

## מטלה GUI-4

מגישות: בתאל טסמה-318648318 תמר יצחק - 316226133

### חלק 1 :

#### CLASS CompareFiles :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WindowsFormsApp1_GUI_2
{
    internal static class CompareFiles
    {
        public static bool equalFiles(DataFile file1, DataFile file2)
        {
            if (file1 != null && file2 != null)
            {
                if (file1.getData().Equals(file2.getData()) &&
file1.getName().Equals(file2.getName()))
                    return true;
                else
                    return false;
            }
            else return false;
        }
        public static int compareSizeFiles(DataFile file1, DataFile file2)
        {
            if (file1.getSize(file1.getData()) > file2.getSize(file2.getData()))
                return 1;
            if (file1.getSize(file1.getData()) < file2.getSize(file2.getData()))
                return -1;
            else
                return 0;
        }
    }
}
```

#### CLASS DataFile:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Linq.Expressions;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```

using System.Xml.Linq;

namespace WindowsFormsApp1_GUI_2
{
    enum FileTypeExntion { TXT = 0, DOC, DOCX, PDF, PPTX }
    internal class DataFile
    {
        string fileName, data;
        FileTypeExntion type;
        static int counter;
        static int unique=1;
        DateTime lastUpdateTime = new DateTime();
        public string getName() { return fileName; }
        public DateTime getTime() { return lastUpdateTime; }
        public string getData() { return data; }

        public FileTypeExntion fileType() { return type; }

        public void setData(string Data)
        {
            this.data = Data;
        }
        public bool setName(string name)
        {
            foreach (char tav in name)
            {
                if (tav == '/' || tav == '\\' || tav == ':' || tav == '?' || tav ==
''' || tav == '<' || tav == '>' || tav == '|')
                {
                    MessageBox.Show("A file name can't contain the following
characters:" + "/ \ : * ? " + "<" + ">" + "|");
                    MessageBox.Show("please enter a valid name");
                    return false;
                }
            }
            fileName = name;
            return true;
        }
        public void setTime()
        {
            lastUpdateTime = DateTime.Now;
        }
        public DataFile(string name, string Data,FileTypeExntion Type)
        {
            setName(name);
            setTime();
            setData(Data);
            this.type = Type;
            counter++;
        }
        public DataFile() : this("sameFile"+unique++, "",FileTypeExntion.PDF)
        {
        }
        public DataFile(DataFile other)
        {
            fileName = other.fileName + "2";
        }
    }
}

```

```

        data = other.data;
        lastUpdateTime = other.lastUpdateTime;
    }
    public int getSize(string Data)
    {
        int count = Data.Length;
        return count;
    }
    public string dir(string name, string data, DateTime time)
    {
        string str;
        int sizeinKB = getSize(data) / 1024;
        str = time + " " + sizeinKB + "KB " + name+"\n";
        return str;
    }
}
}

```

### CLASS QueueFiles:

```

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

namespace WindowsFormsApp1_GUI_2
{
    internal class QueueFiles
    {
        DataFile[] filesArray;
        int freeSpace = 1;

        public QueueFiles()
        {
            filesArray=new DataFile[0];
            freeSpace = -1;
        }

        public bool isEmpty ()
        {
            if (freeSpace <=0)
            {
                return true;
            }
            else return false;
        }

        public void enqueue (DataFile file1)
        {
            if(freeSpace == -1) { freeSpace = 0; }
            foreach (DataFile file in filesArray)
            {

```

```

        if (CompareFiles.equalFiles(file, file1))
        {
            return;
        }
    }
    Array.Resize(ref filesArray, filesArray.Length+1);
    filesArray[freeSpace++] = file1;
}
public DataFile dequeue ()
{
    if (isEmpty()==false)
    {
        DataFile tempfile= filesArray[0];
        if(freeSpace==0)
            return tempfile;
        for (int i =0; i < freeSpace-1; i++)
        {
            filesArray[i] = filesArray[i + 1];
        }
        freeSpace--;
        filesArray[freeSpace] = null;
        Array.Resize(ref filesArray, filesArray.Length -1);
        return tempfile;
    }
    else return null;
}

public DataFile bigFile()
{
    if (freeSpace <= 0)
    {
        MessageBox.Show("There are no files in the system!");
        return null;
    }
    if (freeSpace == 1)
    {
        MessageBox.Show("There is only one file !");
        return null;
    }
    else
    {
        QueueFiles fileLineUp = new QueueFiles();
        DataFile firstfile=dequeue();
        DataFile secondfile = dequeue();
        int result=CompareFiles.compareSizeFiles(firstfile, secondfile);
        DataFile largestFile = result==1?firstfile:secondfile;
        fileLineUp.enqueue(firstfile);
        fileLineUp.enqueue(secondfile);

        while (!isEmpty())
        {
            DataFile anotherFile = dequeue();
            result = CompareFiles.compareSizeFiles(largestFile,
anotherFile);
            if (result == -1)
            {
                fileLineUp.enqueue(largestFile);

