אופק לוי

שאלה 1

הסיפור

ארגון שמכיל בתי ספר, חנויות, ורשימת חנויות קסטרו. ניתן להוסיף\להסיר איברים לארגון, לבצע חיפוש לפי שם, להציג מידע על ארגון ועוד.

המחלקות

ASystem

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

abstract class ASystem

{

string sname;

DateTime established;

public string SName

{

get { return sname; }

set

{

if (value == null) throw new ArgumentNullException("Null name");

sname = value;

}

}

public DateTime Established

{

get { return established; }

set

{

if (value == null) throw new ArgumentNullException("Null DateTime Value is not allowed");

established = value;

}

}

public ASystem(string name, DateTime established)

{ SName = name; Established = established; }

public ASystem(string name)

{ SName = name; established = DateTime.Now; }

// string with information about system usage

public abstract string systemPurpose();

public override string ToString()

{ return sname + " exist since " + established.ToShortDateString(); }

}

}

Castro

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

internal class Castro : Store

{

//t shirts list

private Stack<string> closetTshirts = new Stack<string>();

//is companny owned by the country

private bool ownedByCountry;

//constructor

public Castro(string name, DateTime established, string managerName, bool ownedByCountry) : base(name, established, managerName)

{

OwnedByCountry = ownedByCountry;

}

//constructor

public Castro(string name, DateTime established, string managerName, Stack<string> closetTshirts) : base(name, established, managerName)

{

ClosetTshirts = closetTshirts;

}

//add adding t shirts to the stack

public void AddTShirstFromArray(string[] tshirts)

{

for (int i = 0; i < tshirts.Length; i++)

ClosetTshirts.Push(tshirts[i]);

}

//add adding t shirt to the stack

public void Add(string t)

{

ClosetTshirts.Push(t);

}

//remove n items from stack

public void RemoveNItems(int n)

{

for (int i = 0; i < n && ClosetTshirts.Count > 0; i++)

ClosetTshirts.Pop();

}

//is store owned by country

public bool OwnedByCountry

{

get

{

return ownedByCountry;

}

set

{

ownedByCountry = value;

}

}

//get and set to t shirts stack

public Stack<string> ClosetTshirts

{

get

{

return closetTshirts;

}

set

{

if (value.Peek() == "")

throw new Exception("stack peek cannot be empty string");

closetTshirts = value;

}

}

//return to string

public override string ToString()

{

return base.ToString() + " castro num of t shirts: " + ClosetTshirts.Count + ",Owned by couontry: " + OwnedByCountry;

}

//prints the tostring

public override void Print()

{

Console.WriteLine(ToString());

}

//comoare by number os shirts

public override int CompareTo(object obj)

{

if (obj == null) return 1;

Castro s = obj as Castro;

if (s != null)

return ClosetTshirts.Count.CompareTo(s.ClosetTshirts.Count);

else

throw new ArgumentException("Object is not a castro store");

}

}

}

IStock

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

interface IStock

{

void AddWorth();//add worth to the store

void SubstructWorth();//substructe worth from the store

}

}

ItemsRemoveable

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

interface ItemsRemoveable

{

void RemoveLastAdded();//enable to remove items

}

}

Organizations

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

class Organizations

{

private List<Castro> castros;//all castro stores

private List<Store> stores;//all general stores

private List<School> schools;//all schools

//constructor

public Organizations()

{

Castros = new List<Castro>();

Stores = new List<Store>();

Schools = new List<School>();

}

//constructor

public Organizations(List<Castro> castros, List<Store> stores, List<School> schools)

{

this.Castros = castros;

this.Stores = stores;

this.Schools = schools;

}

//constructor

public Organizations(List<Castro> castros)

{

this.Castros = castros;

Stores = new List<Store>();

Schools = new List<School>();

}

//search school by manager name

public School SearchSchool(string managerName)

{

foreach (School s in Schools)

if (s.ManagerName == managerName)

return s;

return null;

}

//get/set to schools list

public List<School> Schools

{

get

{

return schools;

}

set

{

if (value.Count > 1000)

throw new Exception("Schools cannot be larger the 1K");

schools = value;

}

}

//get/set to stores list

public List<Store> Stores

{

get

{

return stores;

}

set

{

if (value.Count > 1000)

throw new Exception("Stores cannot be larger the 1K");

stores = value;

}

}

//get/set to castros list

public List<Castro> Castros

{

get

{

return castros;

