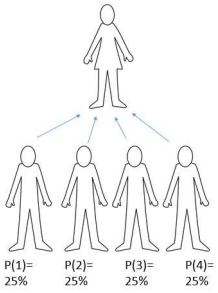
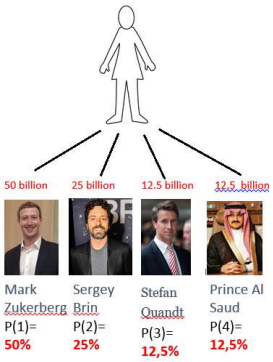
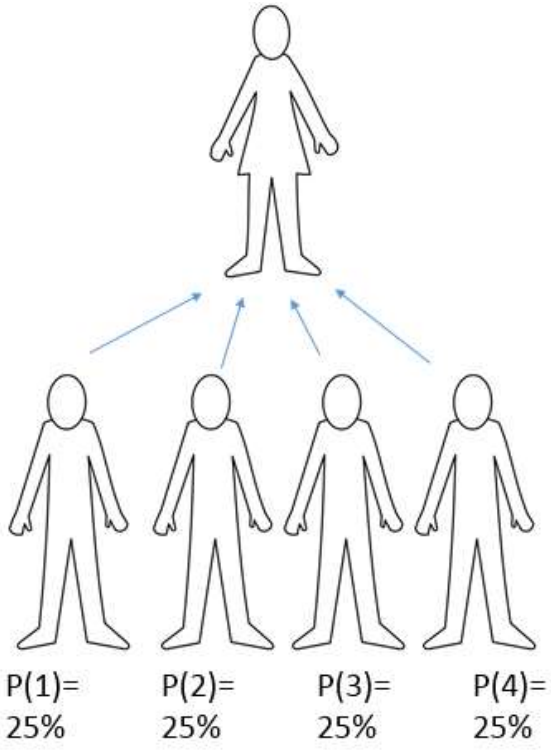
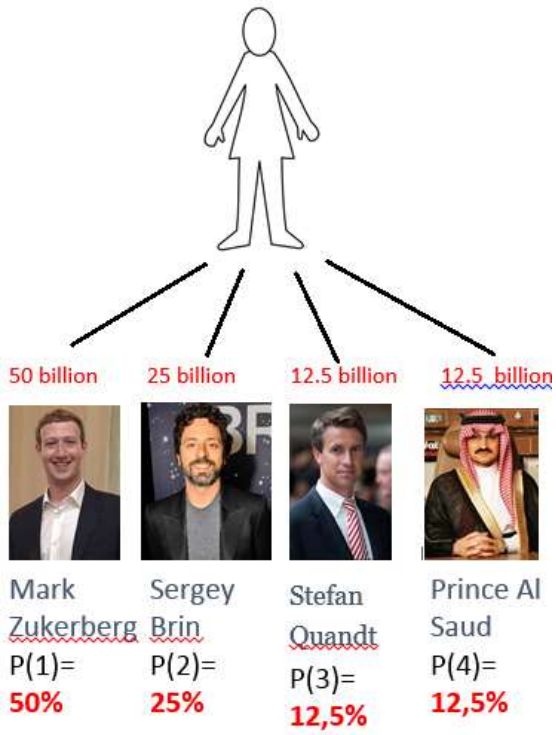
In the last lecture, we considered 2 possible selection situations

Choice under uncertainty (a pig in a poke) All men are the same	Choice under Certainty (pig crawls out of poke) All men are different
  N=4 Average number of questions = lb(N) Entropy	  $\sum_{i=1}^{N} p(i) * \log_2(\frac{1}{p(i)})$

As you can see, that when the pig gets out of the poke the problem becomes much more complicated		
lb(N) Entropy	the second formula is more complicated	$\sum_{i=1}^{N} p(i) * \log_2(\frac{1}{p(i)})$

