

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.Remoting.Messaging;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AsynMethod

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

btIsComplete.Click += btIsComplete\_Click;

btEndInvoke.Click += btEndInvoke\_Click;

btCallback.Click += btCallback\_Click;

}

void btCallback\_Click(object sender, EventArgs e)

{

new Thread(AsyncMethodWithCallback).Start();

}

void btEndInvoke\_Click(object sender, EventArgs e)

{

new Thread(AsyncMethod).Start(new **Action**<IAsyncResult, **Action**<**DateTime**>>((r, d) => { d.EndInvoke(r); }));

}

void btIsComplete\_Click(object sender, EventArgs e)

{

new Thread(AsyncMethod).Start(new **Action**<IAsyncResult, **Action**<**DateTime**>>((n, d) => { while(!n.IsCompleted);}));

}

private void AsyncMethod(object obj)

{

var del = new **Action**<**DateTime**>(AddTextToTextBox);

IAsyncResult result = del.BeginInvoke(**DateTime**.Now, null, null);

(obj as **Action**<IAsyncResult, **Action**<**DateTime**>>).Invoke(result, del);

textBox.BeginInvoke(new **Action**(() => textBox.Text += **DateTime**.Now.ToShortTimeString() + Environment.NewLine));

}

private void AsyncMethodWithCallback()

{

var del = new **Action**<**DateTime**>(AddTextToTextBox);

del.BeginInvoke(**DateTime**.Now, new AsyncCallback(ResultTimeSpan), **DateTime**.Now);

}

private void ResultTimeSpan(IAsyncResult r)

{

**DateTime**? dt=r.AsyncState as **DateTime**?;

textBox.BeginInvoke(new **Action**(() => textBox.Text += "Method work - "+((**DateTime**.Now - dt)).Value.Seconds + Environment.NewLine));

}

private void AddTextToTextBox(**DateTime** start)

{

Thread.CurrentThread.IsBackground = false;

**DateTime** end = start.AddSeconds(3);

while (**DateTime**.Now < end)

{

textBox.BeginInvoke(new **Action**(() => textBox.Text += "....."));

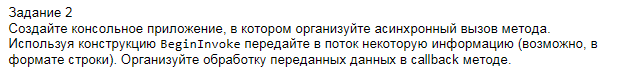
Thread.Sleep(100);

}

}

}

}



using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Remoting.Messaging;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace CallBack

{

class Program

{

static void Main(string[] args)

{

new Thread(MyThread).Start();

Console.WriteLine("Основной поток завершен");

Console.ReadKey();

}

static private void MyThread()

{

new **Func**<String>(ReadText).BeginInvoke(CallBack,null);

}

static private string ReadText()

{

TextReader tr=new StreamReader("1.txt",Encoding.Default);

return tr.ReadToEnd();

}

static private void CallBack(IAsyncResult r)

{

AsyncResult obj = r as AsyncResult;

var del = obj.AsyncDelegate as **Func**<string>;

string str = del.EndInvoke(r);

Console.WriteLine(str);

}

}

}