

private string title;

private string author;

private string publisher;

private int year;

public string GetISBN() { return isbn; }

public void SetISBN(string isbn) { this.isbn = isbn; }

public string GetTitle() { return title; }

public void SetTitle(string title) { this.title = title; }

public string GetAuthor() { return author; }

public void SetAuthor(string author) { this.author = author; }

public string GetPublisher() { return publisher; }

public void SetPublisher(string publisher) { this.publisher = publisher; }

public int GetYear() { return year; }

public void SetYear(int year) { this.year = year; }

public void Show()

{

Console.WriteLine("--------------------------");

Console.WriteLine("Title:" + title);

Console.WriteLine("Author:" + author);

Console.WriteLine("Publisher:" + publisher);

Console.WriteLine("Year:" + year);

Console.WriteLine("ISBN:" + isbn);

Console.WriteLine("--------------------------");

}

public void Input()

{

Console.WriteLine("Title: ");

title = Console.ReadLine();

Console.WriteLine("Author: ");

author = Console.ReadLine();

Console.WriteLine("Publisher: ");

publisher = Console.ReadLine();

Console.WriteLine("ISBN: ");

isbn = Console.ReadLine();

Console.WriteLine("Year: ");

year = int.Parse(Console.ReadLine());

}

public int CompareTo(object incomingobject)

{

// Storing incoming object in temp variable of

// current class type

Book2 incomingemployee = incomingobject as Book2;

return this.title.CompareTo(incomingemployee.title);

}

}

public class Program2

{

public static ArrayList list = new ArrayList();

public static void AddBook()

{

Book2 b = new Book2();

b.Input();

list.Add(b);

}

public static void ShowList()

{

foreach (Book2 b in list)

{

b.Show();

}

}

public static void InputList()

{

int n;

Console.Write("Amount of books: ");

n = int.Parse(Console.ReadLine());

while (n > 0)

{

AddBook();

n--;

}

}

static void Main(string[] args)

{

ArrayList list = new ArrayList();

int n;

Console.Write("Amount of books: ");

n = int.Parse(Console.ReadLine());

while (n > 0)

{

Book2 b = new Book2();

b.Input();

list.Add(b);

n--;

}

foreach (Book2 b in list)

{

b.Show();

}

Console.WriteLine("=========Sort by Title==========");

list.Sort();

foreach (Book2 b in list)

{

b.Show();

}

Console.ReadLine();

}

}

}

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab\_2

{

class Account

{

private int accountID;

private string firstName;

private string lastName;

private decimal balance;

public int AccountID { get => accountID; set => accountID = value; }

public string FirstName { get => firstName; set => firstName = value; }

public string LastName { get => lastName; set => lastName = value; }

public decimal Balance { get => balance; set => balance = value; }

public Account()

{

this.firstName = string.Empty;

this.lastName = string.Empty;

this.balance = 0;

this.accountID = 0;

}

public Account(int aID, string fName, string lName, decimal bl)

{

this.accountID = aID;

this.firstName = fName;

this.lastName = lName;

this.balance = bl;

}

public void Show()

{

Console.WriteLine("--------------------------");

Console.WriteLine("Account ID:" + accountID);

Console.WriteLine("First Name:" + firstName);

Console.WriteLine("Last Name:" + lastName);

Console.WriteLine("Balance:" + balance);

Console.WriteLine("--------------------------");

}

public void Input()

{

Console.Write("Input account ID: ");

accountID = int.Parse(Console.ReadLine());

Console.Write("Input first name: ");

firstName = Console.ReadLine();

Console.Write("Input last name: ");

lastName = Console.ReadLine();

Console.Write("Input balance: ");

balance = decimal.Parse(Console.ReadLine());

}

}

class FirstNameComparer : IComparer

{

int IComparer.Compare(object a, object b)

{

Account x = (Account)a;