But for those who make the choice, the situation is simplified

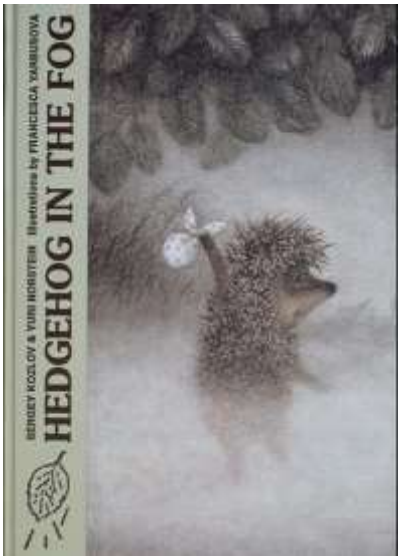


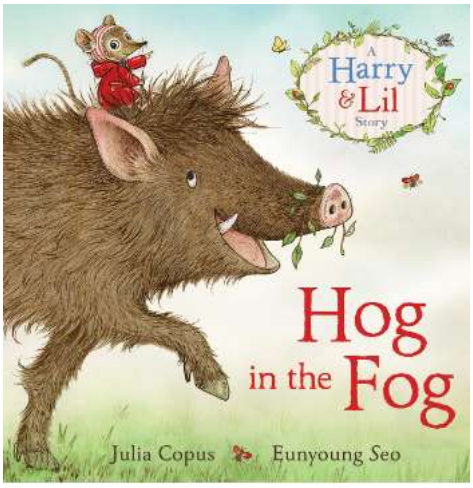


Entropy is decreasing

fog clears,

easier to make the right choice





English analogue

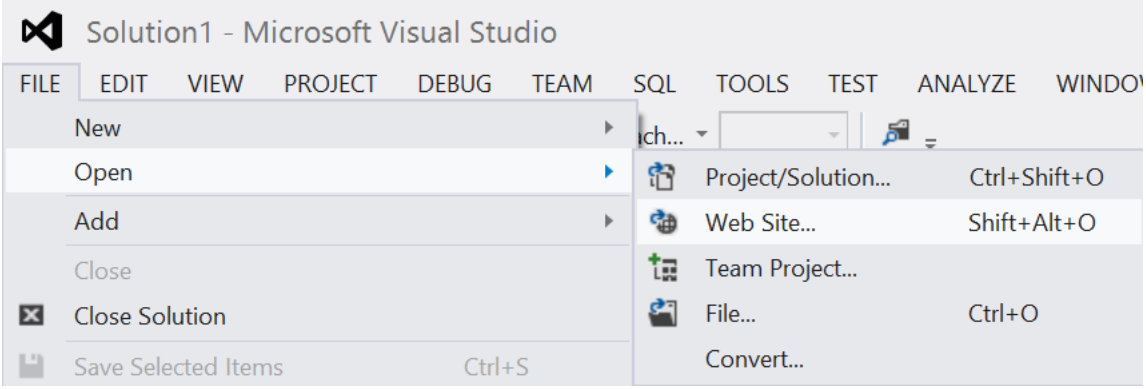
https://alanadomino.w3spaces.com/Shannon.html

- An example from part-time students of how the program should work.

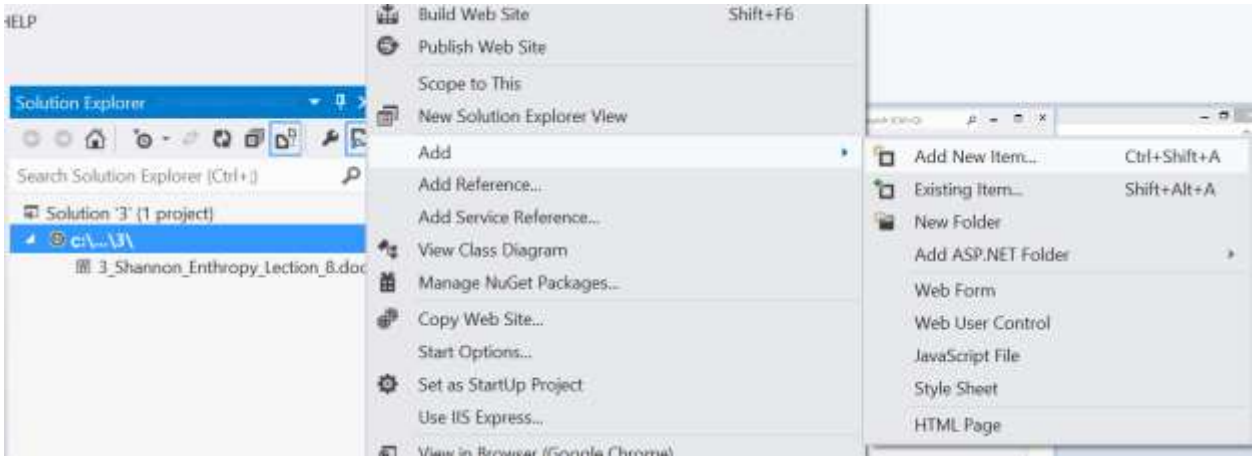
Below is an instruction on how to write such a program.

