موضوع الدرس: المحاضرة 1 — اليوم: السبت

التاريخ: ٣/١٨/٢٠٢٤م — الموافق: ٢٨/١/١٤٤٩هـ

- ال namespace مساحة عمل :
- العمود الفقري في البرمجة المهيكلة لماذا؟! لأنه يمكن تقسيمها ووضعها في عدة ملفات بنفس الاسم (أي مساحة عمل لا تقنية الملفات) حيث أن ال Compiler يفهمها ويعاملها بمساحة عمل واحدة كلما لقى namespace بنفس الاسم.

وهذه الفكرة ليست موجودة في ال Multifile program

- ال class بحد ذاته يعتمد على مبدأ ال namespace. حيث أن ال Compiler يعمل على فتح مساحة عمل باسم ال class ويضع فيه الدوال والمتغيرات العامة وغيّر تركيبة الدوال ومارت class.
- ال class كمفهوم غير ال class في ال Compiler.
- ∴ ال namespace عبارة عن مساحة عمل تعتبر بالنسبة لنا مكتبة library، توزع في أكثر من ملف بحيث أن ال Compiler يقوم بسحبها ووضعها في مساحة عمل واحدة (model واحد في الذاكرة).
- ال namespace تحتوي على متغيرات - دوال - class - struct - union - ثوابت - Nested namespace - enum. هذا بالنسبة لـ C++

في ال C# تلغي المتغيرات لأنه عبارة عن classes (كله كلاسات).

الواجب:

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠ م الموافق: / / ١٤ هـ

- لغة الـ C# :

هي لغة تعتمد على الـ oop وهنا سيكون في الـ namespace pure oop أي نبني class فقط ما حتى الـ main() نفتح لها class وتكون متغيرها static وهنا الـ Compiler يعرفها.

الفرق بينها وبين الـ C++ هو إننا نأخذ الـ main() و نكتب قبلها static ونوضعها داخل class وهذا الـ class يتم تسميته بأي اسم.

```
namespace ConsoleApplication
{ class program
    { static void Main (string [] args)
      { int x;
        int y;
        x = 100;
        y = 50;
        int z = x+y; }
    }
}
```

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

- عند كتابة أي برنامج نحدد المدخلات والمخرجات

مثال برنامج جمع عددين

المدخلات عددين المخرجات عدد واحد

∴ سنحتاج إلى ثلاثة متغيرات

- إدخال القيم إما بال cin أو بالإسناد.

- ال cin وال cout عبارة عن object في ال C++

نقلب الفكرة شوية ونعتبرها class،

في ال cin دالة اسمها get . نعتبرها ساكن.

ويتم استدعاءها كالتالي ⇐ cin::get()

وال Case caale (علامة >>) نعتبرها ساكن.

وال cout نفس الكلام

نريد عمل ال cin وال cout في class واحد.

⇐ باستخدام الوراثة - نعمل class واحدة ونشتقها من الاثنين.

```
class Console : public cin , public cout
{ }
```

وعند الاستدعاء

Console::get() → Console.get()

Console::put() → Console.put()

وهنا يظهر أحد الفروقات في ال classes بين ال C++ وال C#

وهو أن ال :: تستبدل بـ . في حالة الدالة الساكنة وهذا للتسهيل.

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

∴ هنا سنستخدم class واحدة لل Input/output (class وليس object كما في ال C++).

ال Console فيها دوال ال Input/output معاً وكلها static.

- دوال تتبع ال Console :

- Read للقراءة
- Readline قراءة ويترك سطر (محتمل ال endl في الأخير).
- write طباعة
- writeline يطبع ويترك سطر (محتمل ال endl في الأخير).
- Readkey قراءة حرف وتكافئ getch في ال C++.

- أي class فيها دوال ساكنة هي library «مكتبة».
- في ال C# الطباعة لا تكون إلا نصوص فهو لا يتعامل مع format ال int وال float لذلك سيتم تحويلها إلى نص.
- في ال C# تم التزويد بإمكانية التحويل من وإلى النصوص «من وإلى كل الأنواع بشكل عام».
- ما هي الأنواع التي سيتم التعامل معها ؟!

في ال C++ ⇐ int, long, short, char, double, float

في ال C# ⇐ short ⇒ int 16 , int ⇒ int 32

long ⇒ int 64 , char , byte , bool ,

string , float , double , decimal , DataTime.

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

- الـ string :: متضمنة في مكتبة على أساس أنها datatype وفيه كل الإمكانيات التي درسناها.
- unsigned :: في حالة العدد الموجب يزداد الحجم بمقدار بت.

$2^{10} \rightarrow 2^{11}$

- أنواع الـ Datatype للـ C# عبارة عن class أي يتم إنشاء object والـ object يتبعه دوال أو Data، أي عندما نقول int x الـ x هنا object أي يتبعه member function - methode.

```
class program
{ public void main (string [] args)
  { int x , y;
    x = 100;
    y = 50;
    int z = x + y;
    Console.Write(z);
    Console.WriteLine (z.ToString());
    int x2;
    Console.Write ("x2 :");
    string t = Console.ReadLine();
    x2 = Convert.ToInt32(t);
```

دالة تحويل إلى نص

تعامل مع نصوص

cin>>x

دالة تحويل إلى int

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠ م الموافق : / / ١٤ هـ

```
int y2; Console.Write("y2 :");
t = Console.ReadLine();
y2 = Convert.ToInt32(t);
int z2 = x2 + y2;
Console.Write("z2 :");
Console.WriteLine(z2.ToString());
t = Convert.ToString(z2);
Console.WriteLine(t);
Console.WriteLine("z2 :" + t);
Console.ReadKey(); // ≡ system("pause");
static int add (int x, int y) //
{ return x + y; }
}
```

} cin>>y

ثلاث طرق لطباعة z2 ≡ cout<<z

}

نكتب الدوال داخل الكلاس الـ program في أي مكان

```
int i=0
100: i=i+1
cout << i;
if (i<10)
goto 100;
```

```
int i = 0;
do {
i = i + 1;
cout << i; }
while (i<10);
```

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
class program
{ public void main( string [] args)
  { int x ,y ;
      x = 100;
      y = 50;
      int z = add (x,y);
      Console.WriteLine (z.ToString());}
 static int add (int x ,int y)
 { int temp;
      temp = x+y;
      return temp;}
 }
- namespace Console Application
 { class distance
   { private int kilo =0;
      private float meter =0;
      public distance ()
        { }
      public distance(int k , float m)
        { kilo = k;
          meter = m; }
```

الواجب : .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠ م الموافق: / / ١٤ هـ

```
public void print ()
 { Console.WriteLine("kilo :" + kilo.ToString())
   Console.WriteLine("Meter :" + meter.ToString());
 }   //                    اهمل دوال التغليف
}

class program.
{ public void main (string [] args)
  { distance d = new distance (10, 20); //
  أي كائن بنيناه (غير الـ Simple) طريقة تعريفه كأنه مؤشر
    d.print ();
    d.print_kilo();
    d.print_meter ();
    int [] a = new int [10]; // تعريف مصفوفة
    a[0] = 10;  a[1] = 20;
    for (int i = 0; i < 10; i++)
     { t = Console.ReadLine();
       a[i] = Convert.ToInt32 (t); }
    for (int i = 0; i < 10; i++)
    { Console.Write (a[i]);
      Console.Write (" "); }
    Console.ReadKey(); }
 }
}
```

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

المكتبات المدروسة حالياً :

Console Convert Math

وهي مكتبات تحتوي على دوال ساكنة .

- في البرمجة المرئية نسمي الكود (الخطوات البرمجية) بال event .
- متى سينفذ التعليمة . عند ضغط الزر .

عند ضغط الزر نسميه event click

وال event معناه حدث والحدث يرتبط بكلمة عند

عند الضغط (ضغطة واحدة) .

- كل object يتبعه خصائص وأهم خاصية هي خاصية القيمة

- اسم الخاصية هنا هي : Text .

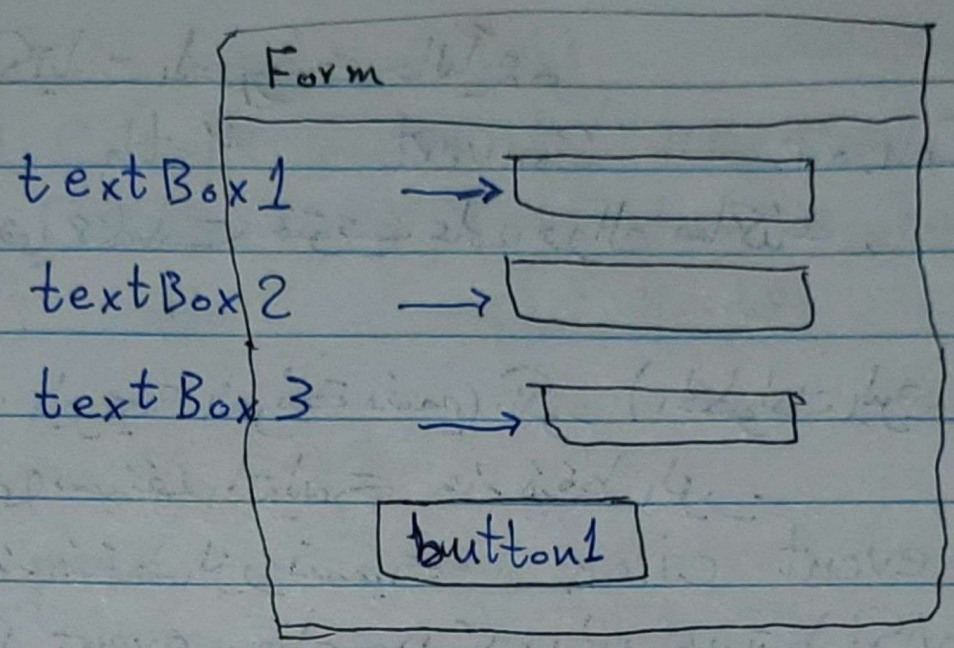
- في هذا الكورس سندرس الخصائص وتوظيفها .

- Lable أداة للكتابة .
- Form خاصية وهي نفس ال model .
- Text خاصية القيمة .
- click حدث (الضغط) .

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ




```
{ int x, y;
  float z;
  x = Convert.ToInt32(textBox1.Text);
  y = Convert.ToInt32(textBox2.Text);
  z = x + y;
  textBox3.Text = z.ToString();
}
```

// نعتبرها منطقة كتابة الـ main()

في المثال السابق :

- سيتم جمع الرقمين عن طريق الزر.
- استبدلنا الـ Console للإدخال والإخراج بالـ textBox
- استخدمنا خاصية الـ Text + Form .

الواجب : .....

موضوع الدرس: المحاضرة 2 — اليوم: السبت

التاريخ: ٢٠٢٤/٨/١٧م — الموافق: ١٤٤٦/٢/١٣هـ

- درسنا نوعان من المشاريع :

1] Console Programming.

2] Windows Forms.

- يمكننا عمل أكثر من main() داخل الـ class وتحدد الـ main() الرئيسية من زر الـ Properties.

- إذا قمت بعمل برنامج عادي انشئ شيء اسمه class Program واكتب برنامجك بشكل طبيعي داخل الـ main() وإذا أردت عمل دوال اكتبها في أي مكان فوق أو تحت الدالة الرئيسية وتضيف لها خاصية الـ static.

- طريقتان لعمل class :

1] داخل الـ namespace نفسه.

2] نفتح ملف جديد ونكتبه داخل الـ namespace ويكون اسمها نفس اسم الـ namespace الموجودة في الملف الأول.

مبدأ الـ Multifile

⇐ في class distance :

في دالة الطباعة إما نطبق الـ kilo والـ meter كل واحد لحاله أو باستخدام الـ + أو باستخدام الـ formate

• تعد البارامتر بيبدأ {0} أول عنصر و {1} ثاني عنصر و {2} ثالث عنصر ... مع إمكانية إضافة تصوص •

الواجب:

العنصر الأول

```
public void print ()
{ Console.WriteLine("Kilo = {0} \n
Meter = {1}", kilo.ToString(), meter.ToString()); }
```

العنصر الثاني

{0}

{1}

- لكي نكون كائن من نوع كلاس نحن بنبنيه ضروري نستخدم new

```
distance d = new distance();
distance d2 = new distance (10, 20);
```

- إذا كبر لدينا الكلاس نقوم بتقسيمه وذلك بإضافة كلمة partial، نعمل في الملف الأول distance1 والملف الثاني distance2 ... ونجمعهم داخل كلاس distance ونضيف للأجزاء كلمة partial

- أسلوب البرمجة تطور من حيث مبدأ التنظيم حيث انتقل إلى مساحات العمل أو إلى نظام الكلاسات أو الكلاس الجزئي ... إلخ.

- إذا أردنا عمل مصفوفة من نوع distance

```
distance [] d = new distance [10];
```

- هنا أنشأنا عشرة مؤشرات حيث كل object يحتاج إلى new

```
for (int i = 0; i < 10; i++)
    d[i] = new distance();
```

الـ d بحد ذاته هو مؤشر والمصفوفة هي مؤشر.

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- قراءة مصفوفة عن طريق دالة :

```
static int [] ReadArray (int n)
 { int [] tp = new int [n];
   for (int i=0; i<n; i++)
     { Console.Write ("Insert a number :");
       string temp = Console.ReadLine();
       tp[i] = Convert.ToInt32 (temp); }
   return tp; }
```

تقرأ المصفوفة وترجع عن طريق اسم الدالة، أي أن الدالة هنا تعيد مصفوفات.

- إذا أردنا أن نعيد المصفوفة عن طريق البارامتر نفسه نجعل الدالة void ونحذف ال return.
- في هذا الكورس نحن ندرس فكر ال visual.
- طبيعة ال Form أن يبيّن العمل في نفس اللحظة.

// الجمع //

```
{ int x, y, z;
x = Convert.ToInt32 (textBox1.Text).Trim();
y = Convert.ToInt32(textBox2.Text.Trim());
z = x + y;
textBox3.Text = Convert.ToString (z); }
```

الواجب : .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

```
//                    // القسمة
{ int x, y, z;
  x = Convert.ToInt32(textBox1.Text.Trim());
  y = Convert.ToInt32(textBox2.Text.Trim());
  if (y == 0)
  { MessageBox.Show("Error");
    textBox3.Clear(); }
  else
  { z = x / y;
    textBox3.Text = Convert.ToString(z); }
}
//                    // إفراغ المناديق
{ textBox1.Clear();
  textBox2.Clear();
  textBox3.Clear();
  textBox1.Focus(); //   من باب التسهيل على المستخدم
}
//                    // الخروج
{ this.Close();
}
```

الواجب: .....

ثلاث طرق لحذف النص :

1] textBox1.Clear ();
2] textBox1.Text = " ";
3] textBox1.Text = null;

- clear() دالة عاملة على البيان method ، تقوم بحذف النص ، تحذف ما داخل الصناديق .
- Focus() دالة عاملة تنقل التركيز إلى صف محدد .
- this مؤشر يؤشر على نفس نوع الكائن .
- ال textBox هو بحد ذاته كائن له دوال وبيانات ، البيان الوحيد الذي يتعامل معه هو ال text .
- Close() دالة عاملة تعمل على إغلاق ال Form . الكائن هنا لا ينهدم وإنما يتحول إلى حالة يسمى disbonse وليس null .
- خاصية ال Read Only للقراءة فقط .
- \r\n نزول سطر
- Trim() method تزيل المسافات من الطرفين .
- label فيها خاصية ال Text
- switch لها امكانيات ومرونة عالية في ال C# أفضل من C العادية .

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

// في حال كان لدينا صندوق يعبر عن العملية

```
{ int x, y, z;
  x = Convert.ToInt32(textBox1.Text.Trim());
  y = Convert.ToInt32(textBox2.Text.Trim());
  string op = textBox4.Text.Trim();
  switch (op)
  {case "+": z = x+y;
             textBox3.Text = Convert.ToString(z);
             break;
   case "-": z = x - y;
             textBox3.Text = Convert.ToString(z));
             break;
   case "*": z = x * y;
             textBox3.Text = Convert.ToString(z));
             break;
   case "/": z = x / y;
             textBox3.Text = Convert.ToString(z));
             break;
   default: MessageBox.Show(" ");
            textBox4.Focus();
  }
}
```

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

* تكليف ⇐ نكتب المثال السابق باستخدام if

- في المثال السابق كان في حال تحديد العملية عن طريق الإدخال.
لكن إذا كانت أريدة للاختيار فقط ؟!
نستخدم الـ ComboBox

```
{ int x , y , z ;
x = Convert.ToInt32(textBox1.Text.Trim());
y = Convert.ToInt32(textBox2.Text.Trim());
int n = ComboBox1.SelectedIndex ; //
```

هنا يتم التعامل على أن الـ 0 هو العنصر الأول والـ 1 هو العنصر الثاني ... وهكذا
-1 لا شيء أي لم يتم اختيار أي خيار من الصندوق.

```
switch (n)
{ case 0: z = x+y ;
          textBox3.Text = Convert.ToString(z);
          break;
          :
          :
}
```

⇐ المثال السابق بينا الخيارات باستخدام SelectedIndex
ويمكننا بناءها بطريقة أخرى باستخدام SelectedItem
⇐

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
{int x ,y ,z;
x = Convert.ToInt32(textBox1.Text.Trim());
y = Convert.ToInt32(textBox2.Text.Trim());
string op = ComboBox1.SelectedItem.ToString();
switch (op)
{ case "+": z = x+y;
            textBox3.Text = Convert.ToString(z);
            break;
            :
}
```

- أيهما أفضل index or item ؟!

كل له حالته، لكن الـ text تفضل لأنها قابلة للفهم أثناء كتابة الكود «شكل الكود يكون واضح»، وفي حالة إضافة أدوات تحدث خريطة إذا استخدمنا الـ index. لهذه الأسباب الـ text تفضل عن الـ index.

- إذا كانت العمليات محصورة بحدود معينة يفضل اختيار صندوق اختياري إما بالأزرار أو بالـ ComboBox.
- الـ ComboBox فيها خاصية الـ Text.
- قائمة الـ ComboBox لا توجد فيها خاصية اسمها الـ Text والـ SelectedItem تقوم بدور الـ Text هنا.

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- إذا أردنا فتح Form جديد :

- Project + new Windows Form

(نوع من نفس الـ object)

- add + Windows Form.

- التنفيذ من الدالة main() يعني إذا فتحنا أكثر من Form

نغير الـ Form المراد تشغيله داخل الـ main()

- الحدث click للأزرار بينما الحدث load الـ Form

يشتغل عند تنفيذ الـ Form. وهو يشبه الـ Constructor

حيث يرتبط بالـ Form عندما يعمل.