}

set

{

if (value.Count>1000)

throw new Exception("Castros cannot be larger the 1K");

castros = value;

}

}

//add castro store

public void AddCastro(Castro c)

{

Castros.Add(c);

}

//remove last added castro store

public void RemoveLastCastro()

{

Castros.RemoveAt(castros.Count-1);

}

//add general store

public void AddStroe(Store s)

{

Stores.Add(s);

}

//remove last added general store

public void RemoveLastStore()

{

Stores.RemoveAt(stores.Count - 1);

}

//add school

public void AddSchool(School s)

{

Schools.Add(s);

}

//remove last added school

public void RemoveLastSchool()

{

Schools.RemoveAt(schools.Count - 1);

}

//print organization information

public void Print()

{

foreach (School s in Schools)

s.Print();

foreach (Castro c in Castros)

c.Print();

foreach (Store s in Stores)

s.Print();

}

}

}

School

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

internal class School : ASystem, ItemsRemoveable

{

private string managerName;

private int numOfStudents;

private Student[] students = new Student[250];

//contructor

public School(string name, DateTime established, string managerName) : base(name, established)

{

this.managerName = managerName;

numOfStudents = 0;

}

//contructor

public School(string name, DateTime established, string managerName, int numOfStudents) : base(name, established)

{

this.managerName = managerName;

this.numOfStudents = numOfStudents;

}

//contructor

public School(string name, DateTime established, int numOfStudents) : base(name, established)

{

this.managerName = "default";

this.numOfStudents = numOfStudents;

}

//print school info

public void Print()

{

Console.WriteLine(ToString());

}

//compare to by date established

public int CompareTo(object obj)

{

if (obj == null) return 1;

School s = obj as School;

if (s != null)

return Established.CompareTo(s.Established);

else

throw new ArgumentException("Object is not a School");

}

//return to string

public override string ToString()

{

return base.ToString() + " Manager name: " + ManagerName + ", Number of s tudents: " + NumOfStudents;

}

//get set students array

public Student[] Students

{

get

{

return students;

}

set

{

if (value.Length == 0)

throw new Exception("students array cannot be with length of zero");

students = value;

}

}

//remove last added student

public void RemoveLastAdded()

{

Students[NumOfStudents] = null;

}

//add a student

public void AddStudent(Student s)

{

if (NumOfStudents >= 250)

throw new ArgumentException("school is full with students");

Students[NumOfStudents++] = s;

}

//get / set manager name

public string ManagerName

{

get

{

return managerName;

}

set

{

if (value.Length == 0)

throw new Exception("manager name cannot be empty");

managerName = value;

}

}

// get set number of students

public int NumOfStudents

{

get

{

return numOfStudents;

}

set

{

if (value < 0)

throw new Exception("num of students cannot br negative");

numOfStudents = value;

}

}

// get system purpose

public override string systemPurpose()

{

return "High school";

}

}

}

Store

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

internal class Store : ASystem, IStock

{

private string managerName;

private int worth;

//contructor

public Store(string name, DateTime established, string managerName) : base(name, established)

{

this.managerName = managerName;

worth = 0;

}

//contructor

public Store(string name, DateTime established, string managerName, int worth) : base(name, established)

{

this.managerName = managerName;

this.worth = worth;

}

//contructor

public Store(string name, DateTime established, int worth) : base(name, established)

{

this.managerName = "default store";

this.worth = worth;

}

//comapre to by worth value

public virtual int CompareTo(object obj)

{

if (obj == null) return 1;

Store s = obj as Store;

if (s != null)

return Worth.CompareTo(s.Worth);

else

throw new ArgumentException("Object is not a store");

}

//return tostring

public override string ToString()

{

return base.ToString() + " Store name: " + StoreName + ", Store worth: " + Worth;

}

// get / set worth value

public int Worth

{

get

{

return worth;

}

set

{

if (value > 1000000)

throw new Exception("Worth cannot be larger the 1M");

worth = value;

}

}

// add 10 to worth value

public void AddWorth()

{

Worth += 10;

}

//print infomation

public virtual void Print()

{

Console.WriteLine(ToString());

}

//substructe from store value

public void SubstructWorth()

{

if (Worth <0)

throw new ArgumentException("Worth cannot be negative");

Worth -= 10;

}

// get set store name

public string StoreName

{

get

{

return managerName;

}

set

{

if (value.Length == 0)

throw new Exception("Store name cannot be empty");

managerName = value;

}

}

// get syste, purpose

public override string systemPurpose()