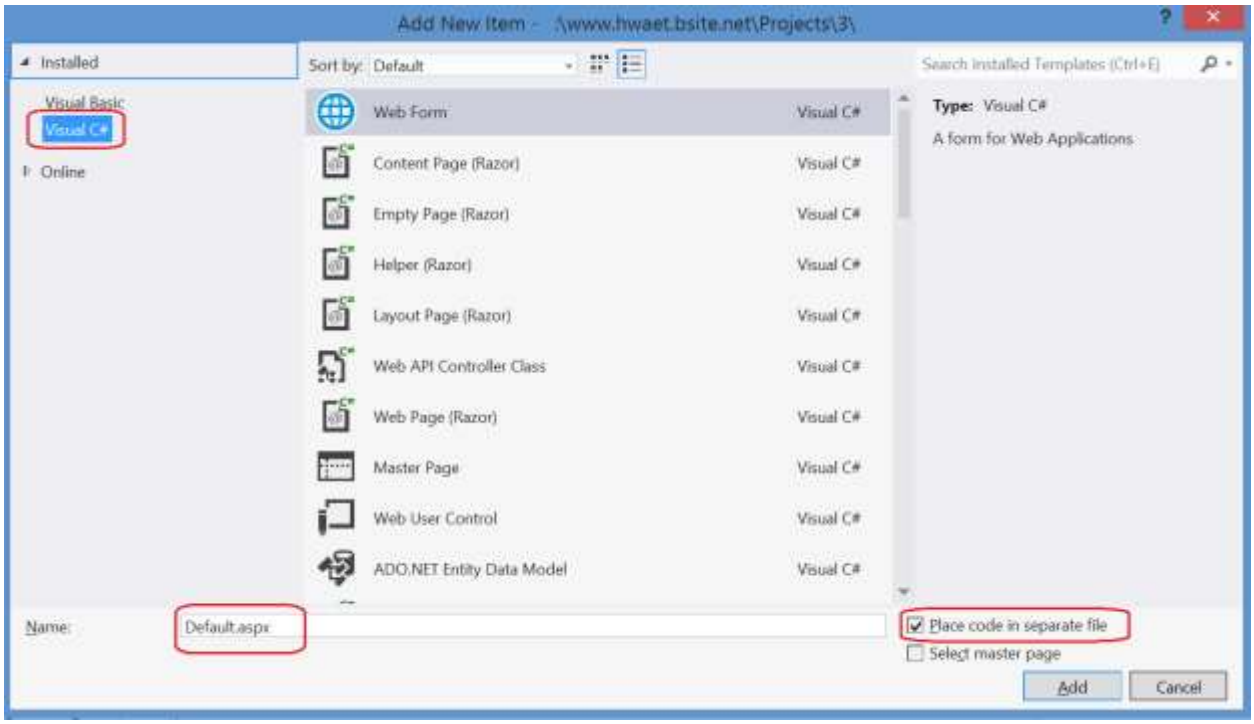
You can do with me, you can do without me on your own - it's faster.

1. Determine where this program will be located
   D:\www.hwaet.bsite.net\Projects\3\
2. Run Visual Studio
3. Open Web Site from this folder