Account y = (Account)b;

if (x == null) return -1;

if (y == null) return 1;

if (ReferenceEquals(x, y)) return 0;

return String.Compare(x.FirstName, y.FirstName);

}

}

//class BalanceComparer : IComparer

//{

// int IComparer.Compare(object a, object b)

// {

// Account x = (Account)a;

// Account y = (Account)b;

// if (x == null) return -1;

// if (y == null) return 1;

// if (ReferenceEquals(x, y)) return 0;

// return x.Balance == y.Balance ? 0 : x.Balance > y.Balance ? 1 : -1;

// }

//}

class AccountList

{

private ArrayList list = new ArrayList();

public void NewAccount()

{

Account a = new Account();

a.Input();

list.Add(a);

}

public void SaveFile()

{

Console.Write("Input file name to save: ");

string fileName = Console.ReadLine();

try

{

FileStream output = new FileStream(fileName, FileMode.CreateNew, FileAccess.Write);

StreamWriter writer = new StreamWriter(output);

foreach (Account a in list)

{

writer.WriteLine("{0},{1},{2},{3}", a.AccountID, a.FirstName, a.LastName, a.Balance);

}

writer.Close();

output.Close();

}

catch (IOException e)

{

Console.WriteLine(e.Message);

}

}

public void LoadFile()

{

Console.Write("Input file name to load: ");

string fileName = "";

fileName = Console.ReadLine();

if (fileName == null || fileName.Trim() == "")

return;

list.Clear();

try

{

FileStream input = new FileStream(fileName, FileMode.Open, FileAccess.Read);

StreamReader reader = new StreamReader(input);

string str;

while ((str = reader.ReadLine()) != null)

{

string[] l = str.Split(',');

Account a = new Account(int.Parse(l[0]), l[1], l[2], decimal.Parse(l[3]));

list.Add(a);

}

input.Close();

reader.Close();

}

catch (IOException e)

{

Console.WriteLine(e.Message);

}

}

public void Report()

{

Console.WriteLine("\*\*\*\*\*\*\*List account\*\*\*\*\*\*\*");

foreach (Account a in list)

{

a.Show();

}

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

public void InputList()

{

int n;

Console.Write("Number of accounts: ");

n = int.Parse(Console.ReadLine());

while (n > 0)

{

NewAccount();

n--;

}

}

public int RemoveAccount()

{

int id;

Console.Write("Input account ID to remove: ");

id = int.Parse(Console.ReadLine());

Account tmp = new Account();

tmp.AccountID = id;

int index = 1;// list.BinarySearch(tmp, new AccoutIDComparer());

if (index < 0)

{

Console.WriteLine("Account ID not in list!");

return 0;

}

else

{

list.RemoveAt(index);

Console.WriteLine("Remove success!!! ");

return 1;

}

}

public void SortByAccountID()

{

Console.WriteLine("Sort by account ID");

//list.Sort(new AccoutIDComparer());

}

public void SortByFirstName()

{

Console.WriteLine("Sort by first name");

list.Sort(new FirstNameComparer());

}

public void SortByBalance()

{

Console.WriteLine("Sort by balance");

//list.Sort(new BalanceComparer());

}

}

class Program3and4

{

public static void Main(string[] args)

{

AccountList accounts = new AccountList();

accounts.InputList();

accounts.Report();

accounts.SaveFile();

accounts.LoadFile();

accounts.Report();

accounts.RemoveAccount();

accounts.Report();

accounts.SortByAccountID();

accounts.Report();

accounts.SortByFirstName();

accounts.Report();

accounts.SortByBalance();

accounts.Report();

}

}

}

**LAB3**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

int numA = int.Parse(txtNumA.Text);

int numB = int.Parse(txtNumB.Text);

int rs =numA + numB;

txtResult.Text = rs.ToString();

}

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

{

Close();

}

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

{

txtNumA.Text = "";

txtNumB.Text = "";

txtResult.Text = "";