```

```

        largestFile = anotherFile;
    }
    else
    {
        fileLineUp.enqueue(anotherFile);
    }
}
while (!fileLineUp.isEmpty())
{
    enqueue(fileLineUp.dequeue());
}
if (result == 0)
{
    MessageBox.Show("The files are equal!");
    return null;
}
else return largestFile;
}
}
public string printQueue()
{
    string str=null;
    if (isEmpty()==true)
    {
        return null;
    }
    DataFile file;
    QueueFiles queueFilearr = new QueueFiles();
    while (isEmpty()==false)
    {
        file=dequeue();//כאן שמור הקובץ שהוצא מהמערך
        str+=file.dir(file.getName(),file.getData(),DateTime.Now);// הדפסת
פרטי הקובץ היוצא
        queueFilearr.enqueue(file);//הוספת הקובץ היוצא בחזרה למערך הזמני
    }
    while(queueFilearr.isEmpty()==false)
    {
        enqueue(queueFilearr.dequeue());//הוספת הקובץ המוצא בחזרה למערך המקורי
    }
    return str;
}
public QueueFiles searchByFileType(FileTypeExtention Type)
{
    QueueFiles sameTypeArr=new QueueFiles();
    QueueFiles tempArr = new QueueFiles();
    DataFile randfile;
    while (isEmpty() == false)
    {
        randfile = dequeue();//כאן שמור הקובץ שהוצא מהמערך
        tempArr.enqueue(randfile);//הוספת הקובץ היוצא בחזרה למערך הזמני
        if (randfile.fileType()==Type) //הדפסת פרטי הקובץ היוצא
            sameTypeArr.enqueue(randfile);
    }
    while (tempArr.isEmpty() == false)
    {
        enqueue(tempArr.dequeue());//הוספת הקובץ המוצא בחזרה למערך המקורי
    }
    if (sameTypeArr.isEmpty() == true)

```

```

        MessageBox.Show("There are no files of this type!");
    }
    return sameTypeArr;
}
}
}

```

## **CLASS PROGRAM:**

```

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

namespace WindowsFormsApp1_GUI_2
{
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

## **FORM1:**

```

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 WindowsFormsApp1_GUI_2
{
    public partial class Form1 : Form
    {
        QueueFiles FilesArray = new QueueFiles();
        public Form1()
        {

```

```

        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        MessageBox.Show("There are no files in the system");
        lblnewfilename.Visible = false;
        lblfiledata.Visible = false;
        lblnewfiletype.Visible = false;
        btnNewfileadded.Visible = false;
        txtNewfilename.Visible = false;
        txtNewfiledata.Visible = false;
        cmbNewfileenum.Visible = false;
    }

    private void btnSelectfileType_Click(object sender, EventArgs e)
    {
        txtFilesbytype.Clear();
        FileTypeExtention sometype= (FileTypeExtention)cmbEnum.SelectedIndex;
        QueueFiles res;
        res=FilesArray.searchByFileType(sometype);
        string re = res.printQueue();
        if (res == null)
            MessageBox.Show("Not enough files in the queue");
        txtFilesbytype.Visible = true;
        txtFilesbytype.Text =re;
        cmbEnum.SelectedIndex = -1;
    }

    private void btnDefaultfile_Click(object sender, EventArgs e)
    {
        DataFile Defaultfile= new DataFile();
        FilesArray.enqueue(Defaultfile);
        MessageBox.Show("Default file added!");
    }

    private void btnNewFile_Click(object sender, EventArgs e)
    {
        lblnewfilename.Visible = true;
        lblfiledata.Visible = true;
        lblnewfiletype.Visible = true;
        btnNewfileadded.Visible = true;
        txtNewfilename.Visible = true;
        txtNewfiledata.Visible = true;
        cmbNewfileenum.Visible = true;
    }

    private void btnNewfileadded_Click(object sender, EventArgs e)
    {
        string filename = null;
        string filedata=null;
    }

```

