

BOSCH Case Study for Information Technologies Smart Start Position

Muhammad Riza Fairuzzabadi

ed.fairuzz@gmail.com / fairuzzabadi17@itu.edu.tr

+90 539 354 16 03 / +62 813 2033 4413

Part 1 - Percentage Calculator

Below function/program write a function that calculates how many percent of the day we have completed. Using datetime module, the program checks the current local time, and function check_time calculates how much percent of the day has been completed by that time. Note that datetime.now()'s output normally includes microsecond, but replaced due to redundancy. The output is as seen below.

In [9]:

```
from datetime import datetime, timedelta

def check_time(hh,mm,ss):
    cr_time = timedelta(hours=hh, minutes=mm, seconds=ss)
    full_day = timedelta(hours=24, minutes=0, seconds=0)
    return cr_time/full_day*100

if __name__ == '__main__':
    time = str(datetime.time(datetime.now().replace(microsecond=0)))
    print("Current Time :", end=" ")
    print(time)
    hh, mm, ss = map(int, time.split(':'))
    print("Percentage of the day :", end=" ")
    print(check_time(hh,mm,ss), end=" ")
    print("%")

    #Or Input it manually
    print("Entered Time :", end=" ")
    time2 = "12:00:00"
    print(time2)
    hh, mm, ss = map(int, time2.split(':'))
    print("Percentage of the day :", end=" ")
    print(check_time(hh,mm,ss), end=" ")
    print("%")
```

```
Current Time : 18:05:31
Percentage of the day : 75.38310185185185 %
Entered Time : 12:00:00
Percentage of the day : 50.0 %
```

Part 2 - Web Scraper

Below is a function/program that uses BeautifulSoup4 Python package to parse the web, and append the required data to three arrays of product name, code, and score. However, it does not work correctly most probably due to the security measures by the website. While printing the parsed html and format it using prettify() as in Line 11, the following message is shown alongside the parsed html lines;

Dear User, your request has been rated as suspicious by our firewall and therefore got blocked.
If this is incorrect please get in touch with us via mcw-website-support@bshg.com by referring to 6518716247708424171.

In case the problem is due to other kind of reason, further investigation will be conducted.

In []:

```
import requests
from bs4 import BeautifulSoup

# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    URL = "https://www.bosch-home.com.tr/urun-listesi/buzdolaplari-derin-dondurucular/
buzdolaplari/alttan-donduruculu-buzdolaplari"
    # Above does not work since the website disallows scraping. Error Message -> Dear
    # User, your request has been rated as suspicious by our firewall and therefore go
    # t blocked
    page = requests.get(URL)
    soup = BeautifulSoup(page.content, 'html.parser')
    print(soup.prettify())

    # scrap product name, product code, and score of the refrigerator products
    pr_name = []
    pr_code = []
    pr_score = []

    product_div = soup.find_all('div', class_='item')
    # iterate through every div container we stored in move_div
    for container in product_div:
        pr_series = container.find('div', class_='product-info js-product-info-wrapper')
        .find('div', class_='m-producttitle').a.h2.find('span', class_='fragment normal std-
header-1').text
        pr_tur = container.find('div', class_='product-info js-product-info-wrapper').
find('div', class_='m-producttitle').a.h2.find('span', class_='fragment normal std-hea
der-2').text
        pr_boy = container.find('div', class_='product-info js-product-info-wrapper').
find('div', class_='m-producttitle').a.h2.find('span', class_='fragment normal std-hea
der-3').text
        pr_tip = container.find('div', class_='product-info js-product-info-wrapper').
find('div', class_='m-producttitle').a.h2.find('span', class_='fragment normal std-hea
der-4').text
        product = pr_series + pr_tur + pr_boy + pr_tip
        pr_name.append(product)

        # code
        the_code = container.find('div', class_='product-info js-product-info-wrapper')
        .find('div', class_='m-producttitle').h2.span.text
        pr_code.append(the_code)

        # score
        score_loc = container.find('div', class_='product-conversion').find('div', cla
ss_='m-productconversionarea').find('div', class_='js-conversion-wrapper').find('div',
class_='a-rating rating').find('span', class_='text number').text
        pr_score.append(score_loc)

    print(pr_name)
    print(pr_code)
    print(pr_score)
```

Part 3 - Malfunctioning Keyboard

Below is function/program that is supposed to understand which sentence between the 2 sentences given is more likely to be the sentence that the machine outputs incorrectly, looking at the conditions where the keys adjacent to the pressed keys are pressed as well. For that functionality, I declared two int score variables for each function. Then a function iterates through the string char by char, and check which key it belongs to, for each of key letters, I assigned different adjacent_keys dictionary data types that includes every keys adjacent to the key which then compared to the next character of the string. If it is the same, the score of the sentence is incremented by one. At last, the sentence with the highest score is the one that is more likely to be incorrect. The output is as seen below.