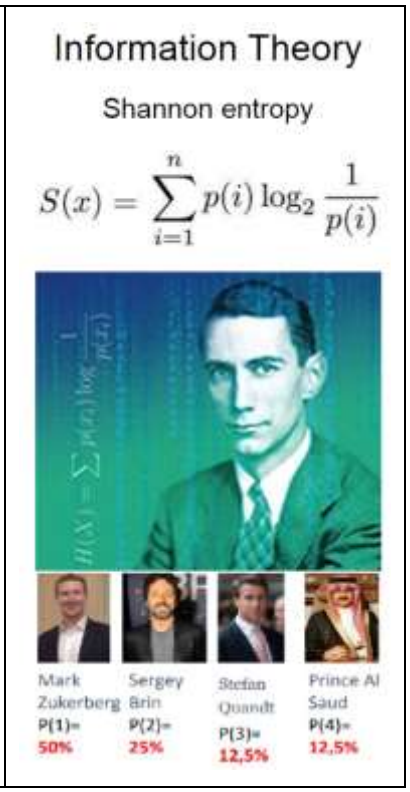
4. Create a Web Form in this folder



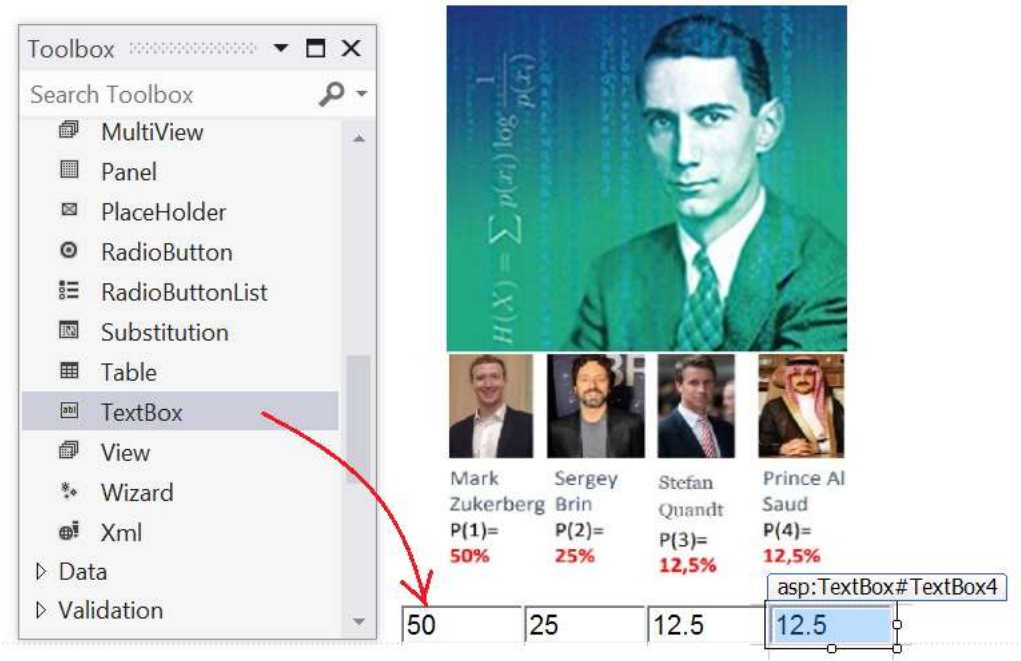


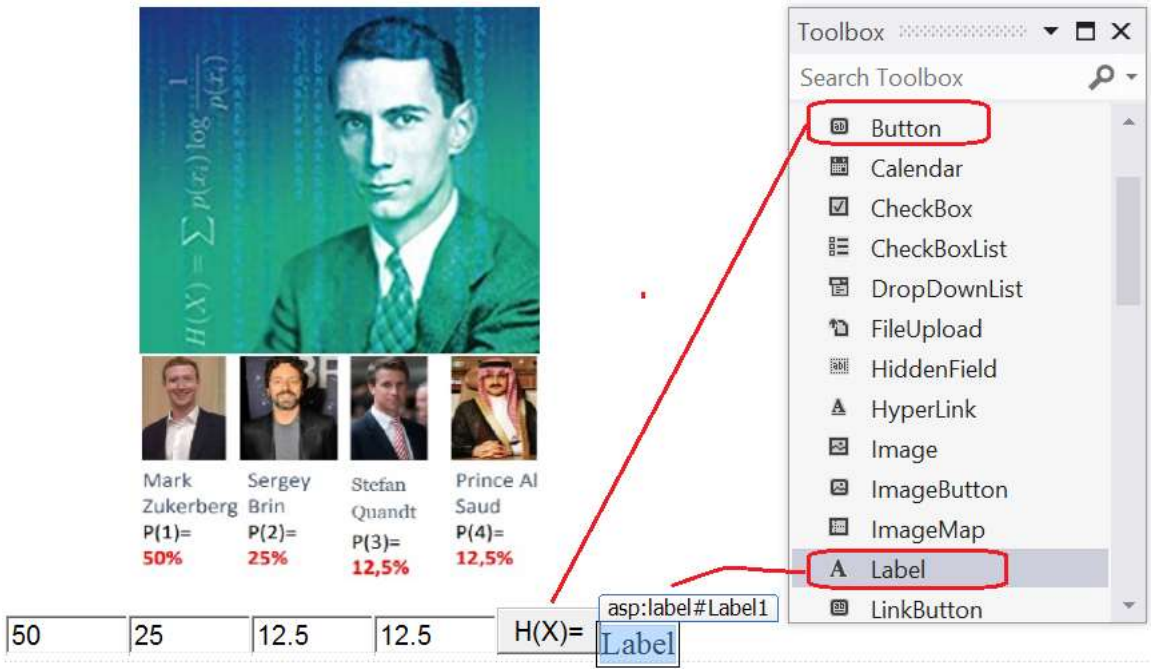
5. Bringing beauty

```
<form id="form1" runat="server">
   <div>
     <center>

       <h1 style=" font-family: sans-serif; ">
                          Information Theory</h1>
       <h2 style=" font-family: sans-serif; ">
                            Shannon entropy</h2>

        <p>
       <img alt="" src="Images/04_Shennon.jpg"
                                  width="300"/></p>
   <p>
       <img alt=""    src="Images/05_Shennon.jpg"
                                  width="300" /></p>
      </center>
   </div>
   </form>
```