```

        FileTypeExtention newtype =
(FileTypeExtention)cmbNewfileenum.SelectedIndex;
        filename=txtNewfilename.Text;
        filedata=txtNewfiledata.Text;
        if (txtNewfilename.Text == "" || txtNewfiledata.Text == "" ||
cmbNewfileenum.SelectedIndex == -1)
        {
            MessageBox.Show("You must enter all the details first!");
            cmbNewfileenum.SelectedIndex = -1;
            txtNewfiledata.Text = "";
            txtNewfilename.Text = "";
        }
        else
        {
            DataFile newFile = new DataFile(filename, filedata, newtype);
            if (newFile.getName() == null)
            {
                txtNewfilename.Text = "";
            }

            else
            {
                FilesArray.enqueue(newFile);
                MessageBox.Show("New File added!");
                cmbNewfileenum.SelectedIndex = -1;
                txtNewfiledata.Text = "";
                txtNewfilename.Text = "";
            }
        }
    }

private void btnRemovefile_Click(object sender, EventArgs e)
{
    FilesArray.dequeue();
    MessageBox.Show("Oldest file removed successfully!");
    txtFileslist.Text = "";
    txtFilesbytype.Text = "";
    txtLargestfile.Text = "";
}

private void btnPrintall_Click(object sender, EventArgs e)
{
    txtFileslist.Text=FilesArray.printQueue();
    if (txtFileslist.Text == "")
        MessageBox.Show("There are no files in the system!");
}

private void btnLargestfile_Click(object sender, EventArgs e)
{
    DataFile largestfile;
    largestfile=FilesArray.bigFile();
    if (largestfile != null)
        txtLargestfile.Text = largestfile.dir(largestfile.getName(),
largestfile.getData(), DateTime.Now);
}

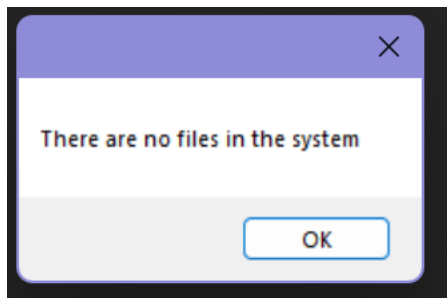
```



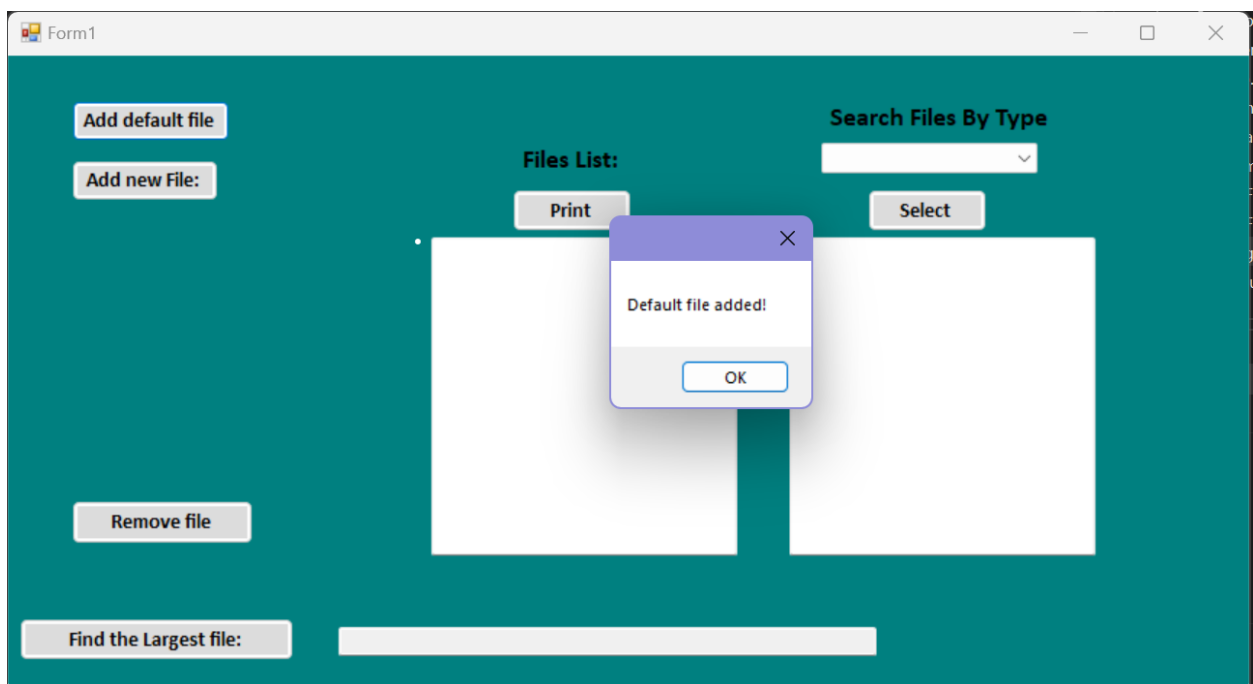
```
}  
  