In [5]:

```
def malfunctioning_keyboard(data):
    total = 0
    adjacent_keys = {}
    for element in range(0, len(data) - 1):
        if data[element] == 'Q' or data[element] == 'q':
            adjacent_keys = {1: 1, 2: 1, 'w': 1, 'a': 1}
        if data[element] == 'W' or data[element] == 'w':
            adjacent_keys = {'q': 1, 's': 1, 'e': 1, 2: 1}
        if data[element] == 'E' or data[element] == 'e':
            adjacent_keys = {'w': 1, 'd': 1, 'r': 1, 3: 1}
        if data[element] == 'R' or data[element] == 'r':
            adjacent_keys = {'e': 1, 'f': 1, 't': 1, 4: 1}
        if data[element] == 'T' or data[element] == 't':
            adjacent_keys = {'r': 1, 'g': 1, 'y': 1, 5: 1}
        if data[element] == 'Y' or data[element] == 'y':
            adjacent_keys = {'t': 1, 'h': 1, 'u': 1, 6: 1}
        if data[element] == 'U' or data[element] == 'u':
            adjacent_keys = {'y': 1, 'j': 1, 'i': 1, 7: 1}
        if data[element] == 'I' or data[element] == 'i':
            adjacent_keys = {'u': 1, 'k': 1, 'o': 1, 8: 1}
        if data[element] == 'O' or data[element] == 'o':
            adjacent_keys = {'i': 1, 'l': 1, 'p': 1, 9: 1}
        if data[element] == 'P' or data[element] == 'p':
            adjacent_keys = {'o': 1, '"': 1, '[': 1, 0: 1}

        if data[element] == 'A' or data[element] == 'a':
            adjacent_keys = {'q': 1, 's': 1, 'z': 1}
        if data[element] == 'S' or data[element] == 's':
            adjacent_keys = {'a': 1, 'w': 1, 'd': 1, 'x': 1}
        if data[element] == 'D' or data[element] == 'd':
            adjacent_keys = {'s': 1, 'e': 1, 'c': 1, 'f': 1}
        if data[element] == 'F' or data[element] == 'f':
            adjacent_keys = {'d': 1, 'r': 1, 'g': 1, 'v': 1}
        if data[element] == 'G' or data[element] == 'g':
            adjacent_keys = {'f': 1, 't': 1, 'h': 1, 'b': 1}
        if data[element] == 'H' or data[element] == 'h':
            adjacent_keys = {'g': 1, 'y': 1, 'j': 1, 'n': 1}
        if data[element] == 'J' or data[element] == 'j':
            adjacent_keys = {'h': 1, 'u': 1, 'k': 1, 'm': 1}
        if data[element] == 'K' or data[element] == 'k':
            adjacent_keys = {'j': 1, 'i': 1, 'l': 1, ',': 1}
        if data[element] == 'L' or data[element] == 'l':
            adjacent_keys = {'k': 1, 'o': 1, ';': 1, '.': 1}

        if data[element] == 'Z' or data[element] == 'z':
            adjacent_keys = {'a': 1, 'x': 1}
        if data[element] == 'X' or data[element] == 'x':
            adjacent_keys = {'z': 1, 's': 1, 'c': 1}
        if data[element] == 'C' or data[element] == 'c':
            adjacent_keys = {'x': 1, 'd': 1, 'v': 1}
        if data[element] == 'V' or data[element] == 'v':
            adjacent_keys = {'c': 1, 'f': 1, 'b': 1}
        if data[element] == 'B' or data[element] == 'b':
            adjacent_keys = {'v': 1, 'g': 1, 'n': 1}
        if data[element] == 'N' or data[element] == 'n':
            adjacent_keys = {'b': 1, 'h': 1, 'm': 1}
        if data[element] == 'M' or data[element] == 'm':
            adjacent_keys = {'n': 1, 'j': 1, 'k': 1, ',': 1}
        total = total + adjacent_keys.get(data[element + 1], 0)
    return total

if __name__ == '__main__':
    Sentence_One = 'Qwuality isd mucxh bettedr than quanmtity. Oine homke run is much
bvetter than two doubnles'
    Sentence_Two = 'Quality is much better than quantity. One home run is much better
than two doubles'
    score_a = malfunctioning_keyboard(Sentence_One)
```

```

score_b = malfunctioning_keyboard(Sentence_Two)
print(score_a)
print(score_b)
if score_a > score_b:
    print("The first sentence is more likely to be an incorrect sentence")
if score_a < score_b:
    print("The second sentence is more likely to be an incorrect sentence")
if score_a == score_b:
    print("Both sentences have the same probability to be incorrect")