6. Making 4-6 textboxes






```
<asp:TextBox ID="TextBox1" runat="server" Width="84px">50</asp:TextBox>
<asp:TextBox ID="TextBox2" runat="server" Width="84px">25</asp:TextBox>
<asp:TextBox ID="TextBox3" runat="server" Width="84px">12.5</asp:TextBox>
<asp:TextBox ID="TextBox4" runat="server" Width="84px">12.5</asp:TextBox>
<asp:Button ID="Button1" runat="server" Text="H(X)=" />

<asp:Label ID="Label1" runat="server" Text=" "></asp:Label>
```

7. The interface is ready - start writing the program

```
protected void
  Button1_Click(object sender, EventArgs e)
    {
        float[] ev = new float[4];
        ev[0] = float.Parse(TextBox1.Text);
        ev[1] = float.Parse(TextBox2.Text);
        ev[2] = float.Parse(TextBox3.Text);
        ev[3] = float.Parse(TextBox4.Text);
        double s = 0;
        double p = 0;
        for (int i = 0; i < ev.Length; i++)
        {
            p = ev[i] / 100;
            s +=  p * Math.Log((1 / p),2 );
        }
        Label1.Text = s.ToString();
```

8. Checking the work of the program

9. Separating I/O and business logic

```
protected void
  Button1_Click(object sender, EventArgs e)
    {
       float[] ev = new float[4];
       ev[0] = float.Parse(TextBox1.Text);
       ev[1] = float.Parse(TextBox2.Text);
       ev[2] = float.Parse(TextBox3.Text);
       ev[3] = float.Parse(TextBox4.Text);
       double s = 0;
       double p = 0;
       for (int i = 0; i < ev.Length; i++)
       {
           p = ev[i] / 100;
           s +=  p * Math.Log((1 / p),2 );
       }
       Label1.Text = s.ToString();
     }
```

```
public partial class _Default : System.Web.UI.Page
{
    float[] ev = new float[4];
    double s;
 protected void Button1_Click(object sender,
                                          EventArgs e)
    {
        Read();
        BusinessLogic();
        Write();
    }
    void Read()
     {
         ev[0] = float.Parse(TextBox1.Text);
         ev[1] = float.Parse(TextBox2.Text);
         ev[2] = float.Parse(TextBox3.Text);
         ev[3] = float.Parse(TextBox4.Text);
     }
    void BusinessLogic()
    {
        s = 0;
        double p = 0;
        for (int i = 0; i < ev.Length; i++)
        {
            p = ev[i] / 100;
            s = s + p * Math.Log((1 / p), 2);
        }
    }
    void Write()
```

	{
	Label1.Text = s.ToString();
	}

This separation of I / O and business logic makes it easy to rewrite the program for another type of application - WinForms or a console application