- أداة الـ List هي قائمة الـ ComboBox وليست فيها خاصية

الـ Text لذلك نستخدم الـ Item حيث أنها تقوم بدور

الـ Text هنا.

```
ListBox1.SelectedItem.ToString().Trim();
```

// حل مشكلة إذا كانت الصناديق فارغة

```
{ int x = 0, y = 0 , z = 0;
 if ( textBox1.Text != null) && (textBox2.Text
                    != null)
  {x=Convert.ToInt32(textBox1.Text.Trim());
   y=Convert.ToInt32(textBox2.Text.Trim());
   ;
   ;
   ;
  }
```

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
else
{ MessageBox.Show (" ");
  TextBox1.Focus (); }
}
//                    إذا كانت أحد المتاديق فاضية
{ int x = 0 ; y = 0 , z = 0 ;
 if ( TextBox1.Text == null)
  { MessageBox.Show (" ");
    TextBox1.focus();
    textBox1.BackColor = Color.Red;
    return ; }
else if (textBox2.Text == null)
  { MessageBox.Show (" ");
    textBox2.focus();
    textBox2.BackColor = Color.Red;
    return ; }
else
{ :
  :
 }
}
```

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- الكود السابق يحرك أين الصندوق الفارغ ويروح لعنده .
- الـreturn في المثال السابق مكروهه بالعمل، حيث لا نريد القطع دائماً .
- BackColor تغير لون الصندوق .
- ForeColor تغير لون النص
- الرسائل التوضيحية يتم عملها عن طريق الـ Label
- التحذيرية يتم عملها عن طريق الـ MessageBox
- الـ Timer مؤقت يستخدم لعمل كود ينفذ مدة معينة من الزمن ثم يتوقف مثلاً تغير لون الصندوق لمدة دقيقة هنا سيستمر التغيير لمدة دقيقة ثم يعود الى لونه السابق . سيتم دراسته لاحقاً .

الواجب : .....

موضوع الدرس: المحاضرة 3 — اليوم: الأحد

التاريخ: ٢٠٢٤/٩/١م — الموافق: ١٤٤٦/٣/٢٨هـ

- مشكلة ← فورم Form

١) تحديد المشكلة ← هدف ← المخرجات ← الركائز الأساسية

← المدخلات

٢) المعالجة Processing

- التحليل في مشكلة الـ Form تتطلب تحديد المدخلات و المخرجات والمعالجة وهي الركائز الأساسية

- المعالجة هي مراحل نقل أي كيف انتقل من الـ input إلى الـ output مثله على مستوى الطاحون

input ⟶ processing ⟶ output

حبوب ⟶ طحن ⟶ طحين

⇐ في حالة الـ Form :

١) ما هي الأدوات المناسبة لكل input .

٢) ما هي الأدوات المناسبة لكل output .

٣) المعالجة ← ما هي أدوات المعالجة مثل Button ...

← ما هي أحداث هذه الأدوات مثل Click, load ...

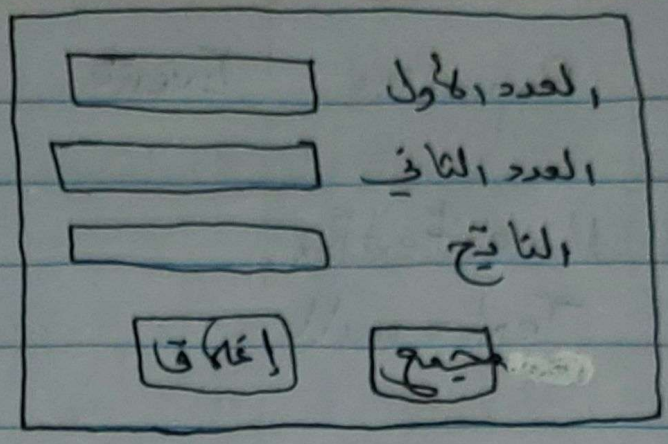
• قد تكون الأدوات التي سحبتها للـ input والـ output هي أدوات معالجة والـ Form بحد ذاته أداة للمعالجة

٤) تحديد خصائص كل أداة من الأدوات السابقة إبتداءً من جدول الخصائص .

الواجب:

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ



- مشكلة جمع رقمين :

١ر ل input ⇐ رقمين ٢ر ل output ⇐ رقم

٣ر الأدوات المناسبة :

3 ⇒ TextBox 3 → Label 2 → Button

جدول الخصائص :

Tools or objects	Properties	Events	ملاحظات
Form	Name = Form Text = " جمع رقمين "	Load تدخله عن طريق تعريفه	وصف شكل أداة
textBox1	Name = textBox1 Text = " " Top = 20 width = 30		

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

Tools or objects	Properties	Events	ملاحظات
TextBox2	Name = TextBox2 Text = null		
TextBox3	Name = TextBox3 Text = " "		
Label1	Name = Label1 Text = "العدد الأول"		
Label2	Name = Label2 Text = "العدد الثاني"		
Label3	Name = Label3 Text = "الناتج"		
Button1	Name = Button1 Text = "جمع"	Click	
Button2	Name = Button2 Text = "إغلاق"	Click	

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

- المسميات التي تستخدمها مسميات تدل على المعنى.
- التحليل يتم بالورق وكل شيء في السابق يجب أن تعكسه على عملك مستقبلاً.
- يجب رسم الـ Form قبل عمل جدول الخصائص، حتى تحصل الأبعاد لكل أداة مثل Top, left ...
- الخصائص المفردة:
  - Top ← المسافة من بداية الـ Form
  - left ← المسافة من جهة اليسار
  - width ← العرض
  - hight ← الارتفاع
  - ⋮
- الخصائص السابقة تستعملها عند بناء الـ Form برمجياً لكن في البيئة يتم تجميع هذه الخصائص في خاصية واحدة تسمى الخاصية المركبة.

  وهي size ← width, hight ,,,

  location ← Top, left ,,,,
- عند كتابة مسميات الأدوات نكتبها أسماء مختصرة

txt Nb1 → textBox number 1,

lb Nb1 → label number 1,

Cmb Nb1 → ComboBox number1,

txt Res → text Box result,

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
Load ()
{ This.Text = "جمع رقمين";          //
  Label1.Text = "جمع رقمين";  //
```

غير مجدية

عنوان الـ Form الحالي وهي خاصية مشابهة للـ

```
  Form1.Text = "جمع رقمين";
//  جميع خصائص الـ Form يمكن نعتها بالاسم الـ name
}
```

```
Load ()
{ int flag = 1;          //
```

التغيير لأول مرة ← باللغة العربية

← للمرة الأخرى ← باللغة E

```
  if (flag == 1)
  { This.Text = "جمع رقمين";
    Label1.Text = "جمع رقمين";  //
```

هنا يكون مجدية

```
    TextBox1.left = 100;  //
```

يجب عليك معرفة

```
    TextBox1.Top = 100; }  //
```

القياسات للجمل

```
else                          //
```

تعرف هذا الأسلوب

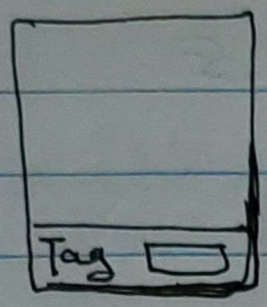
```
{ This.Text = "Sum 2 numbers";
  Label1.Text = "Summation";
  TextBox1.left = 100;
  textBox1.Top = 100; }
}  //
```

في هذا الأسلوب يلتغي التحكم بالماوس

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ



- «الملاحظة» هي خاصية اسمها Tag.
- له وظيفتين : إما يستخدم كتعليق أو يستدعى في أي لحظة داخل البرامج للمساعدة مثلاً.
- في بعض الأوقات لا يستخدم لكتابة ملاحظات وإنما يستخدم لكتابة نص آخر استخدامه برمجياً كقيمة إضافية. لكن قيمة غير مرئية.
- عندما ننزل أدوات معناه أننا أنشأنا object لداخل الـ Form «لداخل الـ Class».

```
public partial class Form0 : Form
{ private textBox textBox1 = new textBox();
  //                في الـ C# لكل object نعمل له new
  :
  :
  :
This.Control.add();   // إضافة مرئية
}
```

- عند الانتهاء من كتابة الخصائص يتم إضافتها مرئياً أما برمجياً فهي مضافة أساساً.
- "\r\n" أمر نزول سطر.
- Partial معناه أن الـ class قسم آخر.

الواجب : .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

```
{   //                    class Form تكملة
  void Form0_Load ()
  { This.Text = "جمع رقمين";
      ⋮
      //                بقية الخصائص
  ListBox1.Items.add ("+");
      ⋮
      //                بقية ال operator
      ⋮
  ListBox1.SelectedItem = "+";
  ListBox1.SelectedIndex = 0;
  ListBox1.Item[0]; }
}   //
```

إعطاء الـ defaulte حتى تعرف وين مؤشر

ـ ينطبق هذا على ال ComboBox أيضاً

- Item ⇐ خاصية مركبة على هيئة مصفوفة object حيث يتم التعامل مع كل عنصر على شكل object.

ListBox → Selected Item
→ Selected Index

```
//                شرط التحقق من أن المدخلات عبارة عن أرقام
if ((textBox1.Text.Trim() == "") || (!is_Numeric
        (textBox1.Text.Trim())))
      ⋮
```

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

// بناء دالة IsNumeric

```
bool IsNumeric (string str)
{ str = str.Trim();
  if (str == "")
      return false;
  for (int i=0; i<str.length; ++i)
      if ((str[i]<48) || (str[i]>57))
          return false;
  return true; }
```

- Try and catch

تصطاد الخطأ وتمنع قطع البرنامج حيث تسمح بالمواصلة في التنفيذ وتمسك الخطأ وأنت بالبرنامج.

```
try
{ x = Convert.ToInt32 (textBox1.Text);
  y = Convert.ToInt32 (textBox2.Text); }
catch (Exception exp)
{ MessageBox.show ("Error");
  textBox1.Focus();
  return; }
```

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

// في حال أردنا معرفة الخطأ في أي مكان

```
try
{x = Convert.ToInt32(textBox1.Text); }
catch (Exception exp)
{ MessageBox.show("Error in TextBox 1");
  textBox1.Focus();
  return; }
try
{ y = Convert.ToInt32(textBox2.Text); }
catch (Exception exp)
{ MessageBox.show("Error in TextBox 2");
  textBox2.Focus();
  return; }
```

⇐ الواجب: try && catch بدون return .

- مثال الأيقونة ico وكيف تزيد الحجم والارتفاع وكذلك التحريك .

الواجب : .....