```

15

7

The first sentence is more likely to be an incorrect sentence

Part 4 - Crypto Currency Calculator

In Part 4, a simple Crypto Currency trading program is implemented. Being a GUI program, C# with Winforms (.NET Framework 5.0) was preferred. The functionalities of the program are

1. Follow live exchange rates from USD to five Cryptocurrencies, and other way around.
2. Buy, sell and calculate the transaction of each currencies.
3. A Wallet functionality. Showing how much money left in the currency.
4. Since it is a Crypto Calculator, the amount of Dollar left or gain after transaction is not calculated. For the same reason the dollar amount won't change during purchase for convenience.
5. ExcelMapper Nuget Package is used to Read data from .xlsx file. The program's screenshot and codes program.cs, form1.cs, and form1.Designer.cs are given below. Note that since it is not written in Python, the codes are not run (no output). All files are attached to the email.

□

Program.cs

In []:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace CryptoCurrencyCalculator_Bosch4
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.SetHighDpiMode(HighDpiMode.SystemAware);
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new CryptoCalculator());
        }
    }
}

```

Form1.cs

In []:

```

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.Runtime.InteropServices;
using System.Windows.Forms;
using Ganss.Excel;
```

```
namespace CryptocurrencyCalculator_Bosch4
```

```
{

    public partial class CryptoCalculator : Form
    {
        private double DollarWal;
        private double BTCWal;
        private double BCHWal;
        private double LTCWal;
        private double XRPWal;
        private double ETHWal;
        private double SecondWal;

        private double BTCcoeff;
        private double BCHcoeff;
        private double LTCcoeff;
        private double XRPcoeff;
        private double ETHcoeff;

        private double YourBTC;
        private double YourBCH;
        private double YourLTC;
        private double YourXRP;
        private double YourETH;

        private List<double> arrayer = new List<double>();

        public CryptoCalculator()
        {
            InitializeComponent();
        }

        public record Crypto(
            string name,
            double Price
        );

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void Opener_Click(object sender, EventArgs e)
        {
            string file = @"D:\Crypto.xlsx";
            var crypto = new ExcelMapper(file).Fetch<Crypto>();
            foreach (var st in crypto)
            {
                arrayer.Add(st.Price);
            }
            BTCcoeff = arrayer[0];
            BCHcoeff = arrayer[1];
            LTCcoeff = arrayer[2];
            XRPcoeff = arrayer[3];
            ETHcoeff = arrayer[4];
        }

        private void CalculateButton_Click(object sender, EventArgs e)
```

```

    {
        DollarWal = double.Parse(DollarValue.Text);
        BTCWal = DollarWal * 1 / BTCcoeff;
        BCHWal = DollarWal * 1 / BCHcoeff;
        LTCWal = DollarWal * 1 / LTCcoeff;
        XRPWal = DollarWal * 1 / XRPcoeff;
        ETHWal = DollarWal * 1 / ETHcoeff;

        btcval.Text = BTCWal.ToString();
        bchval.Text = BCHWal.ToString();
        ltcval.Text = LTCWal.ToString();
        xrpval.Text = XRPWal.ToString();
        ethval.Text = ETHWal.ToString();
    }

private void BuyBTC_Click(object sender, EventArgs e)
{
    if(YourBTC == 0)
    {
        YourBTC = BTCWal;
    }
    else
    {
        YourBTC = YourBTC + BTCWal;
    }
    btcval2.Text = YourBTC.ToString();
}

private void BuyBCH_Click(object sender, EventArgs e)
{
    if (YourBCH == 0)
    {
        YourBCH = BCHWal;
    }
    else
    {
        YourBCH = YourBCH + BTCWal;
    }
    bchval2.Text = YourBCH.ToString();
}

private void BuyLTC_Click(object sender, EventArgs e)
{
    if (YourLTC == 0)
    {
        YourLTC = LTCWal;
    }
    else
    {
        YourLTC = YourLTC + LTCWal;
    }
    ltcval2.Text = YourLTC.ToString();
}

private void BuyXRP_Click(object sender, EventArgs e)
{
    if (YourXRP == 0)
    {
        YourXRP = XRPWal;
    }
    else
    {
        YourXRP = YourXRP + XRPWal;
    }
    xrpval2.Text = YourXRP.ToString();
}

private void BuyETH_Click(object sender, EventArgs e)
{
    if (YourETH == 0)

```