{

return "Store in stock market";

}

}

}

Student

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

class Student

{

public string Name { get; set; }

static int num = 0;

//contrucor

public Student()

{

Name = "default "+(++num);

}

}

}

Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ofek1

{

class Program

{

static void Main(string[] args)

{

//organization instance

Organizations organizations = new Organizations();

List<Castro> castros\_copy;

List<Store> stores\_copy;

List<School> schools\_copy;

Stack<string> tshirt\_copy;

//some t shirts array

string[] t\_shirsts = new string[] { "white t shirt", "red t shirt" };

for (int i = 0; i < 3; i++)

{

//castro store instance

Castro castro = new Castro("castro " + (i + 1), RandomDate(), RandomName(), RandomBool());

castro.AddTShirstFromArray(t\_shirsts);

castro.Add("t shirt "+(i+1));

castro.RemoveNItems(1);

Console.WriteLine(castro.OwnedByCountry);

tshirt\_copy = castro.ClosetTshirts;

Console.WriteLine(castro.ToString());

castro.Print();

if(organizations.Castros.Count>0)

Console.WriteLine(organizations.Castros[0].CompareTo(castro));

// store instance

Store store = new Store("store " + (i + 1), RandomDate(), RandomName());

if (organizations.Stores.Count > 0)

Console.WriteLine(organizations.Stores[0].CompareTo(store));

Console.WriteLine(store.ToString());

Console.WriteLine(store.Worth);

store.AddWorth();

store.AddWorth();

store.SubstructWorth();

Console.WriteLine(store.StoreName);

Console.WriteLine(store.systemPurpose());

//school instance

School school = new School("school " + (i + 1), RandomDate(), RandomName());

school.Print();

if (organizations.Schools.Count > 0)

Console.WriteLine(organizations.Schools[0].CompareTo(school));

Console.WriteLine(school.ToString());

Console.WriteLine(school.Students);

school.AddStudent(new Student());

school.AddStudent(new Student());

school.RemoveLastAdded();

Console.WriteLine(school.ManagerName);

Console.WriteLine(school.NumOfStudents);

Console.WriteLine(school.systemPurpose());

organizations.AddCastro(castro);

organizations.AddStroe(store);

organizations.AddSchool(school);

School s = organizations.SearchSchool("some manager");

if(s!=null)

{

Console.WriteLine("found school with manger name");

}

else

{

Console.WriteLine(" school with manger name didn't found");

}

}

//print organization info

organizations.Print();

castros\_copy = organizations.Castros;

stores\_copy = organizations.Stores;

schools\_copy = organizations.Schools;

for (int i = 0; i < 3; i++)

{

organizations.RemoveLastSchool();

organizations.RemoveLastCastro();

organizations.RemoveLastStore();

}

organizations.Print();

Console.ReadKey();

}

static Random rand = new Random();

//generate random name

public static string RandomName()

{

string[] names = new string[] { "avi", "yossi", "david", "moshe", "abraham", "adam", "tomer", "keshet", "ben", "tom" };

return names[rand.Next(0, names.Length)];

}

//generate random number

public static int RandomNumber()

{

return rand.Next(0, 1000);

}

//generate random bool

public static bool RandomBool()

{

return rand.Next(0, 2) == 0 ? false : true;

}

//generate random date

public static DateTime RandomDate()

{

DateTime start = new DateTime(1995, 1, 1);

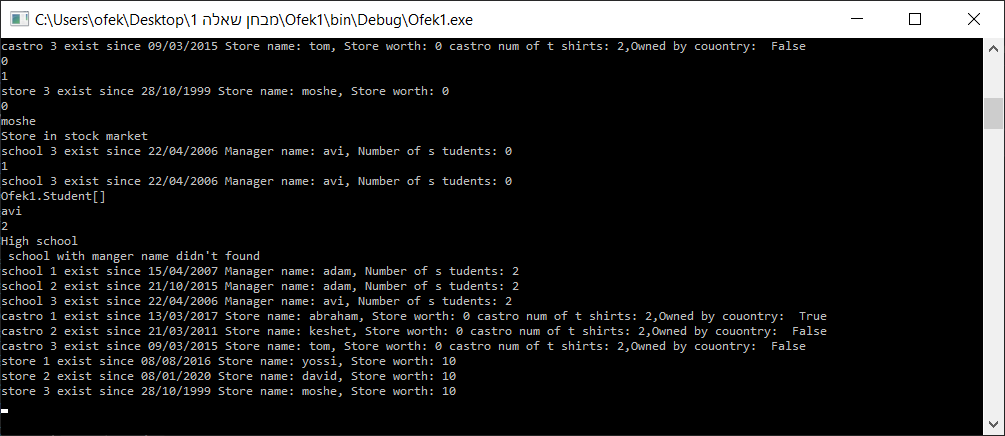
int range = (DateTime.Today - start).Days;

return start.AddDays(rand.Next(range));