موضوع الدرس: المحاضرة 4 — اليوم: السبت

التاريخ: ٧/٩/٢٠٢٤م — الموافق: ٤/٣/١٤٤٦هـ

- المدروس حتى الآن :

Classes	Tools	Properties	method	Events
Convert	Form	Text	Close	Constructor
Console	TextBox	Forcolor	Clear	Click
Math	Label	Back color	Focus	Load
Color	Button	Name		dbClick
	ComboBox	Font		
	List Box	Size — width, hight		
		Location — left, Top		
		SelectedItem		
		SelectedIndex		
		items		
		Length		

في الـ C++ كانت عبارة عن مكاتب include ... بينما الآن أصبحت كلاسات تحوي دوال ساكنة يتم استدعائها عن طريق (.)

iostream

تحولت

Console

- دالة المستمع تغيير حدث.

- يجب علينا في كل محاضرة إضافة دوال ومكاتب وخصائص و... جديدة مع كيفية الاستخدام والأمثلة للأدوات.

ListBox.Clear();

دالة الـ Clear تمسح الذي داخل الـ List Box كاملاً.

الواجب:

- عند فتح أكثر من Form، كيف نتعدد الـ Form الذي نحتاجه ؟!
نذهب إلى الكلاس الرئيسي program class ونكتب اسم الـ Form المراد تنفيذه، وهذه أشياء تقنية وليس لها أية علاقة بالبرمجة.
- بعض الخصائص في الـ Form :
- Text عنوان الـ Form.
- right layout, right to lift التصميم يبدأ من اليمين إلى اليسار.
- lift layout, lift to right التصميم يبدأ من اليسار إلى اليمين.
- ShowIcon إظهار الأيقونة.
- ShowInTask إظهاره في قائمة شريط المهام.
- minimizedBox تصغير الأيقونة «زر الإخفاء» ▭
- maximizedBox تكبير الأيقونة ▭
- ControlBox صندوق التحكم كاملاً ▭ ▭ ▭
وفي بعض الأحيان تجعله على شكل flat حيث تظهر على شكل لوحة ← Form بدون header على هيئة flat ←
وتغير الـ Background، لون أو صورة.
- Background إما img أو Color
باستخدام الأمر
BackGrounded Img or Ground Img layout.

- this.WindowState = FormWindowsState.Maximized; ⇐ عند تشغيل الـ Form يكون بملء الشاشة

- this.BackgroundImageLayout = ⟵ بعد = تظهر خيارات وهذه تسمى enum ، وتعتبره كلاس فيها static . ويمكننا استخدام أرقام لأنها ثوابت .

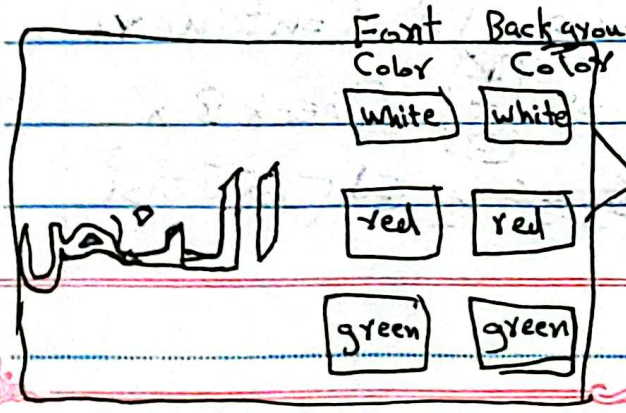
- التعامل مع الـ Form أيضاً يحتوي لقوانين البرمجة وقوانين الماوس ... الخ .

- أيضاً الحدث لديه ، في حال نقرنا نقرتين يشتغل الـ Load ، وإذا أردنا تغييرها يبشا إذا ضغطنا نقرتين تشتغل خاصية أخرى تغير عن فائدة لخصائصها .

- كل أداة لديها العشرات من الأحداث ، ولكي تصل إلى أي حدث من الأحداث تعتمد على صندوق الخصائص من الخيار event وهو يشبه الصاعقة ↯ .

- TextBox1.FontColor = ; ⟵ تعدد خيارات مطروحة داخل كلاس أو enum اسمها Color

- label1.TextBox1 = Color.White; ⟵ تغيير ألوان عند عملها بالأزرار



الشخص

باستخدام الأزرار Buttons

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- يمكننا استخدام ال ListBox و TextBox و ComboBox وغيرها بدلاً من الأزرار في المثال السابق.

- Radio Button :

أداة تستخدم إذا كان هناك خيارات متعددة وتمتاز بأنها تقبل خيار واحد فقط.

- ال Form يعتبر حاوية "يحتوي ما عليه".

- ال RadioButton يعتبروا مجموعة واحدة أي أنه لن تستطيع أخذ سوى خيار واحد فقط، ولحل هذه المشكلة نقوم بإنزال أداة حاوية بحيث تكون sub Form أي Form داخل Form وهي أداة بسيطة، نستخدم هذه الأداة لعمل Form داخل Form وهي في تلك اللحظة ال Contiener تسمى Banel أو GroupBox.

- ال Radio Button نضع فيها العناصر كأنها Form "هي تعتبر Form داخل Form".

- الفرق بين الأداتين Banel و GroupBox هو أن ال Banel ليس لديه عنوان.

- Continer أداة وهي حاوية مثل ال Form

- Check Box :

أداة تستخدم لتحديد عدة خيارات من مجموعة.

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

	Radio Button	Check Box
المميزة	خيار واحد	عدة خيارات
الخاصية	Checked	if-else

- وتنظيم داخل "Panel" "Group Box"

الواجب : .....

موضوع الدرس: المحاضرة 5 — اليوم: الثلاثاء

التاريخ: 10/9/2024م — الموافق: 7/3/1446هـ

- إذا فتحنا مصدر الكود نجد كلاس form1 مشتقة من form
هي كلاس وإذا ذهبنا إلى Program نجد بالـ main
عند Run ⇐ new form1
أي اعتبرنا أن الـ form1 كائن ⇐ أرسلت له كائن ⇐
في هذه الحالة أستطيع عمل:

```
class Program
{ // main
  { form1 f = new form1 (); // أعرف الكائن أنا وليس شرط أعتمد عليه لأنه class
    Run (f); // Run (new form1);
    // يفضل في هذه الحالة f تنهيه ⇐ تدمره ⇐ عند انتهاء عمله.
    f.dispose (); } // طريقة تدميره ⇐ على أساس أنه class عادي ⇐.
}
```

- بلغة C العادية نعرف كلاس بدون new ونعطيه البارامتر للـ Constractor
- بينما في الـ C# والجافا أي كلاس نعمل له new.
- كله بوينتر لكن ليس بهيئة البوينتر
- يستخدم إمكانية البوينتر بدون ما يعطيك شيء ⇐ الإمكانيات يستخدمها الكلاس وليس أنت ⇐.
- قاعدة الإحلال: في new form1 = f أي جعلنا new form1 تساوي f ⇐ إحلال في الرياضيات أي إحدى الطرفين ⇐

الواجب:

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠ م الموافق : / / ١٤ هـ

كلاس الـ Form1 موجود بملف وكلاس الـ Form بملف آخر

```
class Form1 : Form
{ public Form1 ()
  { Initialize compound () ;   // تعمل new لكل الأحداث
    // Partial                 موجودة معانا بالتعريفات
    click ()    //   خزن
    { }                                  نقصد بهم
    click ()    //   كتابة                الأزرة
    { }                                  بنيه كلاس .
}
main ()
{ object O = new Object ();
  distance d = new distance ();
  O = d ;        // implicit        تحويل ميهم
  distance d2 = new distance();
  d2 = (distance) O ;  // Explicit
                  // cast            تحويل نوع
  O.ToString () ;  //                تعمل لأنها موجودة بالكلاس
  //O. setkilo (10);                 في حالة الوصول لأي
  ((distance) O) .set kilo();        عضو ليس بالـ
}                                    Parent
```

خذ الـ O على هيئة شكل distance

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- C# مبنية على الوراثة .
- كلاس ال Object هو ال parent لأي كلاس متى ال Form
- أي أن كلاس ال Object يساوي أي كائن آخر داخل النظام كامل .
- في المثال السابق إذا قلنا d. تظهر method لا نبنيها نحن داخل كلاس ال distance1 مثل ToString() ، هذه الدوال جاءت من ورثها من كلاس object .
- الوراثة من كلاس object في اللغة أوتوماتيكياً في الخلفية وال C# يتم على وراثة إجبارية من هذا الكلاس object وتوجد داخلها هذه الدوال .
- دوال كلاس object :

GetType(), Equales(), GetHashCode(), ToString()

- كلاس ال parent يؤشر على كلاس child

```
{ //                تكملة البرنامج السابق
  int x;
  Object [] O2 = new Object [10]; //
  O2[0] = d;
  O2[1] = x;
  //                f  يافرض أن لدينا دالة
  f(object x);
  f(d);
  f(label1);
```