}  
}
```

### קבצי פלט:

בטעינת הקובץ מופיעה ההודעה הבאה :



בלחיצה על כפתור ADD DEFAULT FILE :



בלחיצה על כפתור ADD NEW FILE :

## יפתחו האופציות הבאות

The screenshot shows a Windows application window titled "Form1" with a teal background. On the left, there are several controls: a button "Add default file", a button "Add new File:", a text input "File Name:" followed by a white box, a text input "File Data:" followed by a white box, a dropdown menu "File Type:" with a downward arrow, a button "Add", and a button "Remove file". At the bottom left is a button "Find the Largest file:" followed by a long white progress bar. In the center, under the heading "Files List:", there is a button "Print" and a large empty white rectangular area. On the right, under the heading "Search Files By Type", there is a dropdown menu with a downward arrow and a button "Select" above another large empty white rectangular area.

דוגמה להכנסת שם קובץ שגוי :

This screenshot shows the same application window as before, but with an error dialog box overlaid in the center. The dialog box has a purple title bar with a close button (X) and contains the text: "A file name can't contain the following characters: / \ : \* ? < > |". There is an "OK" button at the bottom right of the dialog. In the background, the "File Name:" input field now contains the text "c?".

לאחר הופעת ההודעה, תא השם מתרוקן אוטומטית וניתן להכניס שם תקין

The screenshot shows a Windows application window titled 'Form1' with a teal background. On the left, there are input fields for 'File Name' (containing 'c?'), 'File Data' (containing 'dkjm'), and 'File Type' (a dropdown menu set to 'DOC'). Above these are buttons for 'Add default file', 'Add new File:', 'Add', and 'Remove file'. On the right, there is a 'Search Files By Type' dropdown menu, a 'Files List:' label, and buttons for 'Print' and 'Select'. A modal dialog box is centered over the application, displaying the message 'please enter a valid name' with an 'OK' button. At the bottom, there is a button labeled 'Find the Largest file:' followed by a horizontal bar.

במידה ואחד הערכים לא הוכנס ע"י המשתמש תוצג ההודעה הבאה והתאים ירוקנו אוטומטית:

This screenshot shows the same application window 'Form1' but with different data. The 'File Name' field now contains 'gui' and the 'File Data' field contains 'cocomelon'. The 'File Type' dropdown menu is currently empty. The modal dialog box now displays the message 'You must enter all the details first!' with an 'OK' button. All other UI elements, including buttons and labels, remain the same as in the previous screenshot.

לאחר הכנסת ערכים תקינים:

Form1

Add default file

Add new File:

File Name: baby

File Data: cocomelon

File Type: TXT

Add

Remove file

Find the Largest file:

Files List:

Print

Search Files By Type