10. Migrating the program from the Web Form to the console

Default.aspx	**_Default.cs** => _Default.txt

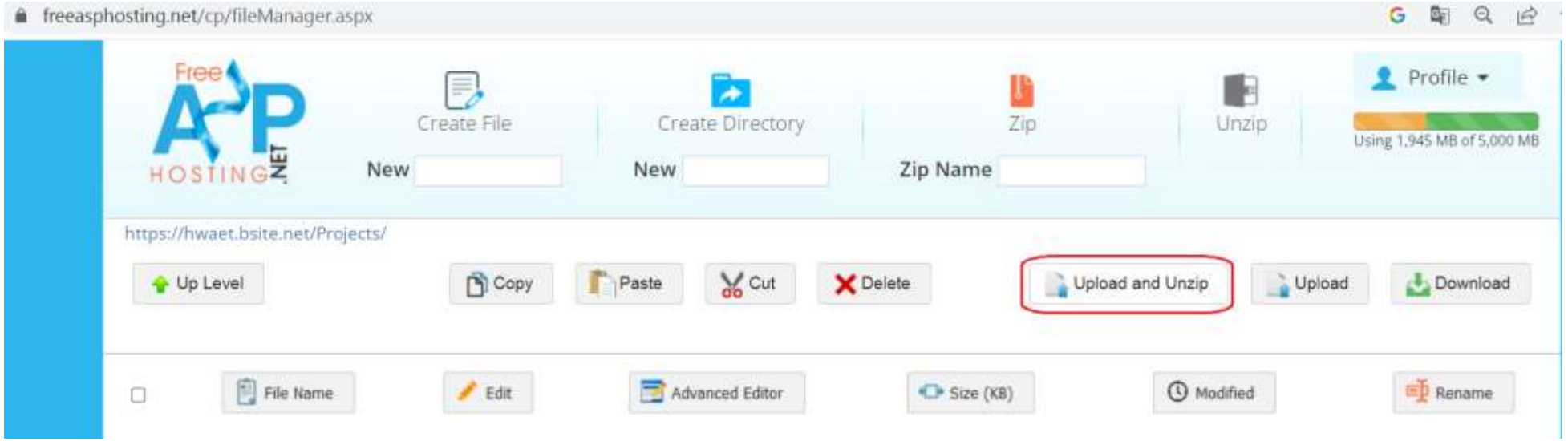
public partial class _Default :
System.Web.UI.Page
{
float[] ev = new float[4];
double s;
protected void Button1_Click(object sender,
 	
EventArgs e)
{
Read();
BusinessLogic();
Write();
}
void Read()
{
	using System;
	public class _Default
	{
	float[] ev = new float[4];
	double s;
	public static void Main(string[] args)
	{
	_Default _d=new _Default();
	// For Calling a non-static method from a static,
	// need to create an instance (object) of the class
	_d.Read(args);
	_d.BusinessLogic();
	_d.Write();
	}
	void Read(string[] args)