```

        {
            YourETH = ETHWal;
        }
        else
        {
            YourETH = YourETH + ETHWal;
        }
        ethval2.Text = YourETH.ToString();
    }

private void CalculateBTC_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    BTCWal = SecondWal * BTCcoeff;
    DollarValue.Text = BTCWal.ToString();
}

private void CalculateBCH_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    BCHWal = SecondWal * BCHcoeff;
    DollarValue.Text = BCHWal.ToString();
}

private void CalculateLTC_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    LTCWal = SecondWal * LTCcoeff;
    DollarValue.Text = LTCWal.ToString();
}

private void CalculateXRP_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    XRPWal = SecondWal * XRPcoeff;
    DollarValue.Text = XRPWal.ToString();
}

private void CalculateETH_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    ETHWal = SecondWal * ETHcoeff;
    DollarValue.Text = ETHWal.ToString();
}

private void sellbtc_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    BTCWal = SecondWal * BTCcoeff;
    DollarValue.Text = BTCWal.ToString();
    YourBTC -= SecondWal;
    btcval2.Text = YourBTC.ToString();
}

private void Sellbch_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    BCHWal = SecondWal * BCHcoeff;
    DollarValue.Text = BCHWal.ToString();
    YourBCH -= SecondWal;
    bchval2.Text = YourBCH.ToString();
}

private void sellltp_Click(object sender, EventArgs e)
{
    SecondWal = double.Parse(Inputer.Text);
    LTCWal = SecondWal * LTCcoeff;
    DollarValue.Text = LTCWal.ToString();
    YourLTC -= SecondWal;
    ltcval2.Text = YourLTC.ToString();
}

```

```

    }

    private void sellXRP_Click(object sender, EventArgs e)
    {
        SecondWal = double.Parse(Inputer.Text);
        XRPWal = SecondWal * XRPcoeff;
        DollarValue.Text = XRPWal.ToString();
        YourXRP -= SecondWal;
        xrpval2.Text = YourXRP.ToString();
    }

    private void SellETH_Click(object sender, EventArgs e)
    {
        SecondWal = double.Parse(Inputer.Text);
        ETHWal = SecondWal * ETHcoeff;
        YourXRP -= SecondWal;
        ethval2.Text = YourETH.ToString();
        DollarValue.Text = ETHWal.ToString();
    }
}
}

```

Form1D.Designer.cs

In []:

```

namespace CryptoCurrencyCalculator_Bosch4
{
    partial class CryptoCalculator
    {
        /// <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.Dollar = new System.Windows.Forms.Label();
            this.CalculateButton = new System.Windows.Forms.Button();
            this.DollarValue = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.BuyBTC = new System.Windows.Forms.Button();
            this.button3 = new System.Windows.Forms.Button();

```