Select

New File added!

OK

בלחיצה על PRINT :

Form1

Add default file

Add new File:

File Name:

File Data:

File Type:

Add

Remove file

Find the Largest file:

Files List:

Print

Search Files By Type

Select

24/05/2023 19:12:17 0KB sameFile1

24/05/2023 19:12:17 0KB baby

בחיפוש קובץ ניתן לבחור ערך מרשימה של ערכים:

Form1

Add default file
Add new File:
File Name:
File Data:
File Type:
Add
Remove file

Files List:
Print
24/05/2023 19:12:17 0KB sameFile1  
24/05/2023 19:12:17 0KB baby

Search Files By Type
PDF
Select

Find the Largest file:

בלחיצה על כפתור SELECT :  
יופיע הקובץ המתאים לסוג הנבחר (אם קיים)

Form1

Add default file
Add new File:
File Name:
File Data:
File Type:
Add
Remove file

Files List:
Print
24/05/2023 19:12:17 0KB sameFile1  
24/05/2023 19:12:17 0KB baby

Search Files By Type
Select
24/05/2023 19:15:37 0KB sameFile1

Find the Largest file:

## בלחיצה על FIND THE LARGEST FILE יוחזר הקובץ הכי גדול (לפי התוכן) במידה וקיימים שני קבצים שונים

The screenshot shows a web application window titled 'Form1'. On the left, there are input fields for 'File Name', 'File Data', and 'File Type', along with buttons for 'Add default file', 'Add new File:', 'Add', and 'Remove file'. In the center, the 'Files List:' section contains a 'Print' button and a list of two files: '24/05/2023 19:12:17 0KB sameFile1' and '24/05/2023 19:12:17 0KB baby'. On the right, the 'Search Files By Type' section has a dropdown menu and a 'Select' button. At the bottom, the 'Find the Largest file:' button is highlighted, and the text '24/05/2023 19:17:18 0KB baby' is displayed next to it.

## בלחיצה על כפתור REMOVE FILE : תופיע ההודעה הבאה ולאחר מכן ירוקנו כל תיבות הטקסט

This screenshot shows the same 'Form1' application window, but with a confirmation dialog box overlaid in the center. The dialog box has a purple header with a close button (X) and contains the text 'Oldest file removed successfully!' with an 'OK' button below it. The background interface remains the same, showing the file list and search options.

Form1

Add default file

Add new File:

File Name:

File Data:

File Type:

Add

Remove file

Files List:

Print

Search Files By Type

Select

24/05/2023 19:20:30 0KB baby

Find the Largest file:

בלחיצה על FIND THE LARGEST FILE :  
במידה וקיים קובץ אחד

Form1

Add default file

Add new File:

File Name:

File Data:

File Type:

Add

Remove file

Files List:

Print

Search Files By Type

Select

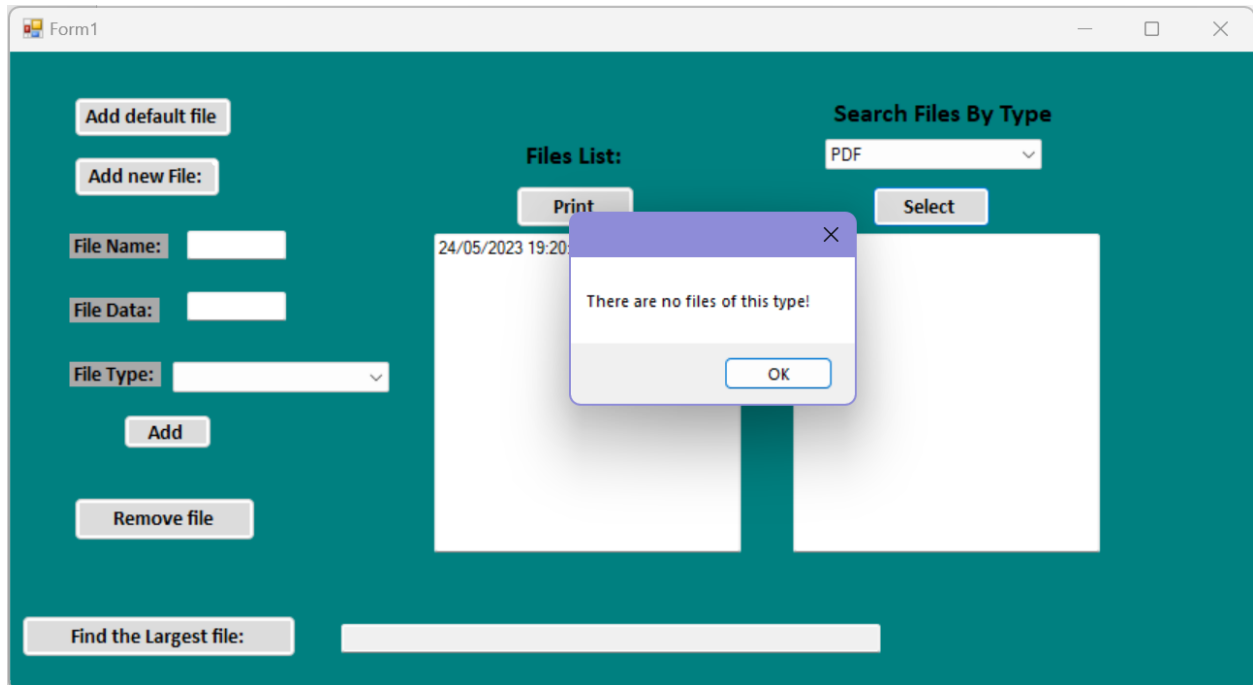
24/05/2023 19:20:30 0

There is only one file !

OK

Find the Largest file:

בלחיצה על FIND THE LARGEST FILE במידה ואין קבצים מהסוג הנבחר:



חלק 2 :

CLASS program:

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

namespace WindowsFormsApp1_GUIOrder
{
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```