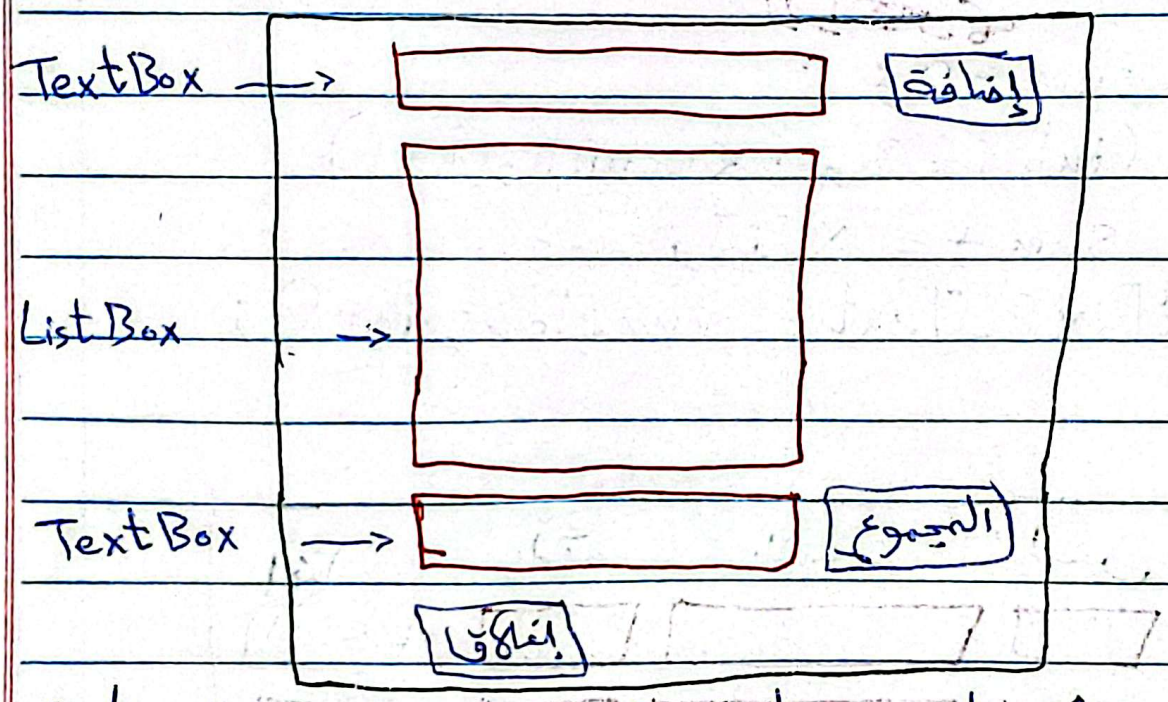
مصفوفة من نوع object تستطيع جعلها تؤشر على أي نوع . أنواع مختلفة .

الواجب : .....

كذلك بالنسبة للأدوات الأخرى

```
MessageBox.Show(o.ToString());
if (o is label) //للتحقق من أن هو لو أنا عارف أنه label
    ((label)o).Text = "AA"; // Text تحول لأن ال
}                           ليستا من أعضاء ال Object
```

لأن ال Object ليس لديه سوى دوال الظاهرة لدينا لكن في حالة ظهور دالة أو خاصية ليست موجودة في ال Object نقل له خده على ما تحتاجه أنت.

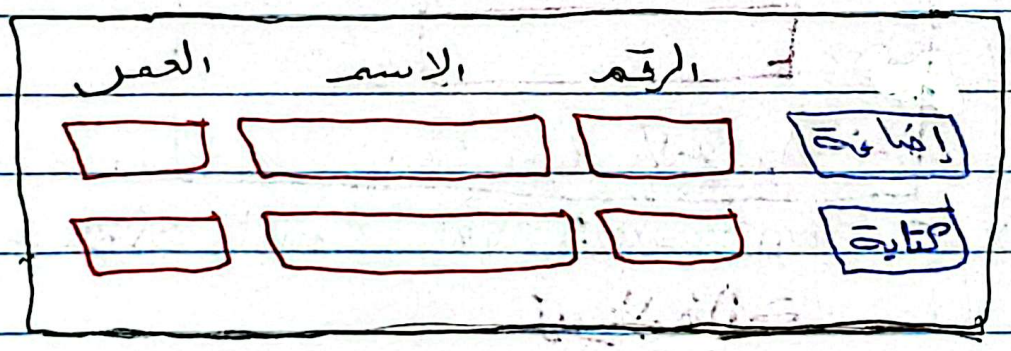



```
int x, count = 0; int const size = 100;
//                زر الإضافة
{ x = Convert.ToInt32(textBox1.Text);
  textBox1.Clear();
```

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
textBox1.Focuse();
int [] x = new int [size];
if (Count < size)
{ x[Count] = Convert.ToInt32(textBox1.
        Text.Trim()); }
    Count ++; }
}
//          زر المجموع
{ int sum = 0;
  for (int i = 0; i < count; i++)
        sum += x[i];
  textBox2.Text = sum.ToString();
}
```




```
struct person
{ int Code;
```

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
string name;
int age; }
 // Global
person p = new person ();
//  Click()  الحدث
{ P.name = textBox2.Text;
  P.Code = Convert.ToInt32(textBox1.Text);
  P.age = Convert.ToInt32(textBox3.Text);
}
// في المثال السابق في حال P كانت عبارة عن مصفوفة
{ if (count < size)
  { P[count] = new person ();
      :
      :
      }
}
```

باعتبار أن count و size و P مصفوفة معرفة Global

عند عمل مصفوفة من نوع كلاس أو struct ، عند التعريف نعمل [ ] ~ new هنا تعتبر أننا عرفنا مؤشرات فقط ، وعند الاستخدام يجب عمل new لها حتى تنشأ له موقع في ذاكرة الـ data الأساسية مثل int, float ~ إلخ.

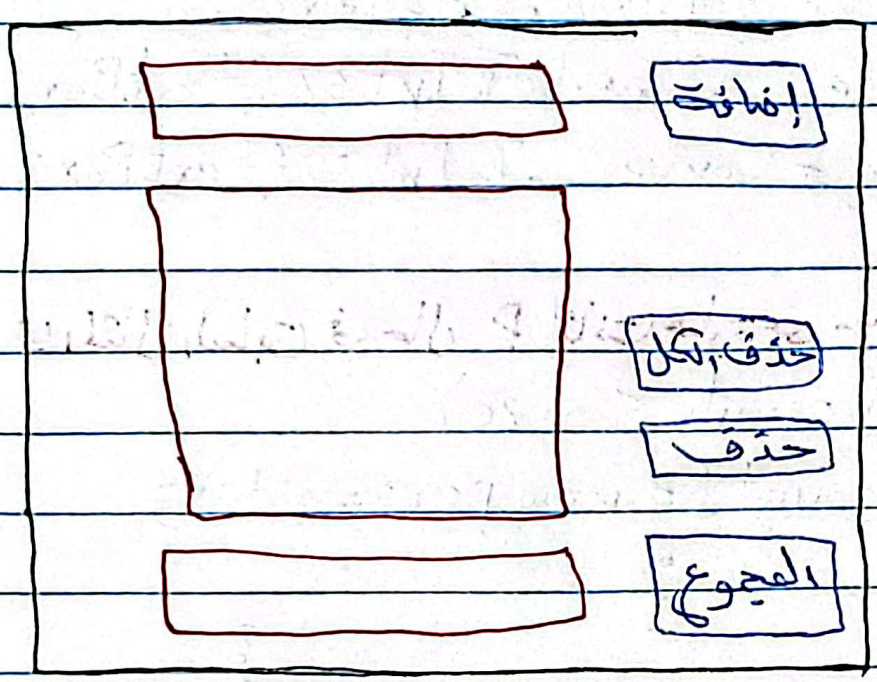
الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

ListBox1.Items ← عبارة عن مصفوفة

- Count عدد العناصر
- add إضافة عنصر
- Remove حذف عنصر حسب القيمة
- RemoveAt حذف عنصر حسب الموقع




```
//                    // الحذف
{ if (ListBox1.SelectedIndex != -1)//
    { ListBox1.Items.Remove
                (ListBox1.SelectedItem);
      // OR
      RemoveAt (ListBox1.SelectedIndex);
}
```

SelectedItems != null

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
//                زر الإضافة
{ if (textBox1.Text.Trim() != "")
    { ListBox1.Items.Add (textBox1.Text.Trim());
      textBox 1.Clear();
      textBox1.Focuse ();  }
}

//                زر حذف الكل
{ ListBox1.Clear();
}

//                زر المجموع
{ int sum = 0;
  for (int i=0 ; i < count ; i++)
   { sum += Convert.ToInt32 (ListBox1.Items[i]);
   }
  textBox2.Text = sum.ToString ();
}
```

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

- Enable - disable :
يجعل الزر يظهر لكن بلون خافت ولا نستطيع استخدامه في حال لم يتحقق الشرط.

- visable - unvisable :
يجعل الزر مخفياً طالما أن الشرط غير محقق.

- احتياطات برمجية واحتياطات مرئية :
احتياطات برمجية نكتبها في بينما المرئية تأتي مع الـ visual.

- الفرق بين panel و GroupBox :
GroupBox يمتلك عنوان Text وحواف
بينما panel لا يمتلك عنوان ولا حواف.
panel و GroupBox كأنه Form داخل Form.

الواجب : .....

موضوع الدرس: المحاضرة 6 اليوم: السبت

التاريخ: ٢٨/ ٩/ ٢٠٢٤م الموافق: ٢٥/ ٣/ ١٤٤٦هـ

- لكي تتعلم كيف تبني الأدوات يجب أن تتعلم كيفية تقن التعامل معها أولاً.
- Background Image خلفية الـ Form أو picture
- من خصائص الـ Image :
- Tile : تكرار الصورة غير مساحة عنصر التحكم.
- Center : تمركز الصورة.
- Zoom : ضبط الصورة بحيث تتغير أبعادها تلقائياً لتناسب حجم عنصر التحكم.
- Stretch : يسمح للصورة أن تغير أبعادها لتناسب مع حجم عنصر التحكم بغض النظر عن نسبة العرض والارتفاع الأصلي للصورة.

// تكبير الـ Form //

{ this.WindowState = FormWindowState.Maximized ; }

- عند تفعيل اختصار alt + let تضيف @ أمام الحرف بالأداة تفسيرها.
- يجب أن يكون الـ Form قابل أن يتلائم على قدر الأدوات stretch
- AutoSize يسمح لك بالتمدد والانكماش.
- AutoScroll عندما يتجاوز محتوى Form أو عنصر التحكم حجم المنطقة المرئية حيث يتم عرض أشرطة التمرير Scrollbars تلقائياً.

الواجب:

PictureBox :-

فيها خاصية Image وكذلك لها ميزة الرسم
وفيها Background Image كذلك .
نستطيع استخدام الـ Panel أو GroupBox لكنهما
يمتلكان Background Image ولا يمتلكان Image .
في حالة Background Image فيه stretch و Center و ...
بينما الـ Image ليس مثلها . الصورة تظهر كما هي
normal وهذا هو الفرق بينهما .
Bitmap هي من مهمته فتح image من الملف وتكون
مصفوفة ذات بعدين ← الصورة عبارة عن مصفوفة ذات
بعدين والألوان عبارة عن أرقام .
Background Image تحتاج كائن من نوع Bitmap ،