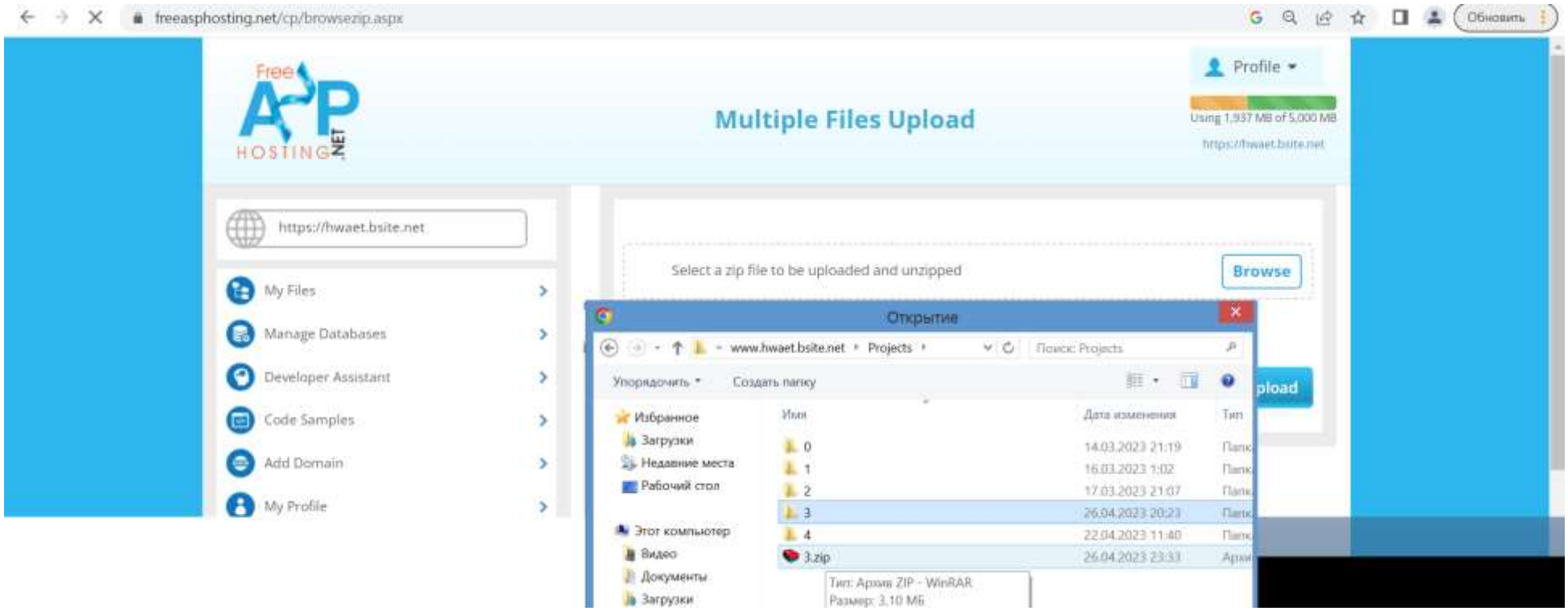
```
            ev[0] = float.Parse(TextBox1.Text);
            ev[1] = float.Parse(TextBox2.Text);
            ev[2] = float.Parse(TextBox3.Text);
            ev[3] = float.Parse(TextBox4.Text);
        }
        void BusinessLogic()
        {
            s = 0;
            double p = 0;
            for (int i = 0; i < ev.Length; i++)
            {
                p = ev[i] / 100;
                s = s + p * Math.Log((1 / p), 2);
            }
        }
        void Write()
        {
            Label1.Text = s.ToString();
        }
}
```

```
        {
            ev[0] = float.Parse(args[0]);
            ev[1] = float.Parse(args[1]);
            ev[2] = float.Parse(args[2]);
            ev[3] = float.Parse(args[3]);
        }
        void BusinessLogic()
        {
            s = 0;
            double p = 0;
            for (int i = 0; i < ev.Length; i++)
            {
                p = ev[i] / 100;
                s = s + p * Math.Log((1 / p), 2);
            }
        }
        void Write()
        {
            Console.WriteLine("H="+s.ToString());
        }
}
```

```
C:\www.hwaet.bsite.net\Projects\3>C:\Windows\Microsoft.NET\Framework\v3.5\csc _Default.cs
Компилятор Microsoft (R) Visual C# 2008 версии 3.5.30729.7903
для Microsoft (R) .NET Framework версии 3.5
(C) Корпорация Майкрософт (Microsoft Corp.). Все права защищены.


C:\www.hwaet.bsite.net\Projects\3>_Default 50 25 12,5 12,5
H=1,75

C:\www.hwaet.bsite.net\Projects\3>
```

Finalize(). Download everything to hosting




freeasphosting.net/cp/browsezip.aspx
Обновить
Free ASP HOSTING .NET
Multiple Files Upload
Profile
Using 1,937 MB of 5,000 MB
https://hwaet.bsite.net
https://hwaet.bsite.net
My Files
Manage Databases
Developer Assistant
Code Samples
Add Domain
My Profile
Select a zip file to be uploaded and unzipped
Browse
Открытие
www.hwaet.bsite.net ▸ Projects ▸
Поиск: Projects
Упорядочить
Создать папку
Избранное
Загрузки
Недавние места
Рабочий стол
Этот компьютер
Видео
Документы
Загрузки
Имя
Дата изменения
Тип
0
1
2
3
4
3.zip
14.03.2023 21:19
16.03.2023 1:02
17.03.2023 21:07
26.04.2023 20:23
22.04.2023 11:40
26.04.2023 23:33
Тип: Архив ZIP - WinRAR
Размер: 3,10 МБ