```

    }
}

```

### CLASS Pizza:

```

using System;
using System.Collections.Generic;
using System.Diagnostics.Tracing;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Xml.Linq;
using System.Xml.Schema;

namespace WindowsFormsApp1_GUIOrder
{
    internal class Pizza
    {
        string toppingname;
        string size;
        int price;
        public Pizza()
        {
            toppingname = "Olives";
            size = "Family size";
            price = 33;
        }
        public Pizza (string toppingname, string size)
        {
            this.toppingname = toppingname;
            this.size = size;
            if (size.Equals("Family size"))
                price = 30;
            else
                price = 15;
            if (!(toppingname == "None"))
                price += 3;
        }

        public int getPrice()
        {
            return price;
        }
        public string pizzatoString()
        {
            string res = "\n\r Your Pizza: " + size + " with " + toppingname + "
price: " + price+"\n\r";

            return res;
        }
    }
}

```

### CLASS Drinks:

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1_GUIOrder
{
    internal class Drinks
    {
        string name;
        int price;
        public Drinks (string name)
        {
            this.name = name;
            this.price = drinkPrice(name);
        }

        public int getPrice()
        {
            return price;
        }
        private int drinkPrice(string drinkname)
        {
            switch(drinkname)
            {
                case "None":
                    return 0;

                case "Coke":
                    return 14;

                case "Sprite":
                    return 12;

                case "Fanta":
                    return 12;

                case "Water":
                    return 10;
            }
            return 0;
        }
        public string drinktoString()
        {
            string res = "\n\r Your Drink: " + name + " price: " + price+"\n\r";

            return res;
        }
    }
}

```

**CLASS Orders:**

```

using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;

namespace WindowsFormsApp1_GUIOrder
{
    internal class Orders
    {
        Pizza[] pizzaArr = new Pizza[0];
        public int pizzaindex = 0;
        Drinks[] drinkArr = new Drinks[0];
        public int indexdrink = 0;
        static int totalprice=0;
        public Orders()
        {
            pizzaArr = new Pizza[0];
            drinkArr = new Drinks[0];
        }
        public void total()
        {
            totalprice = 0;
        }
        public void addNewPizza(Pizza yourpizza)
        {
            Array.Resize(ref pizzaArr, pizzaArr.Length + 1);
            pizzaArr[pizzaindex++] = yourpizza;
            totalprice+=yourpizza.getPrice();
        }
        public void addNewDrink(Drinks yourdrink)
        {
            Array.Resize(ref drinkArr, drinkArr.Length + 1);
            drinkArr[indexdrink++] = yourdrink;
            totalprice+=yourdrink.getPrice();
        }
        public string orderdetails()
        {
            string res=null;
            for (int i = 0; i < pizzaArr.Length; i++)
            {
                if (pizzaArr[i] != null)
                {
                    res += pizzaArr[i].pizzatoString() + " ";
                    res += "\n";
                }
            }
            for (int i = 0; i < drinkArr.Length; i++)
            {
                if (drinkArr[i] != null)
                {
                    res += drinkArr[i].drinktoString();
                    res += "\n";
                }
            }
        }
    }
}

```

```

        }
    }
    if(totalprice!=0)
    res += "\n\r Total :" + totalprice;
    return res;
}
public void deleteArr()
{
    for (int i = pizzaArr.Length; i >0; i--)
    {
        Array.Resize(ref pizzaArr, pizzaArr.Length - 1);
        pizzaindex--;
    }
    for (int i = drinkArr.Length; i >0; i--)
    {
        Array.Resize(ref drinkArr, drinkArr.Length - 1);
        indexdrink--;
    }
}

}
}