```
//                    زر العرض
{ string str = "C:\\ إمتداد \\ إمتداد
                                    bmp" ;
                    // = @"C:\ ___ \ ___ " ;
```

هذه نعملها بدل الايل \ ، حيث يأخذ كل فاصلة اثنتين ،

```
Bitmap temp = new Bitmap (str) ;
```

// نقدر نختصر اللي فوق بسطر واحد عن طريق الإحلال

```
// Bitmap temp = new Bitmap ( الامتداد ) ;
pictureBox1.BackgroundImage = temp ;
}
```

- عيوب الكود السابق يطلع الصورة مرة واحدة

// // زر العرض

```
{ try
  { string fname = textBox1.Text.Trim(); // ادخال المسار
    Bitmap temp = new Bitmap (fname);
    pictureBox1.Image = temp; }
  catch (Exception ex)
    { MessageBox.show (ex.Message); }
}
```

- عيوب الكود السابق هو أننا نكتب الإمتداد من تلقاء أنفسنا .
الحل نستعمل Browser
يرمجيته جاهزة لفتح الملفات . File OpenDialog

// // زر العرض

```
{ if (OpenFileDialog1.ShowDialog() ==
      DialogResult.OK)
  { string fname = OpenFileDialog1.FileName;
    Bitmap temp = new Bitmap (fname);
    pictureBox1.Image = temp; }
}
```

. Cancel أو OK إما ShowDialog دالة تعيد قيمة

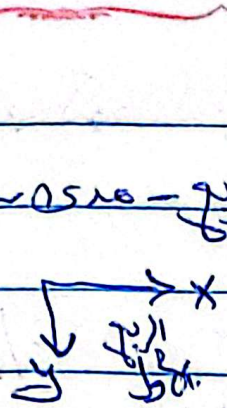
موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- PictureBox وسيلة الرسم في ال C#

نستطيع من خلالها رسم دائرة - خط مستقيم - مربع - مدرج - الخ.

- في الرسومات أربع إحداثيات مقلوب



الربع الأول مقلوب :

// الطريقة الأولى - // الرسم

```
{ Bitmap temp = new Bitmap (pictureBox1.
        width, pictureBox1.hight) ;
  int x , y ;
  for ( x = 0 ; x < 40 ; x++)
   { y = x ;
     temp.SetPixel(x, y, Color.Black) ; }
     PictureBox1.Image = temp ;
}
```

- مكتبة ال Graphics :

- عليها كتابة من أشكال الرسم موجود فيه.

- تكون object تؤشر على منطقة الرسم.

- DrawString() إحدى وسائل الطباعة على ال Form أي

هي ال cont ال Form.

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

// أسلوب آخر للرسم

```
{ Graphics p;
  p = this.CreateGraphics();  // منطقة الرسم داخل الـ Form
  p.DrawString("aaa", new Font(FontFamily.
        Families.ToArray(), ...));
  this.CreateGraphics().DrawImage(temp, 0, 0);
}
```

// التحريك

```
{ for (int j=0; j<10; j++)
  { for (int i=0; i<65; i++)
    { Button1.Left += 5;
      delay(100);
      Application.DoEvents(); } }
      System.Threading.Thread.Sleep(20);
      for (int i=0; i<65; i++)
      { Button1.Left -= 5;
        delay(100);
        System.Threading.Thread.Sleep(20);
        Application.DoEvents(); }
}
```

لا تهمل الأحداث الأخرى

الواجب : .....

Timer :-

مؤقت يشتغل بعد فترة زمنية محددة .

```
int i = 0;
int Counter = 0;
{ i++;
 Label1.Text = i.ToString();
 Counter++;
 if (counter >= 100)
      counter = 0;
 string str = "";
  if (counter < 10)
   { str = "0";
     Label1.Text = str + Counter.ToString();
}
```

// Form3 تشغيل //

```
{ Form3 F = new Form3; // كأننا عرفنا مؤشرين
  F.show(); // يختزن كيوبتر في مصفوفة ثانية
}
```

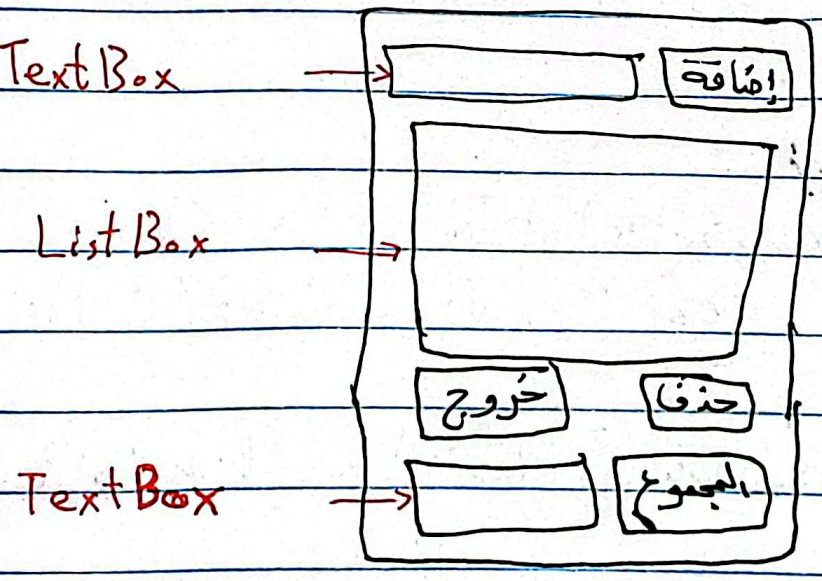
واحد يأخذها F والثاني يروح Global
بحيث يبدأ من المؤشر الأول .

#

موضوع الدرس: المحاضرة 7 اليوم: السبت

التاريخ: ٥/ ١٠/ ٢٠٢٤م الموافق: ٢/ ٤/ ١٤٤٦هـ

- التأثير على أكثر من عنصر لدينا أسلوبين
عن طريق التحديد الماوس simple
والطريقة الثانية تسمى extened
- دالة Random ركيزة أساسية في بعض مواضيع البيانات
الذكاء الاصطناعي Optimization ...
- القيمة الابتدائية التي تعطيها تعتمد على الساعة والزمن و
التاريخ ... الخ والـRandom
- اختيار أكثر من خيار Multi selection
- اختيار واحد فقط Selection mod




```
// Load
{ Random r = new Random ();
  for(int i=0; i<10; i++)
```

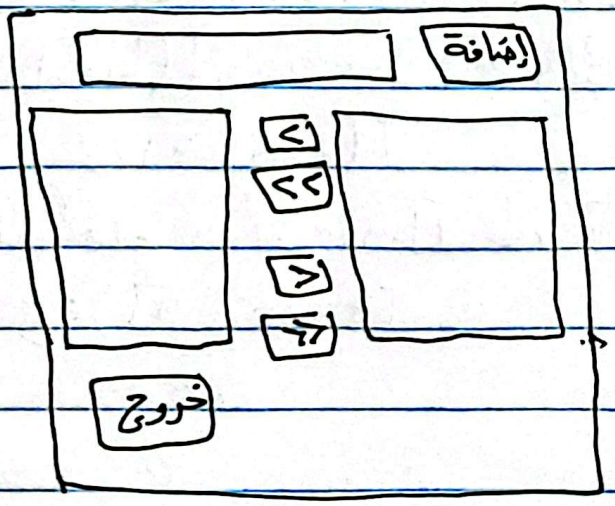
الواجب:

```
{int p = (int)(100 * (r.NextDouble()));
  ListBox1.Items.Add(p.ToString());}
}  int p = r.Next(100);   //  حل آخر
//  زر الإضافة
{ if (textBox1.Text.Trim() != "")
  { listBox1.Items.Add(textBox1.Text.Trim());
    textBox1.Clear();
    textBox1.Focus();}
  else
  { MessageBox.Show("Insert an item");}
}
//  زر الحذف
{ if (ListBox1.SelectedIndex != -1)
    lestBox1.Items.Remove(ListBox1.SelectedItem);
  else
    MessageBox.Show("Selected an Item to Remove");
}
//  زر الجمع
{ int sum = 0;
  for(int i=0; i< ListBox1.Items.Count; i++)
```

```
sum += Convert.ToInt32(ListBox1.Items[i]);
textBox2.Text = sum.ToString();
}
```

- selectedItem آخر عنصر تم اختياره
- selectedIndex الموقع المؤشر عليه بالاختيار
- selectedItems تحديد مجموعة عناصر List-array
  - Add, Count, Remove, RemoveAt
  - تحذف التطليل (تحذفه من قائمة المطلبين) ولا تحذف العناصر.

- في المستقبل تحتاط أن الزر لا يعمل إذا كان SelectedIndex يساوي -1، أو عبر SelectedItem إذا كان يساوي null «غير مؤشر»، أما إذا كان multiselection نفحص إذا كان count أكبر من الصفر.



موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
// الإضافة
{ if (textBox1.Text.Trim != "")
  { ListBox1.Items.Add(textBox1.Text.Trim());
    textBox1.Clear();
    textBox1.Focus(); }
  SetEnable();
}

// الزر >
{ // عنصر واحد مظلل
  ListBox2.Items.Add(ListBox1.SelectedItem);
  ListBox1.Items.Remove(ListBox1.SelectedItem);
  SetEnable();
  // كل العناصر المظللة
  while (ListBox1.SelectedItems.Count > 0)
  { ListBox2.Items.Add(ListBox1.Selected
                Items[0]);
    ListBox1.Items.Remove(ListBox1.Selected
                Items[0]); }
}
```