```

this.BuyBCH = new System.Windows.Forms.Button();
this.BuyLTC = new System.Windows.Forms.Button();
this.BuyXRP = new System.Windows.Forms.Button();
this.BuyETH = new System.Windows.Forms.Button();
this.bchval = new System.Windows.Forms.TextBox();
this.btcval = new System.Windows.Forms.TextBox();
this.ltcval = new System.Windows.Forms.TextBox();
this.xrpval = new System.Windows.Forms.TextBox();
this.ethval = new System.Windows.Forms.TextBox();
this.bchval2 = new System.Windows.Forms.TextBox();
this.btcval2 = new System.Windows.Forms.TextBox();
this.ltcval2 = new System.Windows.Forms.TextBox();
this.xrpval2 = new System.Windows.Forms.TextBox();
this.sellbtc = new System.Windows.Forms.Button();
this.sellbch = new System.Windows.Forms.Button();
this.sellltc = new System.Windows.Forms.Button();
this.sellXRP = new System.Windows.Forms.Button();
this.sellETH = new System.Windows.Forms.Button();
this.Inputer = new System.Windows.Forms.TextBox();
this.Opener = new System.Windows.Forms.Button();
this.ethval2 = new System.Windows.Forms.TextBox();
this.label8 = new System.Windows.Forms.Label();
this.CalculateBTC = new System.Windows.Forms.Button();
this.CalculateBCH = new System.Windows.Forms.Button();
this.CalculateLTC = new System.Windows.Forms.Button();
this.CalculateXRP = new System.Windows.Forms.Button();
this.CalculateETH = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// Dollar
//
this.Dollar.AutoSize = true;
this.Dollar.ForeColor = System.Drawing.SystemColors.Control;
this.Dollar.Location = new System.Drawing.Point(12, 60);
this.Dollar.Name = "Dollar";
this.Dollar.Size = new System.Drawing.Size(38, 15);
this.Dollar.TabIndex = 1;
this.Dollar.Text = "Dollar";
//
// CalculateButton
//
this.CalculateButton.Location = new System.Drawing.Point(272, 87);
this.CalculateButton.Name = "CalculateButton";
this.CalculateButton.Size = new System.Drawing.Size(84, 31);
this.CalculateButton.TabIndex = 2;
this.CalculateButton.Text = "Calculate";
this.CalculateButton.UseVisualStyleBackColor = true;
this.CalculateButton.Click += new System.EventHandler(this.CalculateButto
n_Click);
//
// DollarValue
//
this.DollarValue.Location = new System.Drawing.Point(71, 58);
this.DollarValue.Name = "DollarValue";
this.DollarValue.Size = new System.Drawing.Size(285, 23);
this.DollarValue.TabIndex = 3;
//
// label1
//
this.label1.AutoSize = true;
this.label1.ForeColor = System.Drawing.SystemColors.ButtonFace;
this.label1.Location = new System.Drawing.Point(12, 9);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(148, 15);
this.label1.TabIndex = 0;
this.label1.Text = "Cryptocurrency Calculator";
//
// label2
//
this.label2.AutoSize = true;

```

```

this.label2.ForeColor = System.Drawing.SystemColors.Control;
this.label2.Location = new System.Drawing.Point(12, 181);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(26, 15);
this.label2.TabIndex = 4;
this.label2.Text = "BTC";
//
// label3
//
this.label3.AutoSize = true;
this.label3.ForeColor = System.Drawing.SystemColors.Control;
this.label3.Location = new System.Drawing.Point(12, 210);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(31, 15);
this.label3.TabIndex = 5;
this.label3.Text = "BCH";
//
// label4
//
this.label4.AutoSize = true;
this.label4.ForeColor = System.Drawing.SystemColors.Control;
this.label4.Location = new System.Drawing.Point(12, 242);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(25, 15);
this.label4.TabIndex = 6;
this.label4.Text = "LTC";
//
// label5
//
this.label5.AutoSize = true;
this.label5.ForeColor = System.Drawing.SystemColors.Control;
this.label5.Location = new System.Drawing.Point(12, 274);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(28, 15);
this.label5.TabIndex = 7;
this.label5.Text = "XRP";
//
// label6
//
this.label6.AutoSize = true;
this.label6.ForeColor = System.Drawing.SystemColors.Control;
this.label6.Location = new System.Drawing.Point(12, 298);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(28, 15);
this.label6.TabIndex = 8;
this.label6.Text = "ETH";
//
// label7
//
this.label7.AutoSize = true;
this.label7.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.label7.Location = new System.Drawing.Point(361, 149);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(67, 15);
this.label7.TabIndex = 9;
this.label7.Text = "Your Wallet";
//
// BuyBTC
//
this.BuyBTC.Location = new System.Drawing.Point(47, 340);
this.BuyBTC.Name = "BuyBTC";
this.BuyBTC.Size = new System.Drawing.Size(72, 23);
this.BuyBTC.TabIndex = 10;
this.BuyBTC.Text = "Buy BTC";
this.BuyBTC.UseVisualStyleBackColor = true;
this.BuyBTC.Click += new System.EventHandler(this.BuyBTC_Click);
//
// button3
//
this.button3.Location = new System.Drawing.Point(328, 360);

```