}

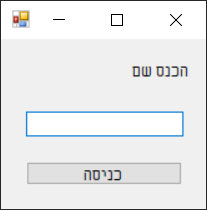
}

}

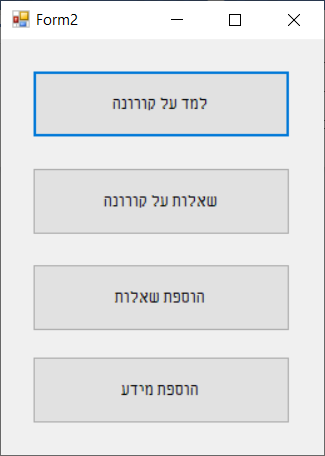
צילום פלט שאלה 1

שאלה 2

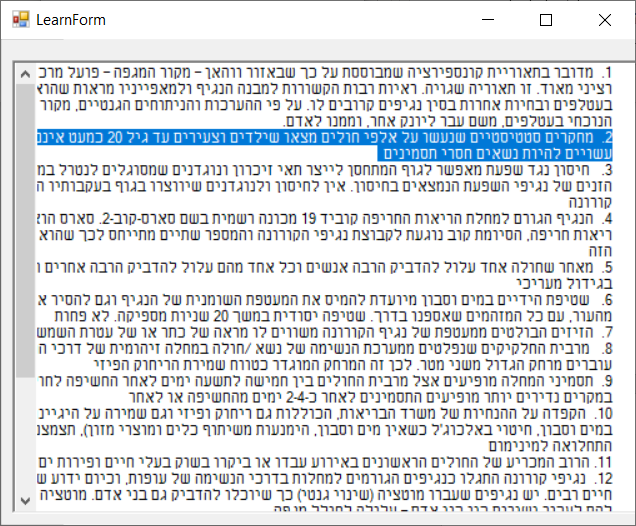
שלב הזדהות של השחקן



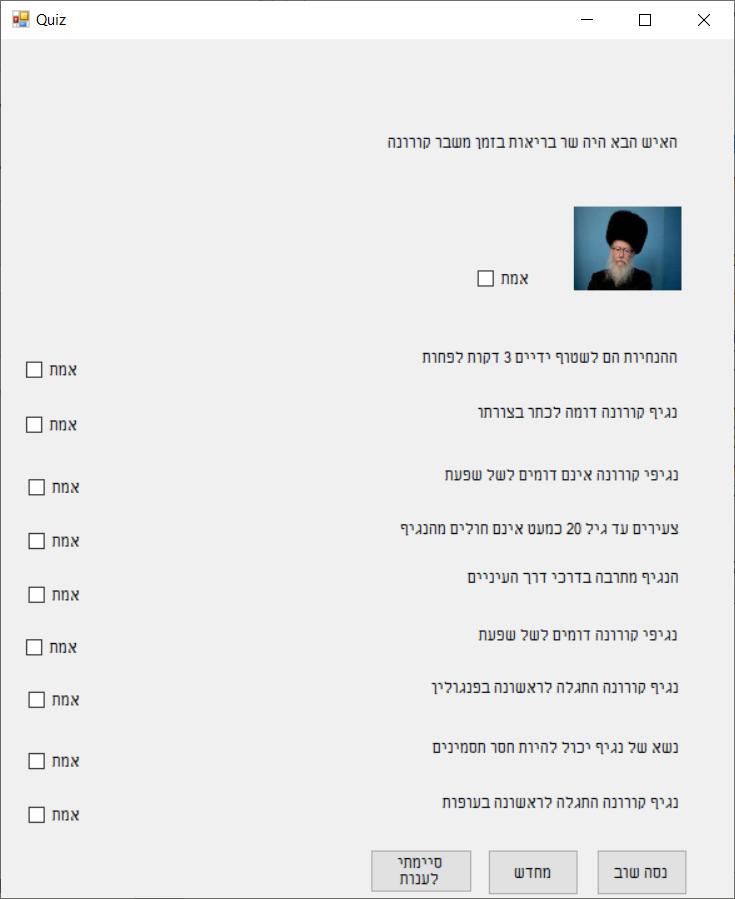
בחירה של השחקן

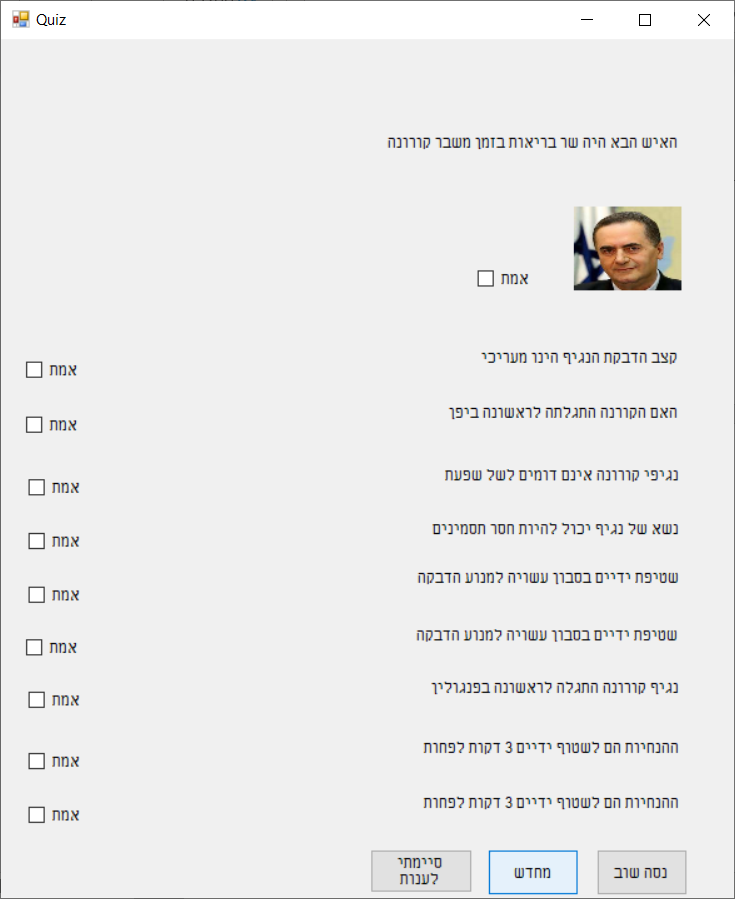


למד על קורונה

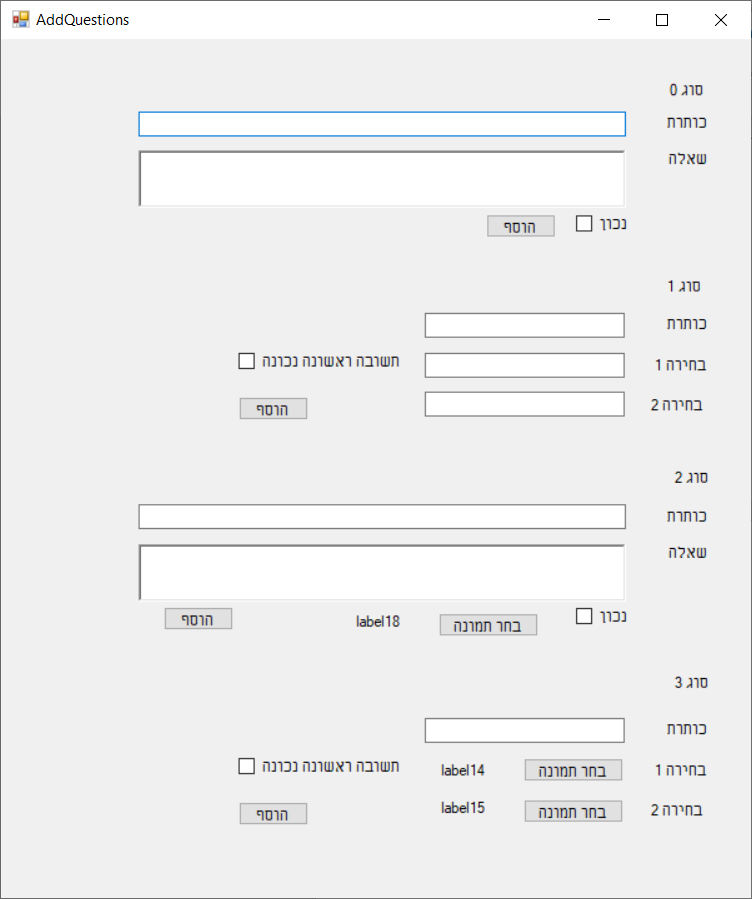


שאלות על קורונה

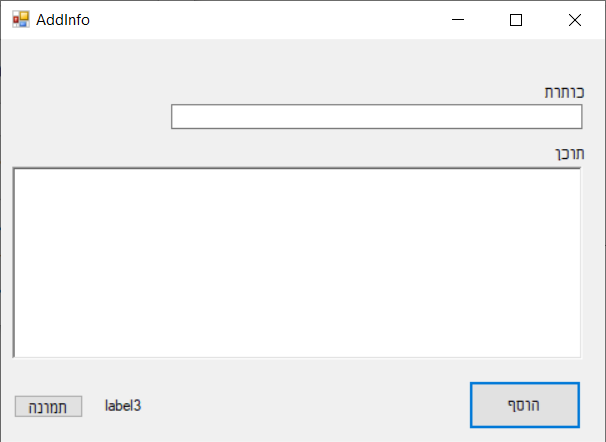




הוספת שאלה



הוספת מידע



DataItem

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace OfekProject

{

class DataItem

{

public string Num { get; set; }

public string Header { get; set; }

public string Text { get; set; }

}

}

AddInfo

namespace OfekProject

{

partial class AddInfo

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.button1 = new System.Windows.Forms.Button();

this.textBox1 = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.richTextBox1 = new System.Windows.Forms.RichTextBox();

this.label2 = new System.Windows.Forms.Label();

this.button2 = new System.Windows.Forms.Button();

this.label3 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// button1

//

this.button1.Location = new System.Drawing.Point(498, 336);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(120, 48);

this.button1.TabIndex = 0;

this.button1.Text = "הוסף";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.Add\_Click);

//

// textBox1

//

this.textBox1.Location = new System.Drawing.Point(181, 64);