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
// << الزر
{ for (int i=0; i < ListBox1.Items.Count; ++i)
      ListBox2.Items.Add (ListBox1.Items[i]);
  ListBox1.Items.Clear();
  SetEnable();
}
private void SetEnable()
{ if (ListBox1.SelectedIndex == -1)
      button2.Enabled = false;
  else
      button2.Enabled = true;
  if (ListBox1.Items.Count == 0)
      button3.Enabled = false;
  else
      button3.Enabled = true;
}
```

عندما ترسل الملف ولديك بيانات كثيرة ، فكل ما نقلنا العناصر وكل عنصر يتم نقله يتم حذفه من الملف الأول ، قد تنقطع الكهرباء بالمنتصف ، لذلك يتم إرسالها للملف الآخر الى أن ينتهي يحذف ما بداخل الملف الأول حديثاً لانتهاء من النقل ».

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
//                    عرفنا form
{ if ((f == null) || (f.IsDisposed))
    { f = new Form10();
      f.show();
      // f.ShowDialoge(); }  لا يسمح للرجوع للسابق
  else
    { f.show();
      f.Focus(); }
  //          Global معرفة f
}
```

- في حال لدينا أزرار ونريد توظيفها لنفس الوظيفة :
إذا كتبنا نفس الكود لكل الأزرار ليس الحل الأمثل وذلك لأنه عندما يتحول البرنامج إلى exe يتضاعف الكود ويأخذ مساحة أكبر
إذا احتمل تعديل على أحد الأزرار لازم نعدل على البقية
- في هذه الحالة إما نستخدم مبدأ التضعيف ، أو نعمل دالة ونستدعيها لكل الأزرار ، أو نكتب الدالة داخل أحد الأزرار ومن ثم نستدعي دالة الحدث للزر لبقية الأزرار.

```
Button1_Click(sender, e);
```

الواجب : .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الواجب: / / ١٤هـ

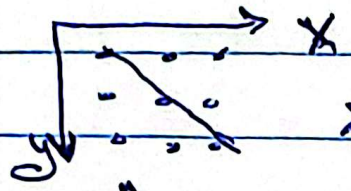
```
//          00 : 00
{ i++;
  if (i>10)
     i=0;
  Label1.Text = "Now" + i.ToString() + "Tick";
  // حل آخر
  if (i<10)
     S = "0";
  else
     S = "";
  Label1.Text = "Now" + S + "Tick";
}
//          نشاط الرسام
{ string filename;
  filename = "C:\A\" + filename
  ↑ مكان تخزين الصور
  Bitmap Bp = new Bitmap(filename);
  pictureBox1.Image = Bp;
}
```

الواجب : .....

موضوع الدرس: تكملة المحاضرة 7 اليوم: الأحد

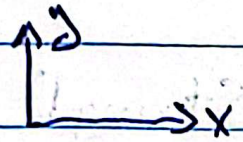
التاريخ: ٦/ ١٠/ ٢٠٢٤م الموافق: ٣/ ٤/ ١٤٤٦هـ

- لكي نرسم نعمل Bitmap في الذاكرة وهي عبارة عن مصفوفة ذات بعدين



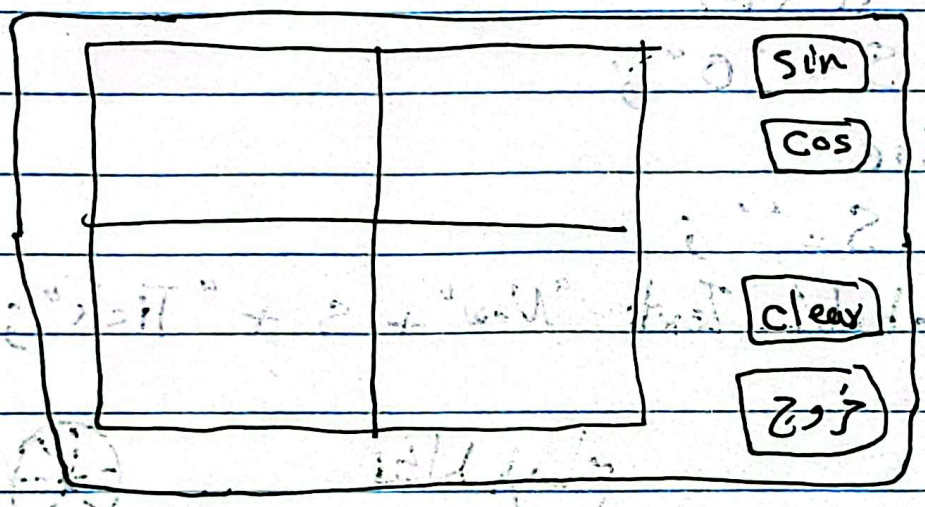
بالمقلوب

في حال أردنا عكس الأبعاد



x في حالة زيادة، y في حالة نقصان

- الأساس هو أن نجيب المنصف بالرسم




```
//        sin        //
{ Bitmap temp = new Bitmap (pictureBox1.Width,
                            pictureBox1.Height);
  int Dwidth = pictureBox1.Width -1,
      Dheight = pictureBox1.Height -1,
      x, y;
  for (x = -150; x < 150; x++)
```

الواجب:

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠ م الموافق : / / ١٤ هـ

```
{ y = (int)(Math.Sin(x*Math.PI/180)*60);
temp.SetPixel((Dwidth/2)+x, (Dheight/2)-y,
        Color.Yellow); }
pictureBox1.Image = temp;
pictureBox1.Refresh();
Graphics g = pictureBox1.CreateGraphics();
g.DrawLine(new Pen(Color.White),
        new Point(0, Dheight/2),
        new Point(Dwidth, Dheight/2));
g.DrawLine(new Pen(Color.White),
        new Point(Dwidth/2, 0),
        new Point(Dwidth/2, Dheight));
}
```

- Create Graphics تؤشر على منطقة الرسم
حيث تسمح لك بانتاج object فيه امكانيات الرسم.
- بالتشكيل tan يحتوي على احتياطات بمنطقة السالب
عند الرسم.

الواجب : .....

موضوع الدرس: المحاضرة 8 — اليوم: السبت

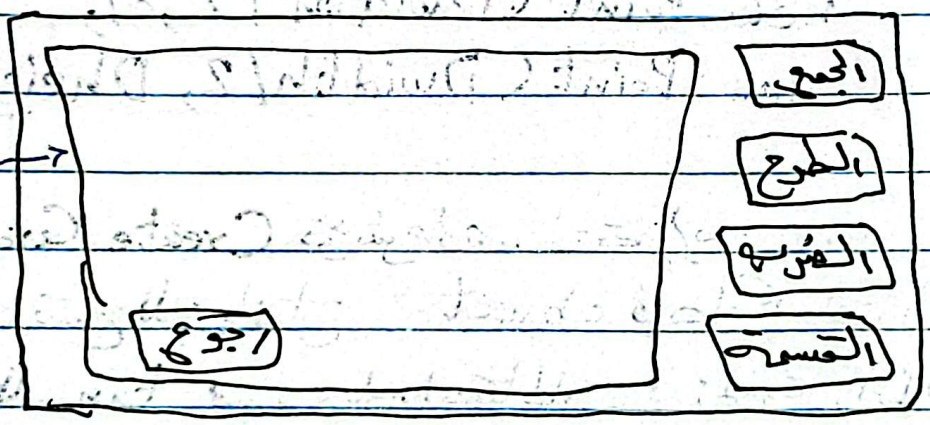
التاريخ: ١٢/١٠/٢٠٢٤م — الموافق: ٩/٤/١٤٤٦هـ

- الـ Form ليس شرطاً أن نشكله بالماوس، يمكننا إنزال الأدوات إليه بشكل عشوائي لكن عند التنفيذ يعرض الـ Form المطلوب.
- نشكل الـ Form برمجياً عن طريق Location ونعمله بالـ Load.

```
Load ()
{ textBox1.Location = new point (10,10);
  textBox2.Location = new point (70,10);
}
```

الـ معناه x,y ، كم مسافة من x وكم مسافة من y

- رؤية الـ Form والمواقع والأهداف، أنت تسحبها فقط وتشكلها برمجياً ← في البرمجة المتقدمة سوف نتعلم هذا الأسلوب، حتى نسحب الأدوات يتم تعريفها برمجياً.




```
// Load                    // Multi Task Form
{ panel1.location = new point (1,1);
  :
  panel1.Visible = false;
  :
```

وهكذا بالنسبة للباقي

الواجب:

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

من أجل المسافات

```
panel5.Location = new point ((panel1.width +10), 1);
this.size = new size (panel1.width, (panel5.width +10), (panel1.height +10));
}
// الجمع //
{ panel1.Visible = True;
  panel2.Visible = false;
  :
  SetEnable (false);
}
public void SetEnable (bool T)
{ Button1.Enable = T; // عند الضغط على أحد الأزرار الكل يلتغي
  :
}                     // عند الضغط على رجوع الكل يتفعل //
```

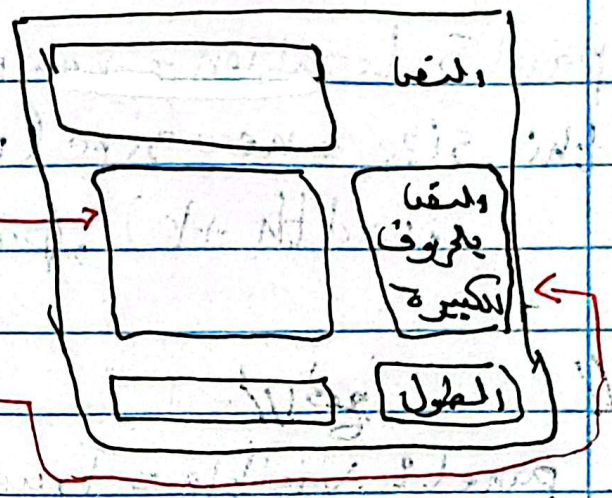
لـ في هذه الحالة نحن وظفنا الأدوات والخصائص حيث قمنا بخاصية الـ Enable للاحتياط.