```
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(8, 8);
this.button3.TabIndex = 12;
this.button3.Text = "button3";
this.button3.UseVisualStyleBackColor = true;
//
// BuyBCH
//
this.BuyBCH.Location = new System.Drawing.Point(125, 340);
this.BuyBCH.Name = "BuyBCH";
this.BuyBCH.Size = new System.Drawing.Size(75, 23);
this.BuyBCH.TabIndex = 11;
this.BuyBCH.Text = "Buy BCH";
this.BuyBCH.UseVisualStyleBackColor = true;
this.BuyBCH.Click += new System.EventHandler(this.BuyBCH_Click);
//
// BuyLTC
//
this.BuyLTC.Location = new System.Drawing.Point(206, 340);
this.BuyLTC.Name = "BuyLTC";
this.BuyLTC.Size = new System.Drawing.Size(75, 23);
this.BuyLTC.TabIndex = 12;
this.BuyLTC.Text = "Buy LTC";
this.BuyLTC.UseVisualStyleBackColor = true;
this.BuyLTC.Click += new System.EventHandler(this.BuyLTC_Click);
//
// BuyXRP
//
this.BuyXRP.Location = new System.Drawing.Point(287, 340);
this.BuyXRP.Name = "BuyXRP";
this.BuyXRP.Size = new System.Drawing.Size(75, 23);
this.BuyXRP.TabIndex = 13;
this.BuyXRP.Text = "Buy XRP";
this.BuyXRP.UseVisualStyleBackColor = true;
this.BuyXRP.Click += new System.EventHandler(this.BuyXRP_Click);
//
// BuyETH
//
this.BuyETH.Location = new System.Drawing.Point(368, 340);
this.BuyETH.Name = "BuyETH";
this.BuyETH.Size = new System.Drawing.Size(75, 23);
this.BuyETH.TabIndex = 14;
this.BuyETH.Text = "BuyETH";
this.BuyETH.UseVisualStyleBackColor = true;
this.BuyETH.Click += new System.EventHandler(this.BuyETH_Click);
//
// bchval
//
this.bchval.Location = new System.Drawing.Point(71, 207);
this.bchval.Name = "bchval";
this.bchval.Size = new System.Drawing.Size(184, 23);
this.bchval.TabIndex = 16;
//
// btcval
//
this.btcval.Location = new System.Drawing.Point(71, 178);
this.btcval.Name = "btcval";
this.btcval.Size = new System.Drawing.Size(184, 23);
this.btcval.TabIndex = 17;
//
// ltcval
//
this.ltcval.Location = new System.Drawing.Point(71, 236);
this.ltcval.Name = "ltcval";
this.ltcval.Size = new System.Drawing.Size(184, 23);
this.ltcval.TabIndex = 18;
//
// xrpval
//
this.xrpval.Location = new System.Drawing.Point(71, 266);
```

```

this.xrpval.Name = "xrpval";
this.xrpval.Size = new System.Drawing.Size(184, 23);
this.xrpval.TabIndex = 19;
//
// ethval
//
this.ethval.Location = new System.Drawing.Point(71, 295);
this.ethval.Name = "ethval";
this.ethval.Size = new System.Drawing.Size(184, 23);
this.ethval.TabIndex = 20;
//
// bchval2
//
this.bchval2.Location = new System.Drawing.Point(261, 207);
this.bchval2.Name = "bchval2";
this.bchval2.Size = new System.Drawing.Size(182, 23);
this.bchval2.TabIndex = 22;
//
// btcval2
//
this.btcval2.Location = new System.Drawing.Point(261, 178);
this.btcval2.Name = "btcval2";
this.btcval2.Size = new System.Drawing.Size(182, 23);
this.btcval2.TabIndex = 23;
//
// ltcval2
//
this.ltcval2.Location = new System.Drawing.Point(261, 236);
this.ltcval2.Name = "ltcval2";
this.ltcval2.Size = new System.Drawing.Size(182, 23);
this.ltcval2.TabIndex = 24;
//
// xrpval2
//
this.xrpval2.Location = new System.Drawing.Point(261, 266);
this.xrpval2.Name = "xrpval2";
this.xrpval2.Size = new System.Drawing.Size(182, 23);
this.xrpval2.TabIndex = 25;
//
// sellbtc
//
this.sellbtc.Location = new System.Drawing.Point(462, 178);
this.sellbtc.Name = "sellbtc";
this.sellbtc.Size = new System.Drawing.Size(75, 23);
this.sellbtc.TabIndex = 27;
this.sellbtc.Text = "Sell";
this.sellbtc.UseVisualStyleBackColor = true;
this.sellbtc.Click += new System.EventHandler(this.sellbtc_Click);
//
// Sellbch
//
this.Sellbch.Location = new System.Drawing.Point(462, 207);
this.Sellbch.Name = "Sellbch";
this.Sellbch.Size = new System.Drawing.Size(75, 23);
this.Sellbch.TabIndex = 28;
this.Sellbch.Text = "Sell";
this.Sellbch.UseVisualStyleBackColor = true;
this.Sellbch.Click += new System.EventHandler(this.Sellbch_Click);
//
// sellltc
//
this.sellltc.Location = new System.Drawing.Point(462, 236);
this.sellltc.Name = "sellltc";
this.sellltc.Size = new System.Drawing.Size(75, 23);
this.sellltc.TabIndex = 29;
this.sellltc.Text = "Sell";
this.sellltc.UseVisualStyleBackColor = true;
this.sellltc.Click += new System.EventHandler(this.sellltc_Click);
//
// sellXRP

```