}

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

{

int numA = int.Parse(txtNumA.Text);

int numB = int.Parse(txtNumB.Text);

int rs = numA - numB;

txtResult.Text = rs.ToString();

}

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

{

int numA = int.Parse(txtNumA.Text);

int numB = int.Parse(txtNumB.Text);

int rs = numA \* numB;

txtResult.Text = rs.ToString();

}

private void button4\_Click(object sender, EventArgs e)

{

int numA = int.Parse(txtNumA.Text);

int numB = int.Parse(txtNumB.Text);

float rs = (float)numA / numB;

txtResult.Text = rs.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab3

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

rbMax.Checked = true;

}

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

{

int numA = int .Parse(txtA.Text);

int numB = int .Parse(txtB.Text);

int A = 0 ;

if(rbMax.Checked == true)

{

if(numA > numB)

{

A = numA;

MessageBox.Show($"Max la:{A}");

}

else

{

A = numB;

MessageBox.Show($"Max la:{A}");

}

}

if (rbMin.Checked == true)

{

if (numA < numB)

{

A = numA;

MessageBox.Show($"Min la:{A}");

}

else

{

A = numB;

MessageBox.Show($"Min la:{A}");

}

}

}

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

{

txtA.Text = "";

txtB.Text = "";

}

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

{

Close();

}

}

}

**LAB4**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

private void Input\_check(object sender, EventArgs e)

{

bool CheckTextBox = true;

bool CheckButton = true;

foreach (var item in this.Controls)

{

if (item is Panel)

{

Panel pnl = (Panel)item;

if (pnl.Name == "panel1")

{

foreach (Control ctr in pnl.Controls)

{

if (txtName.Text == "")

{

CheckTextBox = false;

}

if (ctr is GroupBox)

{

GroupBox gro = (GroupBox)ctr;

foreach (Control ctr1 in gro.Controls)

{

if (ctr1 is CheckBox)

{

CheckBox chk = (CheckBox)ctr1;

if (chk.Checked == true)

{

if (chk.Text.Contains("Fillings"))

{

if (numericUpDown1.Value == 0)

{

MessageBox.Show("Check the number of fillings again!", "Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);

}

}

CheckButton = false;

}

}

}

}

}

if (CheckTextBox == false || (CheckTextBox == true && CheckButton == true))

{

MessageBox.Show("Check the infomation again!", "Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);

}

}

}

}

Payment\_GetPay(sender, e);

}

private void Payment\_GetPay(object sender, EventArgs e)

{

int total = 0;

foreach (var item in this.Controls)

{

if (item is Panel)

{

Panel pnl = (Panel)item;

if (pnl.Name == "panel1")

{

foreach (Control ctr in pnl.Controls)

{

if (ctr is GroupBox)

{

GroupBox gro = (GroupBox)ctr;

foreach (Control ctr1 in gro.Controls)

{

if (ctr1 is CheckBox)

{

CheckBox chk = (CheckBox)ctr1;

if (chk.Checked)

{

foreach (var item1 in gro.Controls)

{

if (item1 is Label)

{

Label itemlb = (Label)item1;

if (itemlb.TabIndex == chk.TabIndex)

{

if (itemlb.Text.Contains("Item"))

{

foreach (var itemupdown in gro.Controls)

{

if (itemupdown is NumericUpDown)

{

NumericUpDown numeric = (NumericUpDown)itemupdown;

if (numeric.TabIndex == itemlb.TabIndex)

{

string a = itemlb.Text.Replace("$", "").Replace(",", "").Replace("/Item", "");

total += Convert.ToInt32(a) \* (int)numeric.Value;

}

}

}

}

else

{

string a = itemlb.Text.Replace("$", "").Replace(",", "");

total += Convert.ToInt32(a);

}

}

}

}

}

}

}

}

}

}

}

}

txtTotal.Text = total.ToString();

}

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

{

Input\_check(sender, e);