this.textBox1.Name = "textBox1";

this.textBox1.Size = new System.Drawing.Size(437, 22);

this.textBox1.TabIndex = 1;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(576, 44);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(42, 17);

this.label1.TabIndex = 2;

this.label1.Text = "כותרת";

//

// richTextBox1

//

this.richTextBox1.Location = new System.Drawing.Point(12, 125);

this.richTextBox1.Name = "richTextBox1";

this.richTextBox1.Size = new System.Drawing.Size(606, 189);

this.richTextBox1.TabIndex = 3;

this.richTextBox1.Text = "";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Location = new System.Drawing.Point(586, 105);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(32, 17);

this.label2.TabIndex = 4;

this.label2.Text = "תוכן";

//

// button2

//

this.button2.Location = new System.Drawing.Point(14, 349);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(75, 23);

this.button2.TabIndex = 5;

this.button2.Text = "תמונה";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.Image\_Click);

//

// label3

//

this.label3.AutoSize = true;

this.label3.Location = new System.Drawing.Point(108, 352);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(46, 17);

this.label3.TabIndex = 6;

this.label3.Text = "label3";

//

// AddInfo

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(644, 396);

this.Controls.Add(this.label3);

this.Controls.Add(this.button2);

this.Controls.Add(this.label2);

this.Controls.Add(this.richTextBox1);

this.Controls.Add(this.label1);

this.Controls.Add(this.textBox1);

this.Controls.Add(this.button1);

this.Name = "AddInfo";

this.Text = "AddInfo";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button button1;

private System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.RichTextBox richTextBox1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Label label3;

}

}

Form1

namespace OfekProject

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.textBox1 = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.SuspendLayout();

//

// textBox1

//

this.textBox1.Location = new System.Drawing.Point(26, 71);

this.textBox1.Name = "textBox1";

this.textBox1.Size = new System.Drawing.Size(167, 22);

this.textBox1.TabIndex = 0;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(136, 23);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(57, 17);

this.label1.TabIndex = 1;

this.label1.Text = "הכנס שם";

//

// button1

//

this.button1.Location = new System.Drawing.Point(26, 120);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(167, 23);

this.button1.TabIndex = 2;

this.button1.Text = "כניסה";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(219, 168);

this.Controls.Add(this.button1);

this.Controls.Add(this.label1);

this.Controls.Add(this.textBox1);

this.Name = "Form1";

this.Text = "Form1";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Button button1;

}

}