```

//
this.sellXRP.Location = new System.Drawing.Point(462, 266);
this.sellXRP.Name = "sellXRP";
this.sellXRP.Size = new System.Drawing.Size(75, 23);
this.sellXRP.TabIndex = 30;
this.sellXRP.Text = "Sell";
this.sellXRP.UseVisualStyleBackColor = true;
this.sellXRP.Click += new System.EventHandler(this.sellXRP_Click);
//
// SellETH
//
this.SellETH.Location = new System.Drawing.Point(462, 295);
this.SellETH.Name = "SellETH";
this.SellETH.Size = new System.Drawing.Size(75, 23);
this.SellETH.TabIndex = 31;
this.SellETH.Text = "Sell";
this.SellETH.UseVisualStyleBackColor = true;
this.SellETH.Click += new System.EventHandler(this.SellETH_Click);
//
// Inputer
//
this.Inputer.Location = new System.Drawing.Point(451, 146);
this.Inputer.Name = "Inputer";
this.Inputer.Size = new System.Drawing.Size(177, 23);
this.Inputer.TabIndex = 32;
//
// Opener
//
this.Opener.Location = new System.Drawing.Point(368, 58);
this.Opener.Name = "Opener";
this.Opener.Size = new System.Drawing.Size(75, 23);
this.Opener.TabIndex = 33;
this.Opener.Text = "Open xlsx";
this.Opener.UseVisualStyleBackColor = true;
this.Opener.Click += new System.EventHandler(this.Opener_Click);
//
// ethval2
//
this.ethval2.Location = new System.Drawing.Point(261, 295);
this.ethval2.Name = "ethval2";
this.ethval2.Size = new System.Drawing.Size(182, 23);
this.ethval2.TabIndex = 34;
//
// label8
//
this.label8.AutoSize = true;
this.label8.ForeColor = System.Drawing.SystemColors.ButtonHighlight;
this.label8.Location = new System.Drawing.Point(485, 128);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(108, 15);
this.label8.TabIndex = 35;
this.label8.Text = "The Amount to Sell";
//
// CalculateBTC
//
this.CalculateBTC.Location = new System.Drawing.Point(543, 178);
this.CalculateBTC.Name = "CalculateBTC";
this.CalculateBTC.Size = new System.Drawing.Size(75, 23);
this.CalculateBTC.TabIndex = 36;
this.CalculateBTC.Text = "Calculate";
this.CalculateBTC.UseVisualStyleBackColor = true;
this.CalculateBTC.Click += new System.EventHandler(this.CalculateBTC_Click);
//
// CalculateBCH
//
this.CalculateBCH.Location = new System.Drawing.Point(543, 207);
this.CalculateBCH.Name = "CalculateBCH";
this.CalculateBCH.Size = new System.Drawing.Size(75, 23);
this.CalculateBCH.TabIndex = 37;

```

k);

```

this.CalculateBCH.Text = "Calculate";
this.CalculateBCH.UseVisualStyleBackColor = true;
this.CalculateBCH.Click += new System.EventHandler(this.CalculateBCH_Click);

//
// CalculateLTC
//
this.CalculateLTC.Location = new System.Drawing.Point(543, 236);
this.CalculateLTC.Name = "CalculateLTC";
this.CalculateLTC.Size = new System.Drawing.Size(75, 23);
this.CalculateLTC.TabIndex = 38;
this.CalculateLTC.Text = "Calculate";
this.CalculateLTC.UseVisualStyleBackColor = true;
this.CalculateLTC.Click += new System.EventHandler(this.CalculateLTC_Click);

//
// CalculateXRP
//
this.CalculateXRP.Location = new System.Drawing.Point(543, 265);
this.CalculateXRP.Name = "CalculateXRP";
this.CalculateXRP.Size = new System.Drawing.Size(75, 23);
this.CalculateXRP.TabIndex = 39;
this.CalculateXRP.Text = "Calculate";
this.CalculateXRP.UseVisualStyleBackColor = true;
this.CalculateXRP.Click += new System.EventHandler(this.CalculateXRP_Click);