}

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

{

string Text1 = "";

foreach (var item in this.Controls)

{

if (item is Panel)

{

Panel pnl = (Panel)item;

if (pnl.Name == "panel1")

{

foreach (Control ctr in pnl.Controls)

{

if (ctr is TextBox)

{

TextBox txt = (TextBox)ctr;

if (txt.Text != "")

{

txt.Text = Text1;

}

}

if (ctr is GroupBox)

{

GroupBox gro = (GroupBox)ctr;

foreach (Control ctr1 in gro.Controls)

{

if (ctr1 is CheckBox)

{

CheckBox chk = (CheckBox)ctr1;

if (chk.Checked)

{

chk.Checked = false;

}

}

if (ctr1 is NumericUpDown)

{

NumericUpDown numberic = (NumericUpDown)ctr1;

numberic.Value = 0;

}

}

}

}

}

}

}

}

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

{

if (txtName.Text != "" && txtTotal.Text != "")

lst\_Add.Items.Add("Name: " + txtName.Text + "\tTotal:" + txtTotal.Text);

else

{

if (txtName.Text != "" || txtTotal.Text != "")

{

MessageBox.Show("Check Name or Total agaim!", "Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);

}

}

txtTotal.Text = "";

}

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

{

string filepath = "E:\\CustomerBillingList.txt";

FileStream fs = new FileStream(filepath, FileMode.Create);

StreamWriter sWriter = new StreamWriter(fs, Encoding.UTF8);

for (int i = lst\_Add.Items.Count - 1; i >= 0; i--)

sWriter.WriteLine(lst\_Add.Items[i].ToString());

sWriter.Flush();

fs.Close();

}

//Read File:

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

{

string filepath = "E:\\CustomerBillingList.txt";

MessageBox.Show((string)File.ReadAllText(filepath));

}

public Form1(string a, string b, string c) : this()

{

label4.Text = a;

label5.Text = b;

label6.Text = c;

}

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

{

Form2 editPrice = new Form2(label4.Text, label5.Text, label6.Text);

this.Hide();

editPrice.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_4

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public Form2(string price, string price1, string price2) : this()

{

textBox1.Text = price;

textBox2.Text = price1;

textBox3.Text = price2;

}

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

{

Form1 f = new Form1(textBox1.Text, textBox2.Text, textBox3.Text);

this.Hide();

f.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_4

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private static bool IsNumber(string val)

{

if (val != "")

{

return Regex.IsMatch(val, @"^[0-9]\d\*\.?[0-9]\*$");

}

else return true;

}

private void btn\_Update\_Click(object sender, EventArgs e)

{

foreach (var item in this.Controls)

{

if (item is TextBox)

{

TextBox txt = (TextBox)item;

if (IsNumber(txt.Text) != true)

{

MessageBox.Show("Check input number again", "Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);

txt.Text = "";

}

else

{

lst\_Numbers.Items.Add(txt.Text);

txt.Text = "";

}

}

}

}

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

{

lst\_Numbers.Items.Clear();

}

private void btn\_Prime\_Click(object sender, EventArgs e)

{

if (lst\_Numbers.Items.Count > 0)

{

double[] b = new double[lst\_Numbers.Items.Count + 1];

int n = 0;

double a = 0;

for (int i = 0; i < lst\_Numbers.Items.Count; i++)

{

bool IsPrime = true;

a = Convert.ToDouble(lst\_Numbers.Items[i].ToString());

if (a < 2)

{

IsPrime = false;

}

for (int j = 0; j < a / 2; j++)

{

if (a % 2 == 0)

{

IsPrime = false; break;

}

}

if (IsPrime)

{

b[n] = a;

n++;

}

}

lst\_Numbers.Items.Clear();

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

lst\_Numbers.Items.Add(b[i].ToString());

}

}

private void btn\_Div3\_Click(object sender, EventArgs e)

{

if (lst\_Numbers.Items.Count > 0)