```

### **CLASS Form1:**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Security.Authentication.ExtendedProtection;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1_GUIOrder
{
    public partial class Form1 : Form
    {
        private int remainingTime = 300;
        Orders yourOrder = new Orders();
        static int ordernum = 1;
        public Form1()
        {
            InitializeComponent();
            timer.Interval = 1000;
            timer.Tick += timer_Tick;
            timer.Start();
        }
    }
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
    clearform();
}
private void clearform()
{
    btnNewpizza.Visible = false;
    btnOurpizza.Visible = false;
    cmbpizzasize.Visible = false;
    cmbpizzatopping.Visible = false;
    cmbdrinkoption.Visible = false;
    btnAdddrink.Visible = false;
    btnAddpizza.Visible = false;
    lbldrinkchoice.Visible = false;
    lblpizza.Visible = false;
    lbltopping.Visible = false;
    btnPay.Visible = false;
    btnClear.Visible = false;
    txtOrdersummary.Visible = false;
    btnOrdersummary.Visible = false;
    lbldone.Visible = false;
    lblpaymentdeets.Visible = false;
    txtordernumber.Visible = false;
    btncostumer.Visible = false;
}
private void buttoncontrol()
{
    btnPay.Visible = true;
    btnOrdersummary.Visible = true;
    txtOrdersummary.Visible = true;
    btnClear.Visible = true;
}
private void btnOurpizza_Click(object sender, EventArgs e)
{
    Pizza ourpizza = new Pizza();
    yourOrder.addNewPizza(ourpizza);
    MessageBox.Show("Pizza selected!");
    buttoncontrol();
}
private void btnNewpizza_Click(object sender, EventArgs e)
{
    cmbpizzasize.Visible = true;
    cmbpizzatopping.Visible = true;
    btnAddpizza.Visible = true;
    lblpizza.Visible = true;
    lbltopping.Visible = true;
}
private void btnAddpizza_Click(object sender, EventArgs e)
{
    if (cmbpizzasize.Text == "" || cmbpizzatopping.Text == "")
    {
        MessageBox.Show("You must choose the size and the topping!");
        cmbpizzasize.SelectedIndex = -1;
        cmbpizzatopping.SelectedIndex = -1;
    }
    else
    {

```

```

        Pizza newpizza = new Pizza(cmbpizzatopping.Text, cmbpizzasize.Text);
        yourOrder.addNewPizza(newpizza);
        MessageBox.Show("pizza selected");
        cmbpizzasize.SelectedIndex = -1;
        cmbpizzatopping.SelectedIndex = -1;
        buttoncontrol();
    }
}
private void btnAdddrink_Click(object sender, EventArgs e)
{
    if (cmbdrinkoption.Text == "")
        MessageBox.Show("You must choose a drink first");
    else
    {
        MessageBox.Show("drink selected");
        Drinks newdrink = new Drinks(cmbdrinkoption.Text);
        yourOrder.addNewDrink(newdrink);
        cmbdrinkoption.SelectedIndex = -1;
        buttoncontrol();
    }
}
private void btnOrdersummary_Click(object sender, EventArgs e)
{
    txtOrdersummary.Text = yourOrder.orderdetails();
}

private void btnNeworder_Click(object sender, EventArgs e)
{
    btnNewpizza.Visible = true;
    btnOurpizza.Visible = true;
    lbldrinkchoice.Visible = true;
    btnAdddrink.Visible = true;
    cmbdrinkoption.Visible = true;
}
private void btnClear_Click(object sender, EventArgs e)
{
    while(yourOrder.indexdrink != 0 || yourOrder.pizzaindex != 0)
    {
        yourOrder.deleteArr();
    }
    yourOrder.total();
    MessageBox.Show("Deleting your order");
    txtOrdersummary.Clear();
}
private void btnDone_Click(object sender, EventArgs e)
{
    if (yourOrder != null)
    {
        this.BackColor = Color.Wheat;
        txtordernumber.Text = "ORDER #" + ordernum++;
        clearform();
        btnNeworder.Visible = false;
        lstmenu.Visible = false;
        lbldone.Visible = true;
        lblpaymentdeets.Visible = true;
        txtordernumber.Visible = true;
        btncostumer.Visible = true;
        timer.Stop();
    }
}

```

```

        lblcountdown.Visible = false;
    }
}
private void resetForm()
{
    btnNeworder.Visible = true;
    lblcountdown.Visible = true;
    lstmenu.Visible=true;
    lbldone.Visible = false;
    lblpaymentdeets.Visible = false;
    txtordernumber.Visible = false;
    btncostumer.Visible = false;
    RestartTimer();
    yourOrder.total();
    txtOrdersummary.Text = "";
    while (yourOrder.indexdrink != 0 || yourOrder.pizzaindex != 0)
    {
        yourOrder.deleteArr();
    }
}

private void timer_Tick(object sender, EventArgs e)
{
    remainingTime--;

    if (remainingTime >= 0)
    {
        lblcountdown.Text =
TimeSpan.FromSeconds(remainingTime).ToString(@"mm\:ss");
        TimeSpan timeSpan = TimeSpan.FromSeconds(remainingTime);
        lblcountdown.Text = timeSpan.ToString(@"mm\:ss");
    }
    else
    {
        timer.Stop();
        lblcountdown.Text = "Time's up! deleting your order..";
        RestartTimer();
        txtOrdersummary.Clear();
        yourOrder.deleteArr();
        yourOrder.total();
    }
}
private void RestartTimer()
{
    remainingTime = 300;
    timer.Start();
}