//
// CalculateETH
//
this.CalculateETH.Location = new System.Drawing.Point(543, 294);
this.CalculateETH.Name = "CalculateETH";
this.CalculateETH.Size = new System.Drawing.Size(75, 23);
this.CalculateETH.TabIndex = 40;
this.CalculateETH.Text = "Calculate";
this.CalculateETH.UseVisualStyleBackColor = true;
this.CalculateETH.Click += new System.EventHandler(this.CalculateETH_Click);

//
// CryptoCalculator
//
this.AutoScaleDimensions = new System.Drawing.SizeF(7F, 15F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.Color.DarkSlateGray;
this.ClientSize = new System.Drawing.Size(658, 400);
this.Controls.Add(this.CalculateETH);
this.Controls.Add(this.CalculateXRP);
this.Controls.Add(this.CalculateLTC);
this.Controls.Add(this.CalculateBCH);
this.Controls.Add(this.CalculateBTC);
this.Controls.Add(this.label8);
this.Controls.Add(this.ethval2);
this.Controls.Add(this.Opener);
this.Controls.Add(this.Inputer);
this.Controls.Add(this.SellETH);
this.Controls.Add(this.sellXRP);
this.Controls.Add(this.sellltc);
this.Controls.Add(this.Sellbch);
this.Controls.Add(this.sellbtc);
this.Controls.Add(this.xrpval2);
this.Controls.Add(this.ltcval2);
this.Controls.Add(this.btcval2);
this.Controls.Add(this.bchval2);
this.Controls.Add(this.ethval);
this.Controls.Add(this.xrpval);
this.Controls.Add(this.ltcval);
this.Controls.Add(this.btcval);
this.Controls.Add(this.bchval);
this.Controls.Add(this.BuyETH);
this.Controls.Add(this.BuyXRP);
this.Controls.Add(this.BuyLTC);

```

```

        this.Controls.Add(this.BuyBCH);
        this.Controls.Add(this.BuyBTC);
        this.Controls.Add(this.label7);
        this.Controls.Add(this.label6);
        this.Controls.Add(this.label5);
        this.Controls.Add(this.label4);
        this.Controls.Add(this.label3);
        this.Controls.Add(this.label2);
        this.Controls.Add(this.DollarValue);
        this.Controls.Add(this.CalculateButton);
        this.Controls.Add(this.Dollar);
        this.Controls.Add(this.label1);
        this.Name = "CryptoCalculator";
        this.Text = "Form1";
        this.Load += new System.EventHandler(this.Form1_Load);
        this.ResumeLayout(false);
        this.PerformLayout();

    }

```

#endregion

```

private System.Windows.Forms.Label Dollar;
private System.Windows.Forms.Button CalculateButton;
private System.Windows.Forms.TextBox DollarValue;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.Label label5;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.Button BuyBTC;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Button BuyBCH;
private System.Windows.Forms.Button BuyLTC;
private System.Windows.Forms.Button BuyXRP;
private System.Windows.Forms.Button BuyETH;
private System.Windows.Forms.TextBox bchval;
private System.Windows.Forms.TextBox btcval;
private System.Windows.Forms.TextBox ltcval;
private System.Windows.Forms.TextBox xrpval;
private System.Windows.Forms.TextBox ethval;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.TextBox btcval2;
private System.Windows.Forms.TextBox ltcval2;
private System.Windows.Forms.TextBox xrpval2;
private System.Windows.Forms.Button sellbtc;
private System.Windows.Forms.Button Sellbch;
private System.Windows.Forms.Button sellltc;
private System.Windows.Forms.Button sellXRP;
private System.Windows.Forms.Button SelleTH;
private System.Windows.Forms.TextBox Inputer;
private System.Windows.Forms.Button Opener;
private System.Windows.Forms.TextBox bchval2;
private System.Windows.Forms.TextBox ethval2;
private System.Windows.Forms.TextBox ethvalbox;
private System.Windows.Forms.Label label8;
private System.Windows.Forms.Button CalculateBTC;
private System.Windows.Forms.Button CalculateBCH;
private System.Windows.Forms.Button CalculateLTC;
private System.Windows.Forms.Button CalculateXRP;
private System.Windows.Forms.Button CalculateETH;

```

}

}