{

int[] a = new int[lst\_Numbers.Items.Count + 1];

int n = 0;

for (int i = 0; i < lst\_Numbers.Items.Count; i++)

{

int d = Convert.ToInt32(lst\_Numbers.Items[i].ToString());

if (d % 3 == 0)

{

a[n] = d;

n++;

}

}

lst\_Numbers.Items.Clear();

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

lst\_Numbers.Items.Add(a[i].ToString());

}

}

private void btn\_Min\_Click(object sender, EventArgs e)

{

if (lst\_Numbers.Items.Count > 0)

{

int[] a = new int[lst\_Numbers.Items.Count + 1];

int n = 0;

int min = Convert.ToInt32(lst\_Numbers.Items[0].ToString());

for (int i = 1; i < lst\_Numbers.Items.Count; i++)

{

int d = Convert.ToInt32(lst\_Numbers.Items[i].ToString());

if (d <= min)

{

min = d;

}

}

lst\_Numbers.Items.Clear();

lst\_Numbers.Items.Add(min.ToString());

}

}

private void btn\_Max\_Click(object sender, EventArgs e)

{

if (lst\_Numbers.Items.Count > 0)

{

int[] a = new int[lst\_Numbers.Items.Count + 1];

int n = 0;

int max = Convert.ToInt32(lst\_Numbers.Items[0].ToString());

for (int i = 1; i < lst\_Numbers.Items.Count; i++)

{

int d = Convert.ToInt32(lst\_Numbers.Items[i].ToString());

if (max <= d)

{

max = d;

}

}

lst\_Numbers.Items.Clear();

lst\_Numbers.Items.Add(max.ToString());

}

}

private void btnDiv2\_Click(object sender, EventArgs e)

{

if (lst\_Numbers.Items.Count > 0)

{

int[] a = new int[lst\_Numbers.Items.Count + 1];

int n = 0;

for (int i = 0; i < lst\_Numbers.Items.Count; i++)

{

int d = Convert.ToInt32(lst\_Numbers.Items[i].ToString());

if (d % 2 == 0)

{

a[n] = d;

n++;

}

}

lst\_Numbers.Items.Clear();

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

lst\_Numbers.Items.Add(a[i].ToString());

}

}

}

}

**LAB5**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lap\_5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

List<DataTable> lst = new List<DataTable>();

List<DataTable> lstTotal = new List<DataTable>();

DataTable tbOrder;

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

{

DialogResult dialog = MessageBox.Show("Do you have Open App?", "Question!", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialog != DialogResult.Yes)

{

Application.ExitThread();

}

cmbTable.Items.Add("Table...");

cmbTable.SelectedIndex = 0;

for (int i = 0; i < 10; i++)

{

cmbTable.Items.Add("Table " + (i + 1));

}

}

private void btnFood(object o, EventArgs e)

{

tbOrder = new DataTable();

tbOrder.Columns.Add("FoodName");

tbOrder.Columns.Add("Amount");

tbOrder.Columns.Add("TableName");

DataRow dr;

dr = tbOrder.NewRow();

string text = ((Button)o).Text;

DataTable dtTotal = new DataTable();

for (int i = 0; i < lst.Count; i++)

{

DataRow data = lst[i].Rows[0];

if (text == data["FoodName"].ToString())

{

int sl = Convert.ToInt32(lst[i].Rows[0]["Amount"]);

lst[i].Rows[0]["Amount"] = sl + 1;

for (int j = 0; j < lst.Count; j++)

{

dtTotal.Merge(lst[j]);

}

dgvFood.DataSource = dtTotal;

return;

}

}

dr["FoodName"] = text;

dr["Amount"] = 1;

dr["TableName"] = cmbTable.Text;

tbOrder.Rows.Add(dr);

lst.Add(tbOrder);

for (int i = 0; i < lst.Count; i++)

{

dtTotal.Merge(lst[i]);