private void btncostumer_Click(object sender, EventArgs e)
{
    resetForm();
}
}
}

```

**קבצי פלט:**

**המערכת שלנו היא עמדת רכישה המוצבת בכניסה לפיצרייה שבעזרתה ניתן לבצע הזמנה אך התשלום מתבצע בדלפק . יש חמש דקות לביצוע ההזמנה**

**בלחיצה על PLACE YOUR ORDER :** הטיימר מתחיל לפעול ותפריט הפיצרייה מוצג תיפתח אופציה להזמנת "פיצת הבית" או פיצה בהרכבה אישית , בנוסף ניתן להזמין רק שתייה

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

**בלחיצה על PICK OUR SIGNATURE PIZZA :** תופיע הודעה ש "פיצת הבית" התווספה להזמנה ובנוסף תיפתח האופציה להדפסת ההזמנה הכוללת, צפייה בסכום המצטבר ומחיקת ההזמנה



MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

04:50

Choose your drink:

Add drink

Order summary

Clear All

Done

03:52

### בלחיצה על MAKE YOUR OWN PIZZA :

נפתחת האופציה לבחירת גודל פיצה ותוספת ולאחר ההוספה (שתתבצע רק במידה ונבחר גם הגודל וגם התוספת) תופיע הודעה שהפיצה התווספה להזמנה

Form1

Place your Order

Pick Our signature Pizza

Make your own Pizza

Pizza size: Choose topping: Choose your drink:

Add pizza Add drink

Order summary

01:57

Clear All Done

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

Form1

Place your Order

Pick Our signature Pizza

Make your own Pizza

Pizza size: Choose topping: Choose your drink:

Add pizza Add drink

Order summary

00:56

Clear All Done

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

pizza selected

OK

### בלחיצה על CHOOSE YOUR DRINK :

רק במידה ובבחר ערך מהרשימה הקיימת תופיע הודעה שהמשקה התווסף להזמנה



MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
	3
	3
	14
	10
	12
	12

drink selected

OK

### בלחיצה על ORDER SUMMARY:

יופיע פירוט מלא של ההזמנה כולל מחיר לכל פריט ומחיר כולל להזמנה כולה

Order summary

02:36

Your Pizza: Family size with Olives price: 33  
Your Pizza: Small with Tuna price: 18  
Your Drink: Coke price: 14  
Total :65

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

### בלחיצה על CLEAR ALL:

תופיע הודעה שמתבצעת מחיקה של ההזמנה ולאחר מכן תיבת הטקסט של פרטי ההזמנה תתרוקן

Form1

Place your Order

Pick Our signature Pizza

Make your own Pizza

Pizza size: Choose topping: Choose your drink:

Add pizza Add drink

Order summary

02:09

Your Pizza: Family size with Olives price: 33  
Your Pizza: Small with Tuna price: 18 Your Drink: Coke price: 14  
Total :65

Clear All

Deleting your order

OK

Done

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

Form1

Place your Order

Pick Our signature Pizza

Make your own Pizza

Pizza size: Choose topping: Choose your drink:

Add pizza Add drink

Order summary

01:41

Clear All

Done

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

### כשנגמר הזמן שבטיימר :

יופיע כיתוב שנגמר הזמן ומתבצעת מחיקה של ההזמנה , ותיבת הטקסט של פרטי ההזמנה תתרוקן.



Form1

Place your Order

Pick Our signature Pizza

Make your own Pizza

Choose your drink:

Order summary

Time's up! deleting your order..

Clear All

Done

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

**בלחיצה על DONE :** תופיע מספר ההזמנה של הלקוח, זמן הכנה והוראות תשלום

Form1

ORDER #1

Your order will be ready in 30 minutes!

Payment at the counter

OK

**בלחיצה על OK :**

ההזמנה מתאפסת וגם הטיימר נפתח החלון להזמנה חדשה עבור הלקוח החדש

Form1

Place your Order

04:39

MENU	Price
Our pizza :Family sized with Olives	33
Family size pizza	30
small pizza	15
Olives	3
Tuna	3
Mushrooms	3
Onions	3
DRINKS:	
Coke	14
Water	10
Sprite	12
Fanta	12

לאחר ביצוע הזמנה חדשה , אפשר לראות שמספר ההזמנה התעדכן בהתאם לכמות ההזמנות שהתבצעו

Form1

ORDER #2

Your order will be ready in 30 minutes!

Payment at the counter

OK