}

dgvFood.DataSource = dtTotal;

}

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

{

try

{

if (dgvFood.Rows[1].Cells[1].ToString().Trim().Length > 0)

{

MessageBox.Show("Please Order!!!", "Notify", MessageBoxButtons.OK);

cmbTable.Text = lst[0].Rows[0]["TableName"].ToString();

}

}

catch

{

}

}

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

{

DataTable dt = new DataTable();

dt.Columns.Add("CustomerName");

dt.Columns.Add("TableName");

dt.Columns.Add("Foods");

DataRow dr;

dr = dt.NewRow();

dr["CustomerName"] = txtCustomerName.Text;

dr["TableName"] = cmbTable.Text;

string foods = "";

for (int i = 0; i < lst.Count; i++)

{

foods += lst[i].Rows[0]["FoodName"] + "-";

}

dr["Foods"] = foods;

dt.Rows.Add(dr);

lstTotal.Add(dt);

DataTable dtTotal = new DataTable();

for (int i = 0; i < lstTotal.Count; i++)

{

dtTotal.Merge(lstTotal[i]);

}

dgvTotal.DataSource = dtTotal;

DataTable dtNew = new DataTable();

dtNew.Columns.Add("FoodName");

dtNew.Columns.Add("Amount");

dtNew.Columns.Add("TableName");

dgvFood.DataSource = dtNew;

txtCustomerName.Text = "";

cmbTable.Text = "";

lst.Clear();

}

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

{

DataTable dtNew = new DataTable();

dtNew.Columns.Add("FoodName");

dtNew.Columns.Add("Amount");

dtNew.Columns.Add("TableName");

dgvFood.DataSource = dtNew;

txtCustomerName.Text = "";

cmbTable.Text = "";

lst.Clear();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lap\_5

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void ResetInput()

{

txtName.Text = string.Empty;

txtAddress.Text = string.Empty;

txtPhone.Text = string.Empty;

dtpkDOB.Text = DateTime.Now.ToString();

}

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

{

foreach (var item in gbxInformations.Controls)

{

if (item is TextBox)

{

TextBox itTextbox = (TextBox)item;

if (itTextbox.Text.Trim().Length == 0)

{

MessageBox.Show("Error input" + itTextbox.Name + "!!!", "Error", MessageBoxButtons.RetryCancel, MessageBoxIcon.Error);

itTextbox.Focus();

return;

}

}

}

ListViewItem listView = new System.Windows.Forms.ListViewItem(new String[] { txtName.Text, dtpkDOB.Text, txtPhone.Text, txtAddress.Text });

lvStaff.Items.Add(listView);

ResetInput();

txtName.Focus();

}

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

{

int total = lvStaff.Items.Count;

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

{

if (lvStaff.Items[i].Text == txtName.Text)

{

lvStaff.Items[i].SubItems[1].Text = dtpkDOB.Text;

lvStaff.Items[i].SubItems[2].Text = txtPhone.Text;

lvStaff.Items[i].SubItems[3].Text = txtAddress.Text;

break;

}

}

txtName.Enabled = true;

ResetInput();

txtName.Focus();

}

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

{

if (lvStaff.SelectedItems.Count > 0)

{

lvStaff.Items.Remove(lvStaff.SelectedItems[0]);

}

}

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

{

this.Close();

}

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

{

if (lvStaff.SelectedItems.Count > 0)

{

ListViewItem item = lvStaff.SelectedItems[0];

txtName.Text = item.SubItems[0].Text;

dtpkDOB.Text = item.SubItems[1].Text;

txtAddress.Text = item.SubItems[2].Text;

txtPhone.Text = item.SubItems[3].Text;

}

//else

//{

// txtName.Text = string.Empty;

// dtpkDOB.Text = string.Empty;

// txtAddress.Text = string.Empty;

// txtPhone.Text = string.Empty;

//}

}

}

}