في الـ Form السابقة أنزلنا الأدوات وعند التنفيذ يطلع بشكل آخر

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ



MultiLine

خاصيتين للتحكم بالحجم

autoSize

```
string Temp = textBox1.Text.Trim();
// التحويل الى حروف كبيرة
{ textBox2.Text = Temp.Upper(); }
// الطول
{ textBox3.Text = Temp.Length.ToString(); }
// تعريف مصفوفة نصية
{//string [] ar = new string [10]; تعريف
// Temp.split ('+'); تقسيم (Token) حسب +
string [] ar = Temp.Split('.');
for (int i=0; i < ar.Length ; i++)
  { ListBox1.Item.Add(ar[i]); }
    textBox4.Text = ar.Length.ToString();
ListBox1.Items.Clear();
char [] aA = Temp.ToCharArray();
```

مصفوفة تحويل الى حروف نص

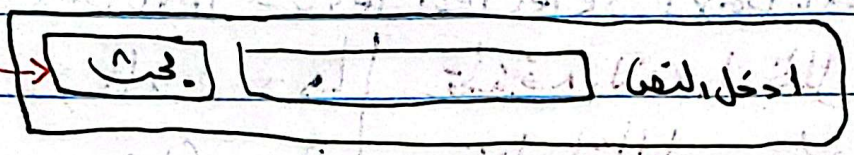
الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
for (int i=0; i< aA.Length; i++)
    ListBox1.Items.Add (aA[i]);
}
```

زر مستقل عن المعالجة




```
{ string Temp = textBox1.Text.Trim();
  string index = TextBox5.Text.Trim();
  int index = Temp.IndexOff (Target); //
```

يبحث داخل النص عن هذا المقطع و ترجع الـ index تبعه

إذا ما لقيه يرجع -1

```
  if (index == -1)
  { MessageBox.Show ("Not Found !"); }
  else
  { textBox1.Select (index, Target.Length); //
```

يظلل الموقع الذي أشر عليه وعلى عدد الأحرف

```
    textBox.Focus (); } //
```

ينقل التركيز لعنده

//

طريقة ثانية لتظليل النص

```
textBox1.SelectionStar = index; //
```

تظليل عن طريق الخصائص

وهو موقع المؤشر سواء لـ التظليل أو لا .

الواجب : .....

// طريقة أخرى

textBox1.SelectionLength = target.Length;

// : خاصية أخرى تأخذ وتعطي

textBox2.Text = textBox1.SelectedText;

textBox1.SelectedText = TextBox2.Text;

تحذف النص المظلل ← | = null;

تزيل التظليل من على النص ← | = 0;

يبحث عن العنصر

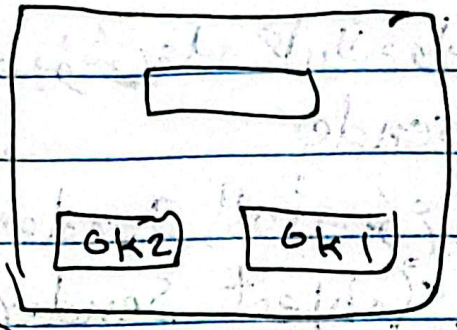
تستبدل العنصر المظلل بالقيمة

}

في حال أردنا معرفة مكان المؤشر في حال أن هناك نص مظلل عليه، التظليل هو يدل المؤشر أي أنه بداية التأشير.

موضوع الدرس: تكملة المحاضرة 8 اليوم: الأحد

التاريخ: ١٣/١٠/٢٠٢٤م الموافق: ١٠/٤/١٤٤٦هـ



مؤشر على الهدف الذي طلب الحدث في نفس اللحظة

```
Button_Click (Object sender, EventArg e)
{ textBox1.Text = null;
    if (sender == Button1)
        MessageBox.show("OK1");
    else if (Sender == Button2)
        MessageBox.show("OK2");
    if (sender is Button)
        ((Button)Sender).Text = "AA";
    OpenFileDialog OpenFileDialoge1 =
        new OpenFileDialoge();
    OpenFileDialoge1.dispose();
}
```

المساواة هنا مساواة عناوين

الغرض هو أن الزرارين مربوطين بنفس الحدث

هل هو من نوع Button

يعرف في هذه اللحظة Local

هي أساساً Form

بعد ما نخلص عمله //

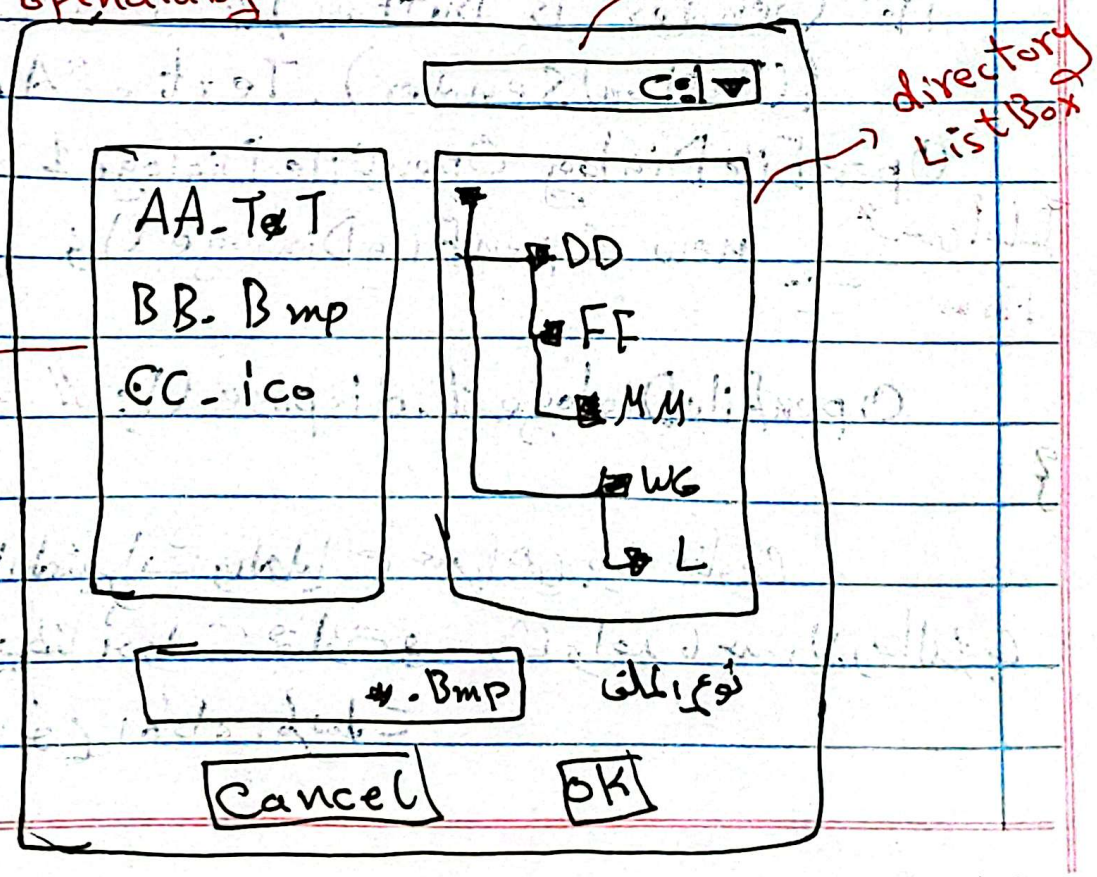
* C# كل المتغيرات والعمليات عليها هي بين عناوين

في حال ربطنا أدوات مختلفة في حدث واحد ما عند الإرسال يجب التحقق من نوع الأداة المرسلة

- Control كلاس يؤثر على كل الأدوات
وبالتالي إذا قلنا (Control) sender
فهو يحول الـ Sender إلى أي نوع.

```
Button2_Click (object Sender, EvenArg e)
{ Button1_Click (sender, e); }
Click ()
{ textBox1.Text += ((Button) sender).Text;
```

لـ تحويل نوع ← خذه على هيئة Button ←
لأنه من نوع object وليس فيه خاصية Text
← تعرين على الأهداف



موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- DriveList Box أداة تظهر الأقراص
- Directory List Box الأدلة الحالية للقرص
- FileList Box الملفات الواقف عليها في تلك المنطقة

- لإظهار هذه الأدوات من قائمة الأدوات + الزر الأيمن بالماوس في منطقة فارغة + تطلع قائمة فيها Add نختار يطلع مربع فيه أدوات ابحث عن اسم الأداة.

```
public string filename; // Global
Load ()
{ ComboBox1.Items.add ("*.*");
  ~ ~ ~ ("*.Bmp");
  ~ ~ ~ ("*.ico");
  ComboBox1.SelectedIndex = 1;
}

drive List Box_Selected Index Changed ()
{ div List Box1.path = drive List Box1.Drive;
  file List Box1.path = drive List Box1.path;
}

div List Box1_selected Index Change ()
{ file List Box1.path = dri List Box1.path;
}
```

هناك أدوات مختلفة تتعامل مع عدة ملفات مثل pdf

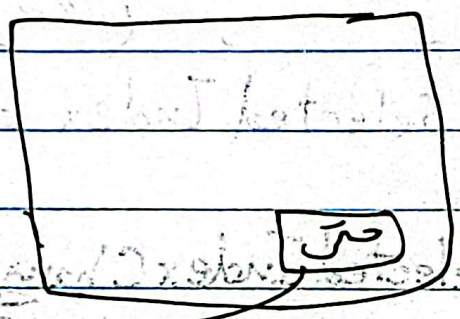
الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠ م الموافق : / / ١٤ هـ

```
ComboBox1_SelectedIndexChange ()
{ fileListBox1.pattern = CombBox1.Text.Trim();
}

  //    OK    //
{ filename = fileListBox1.path + "\\" + fileListBox1.
                filename;
  This.close();}
//    Cancel    //
{ filename = null;
  this.close();}
```



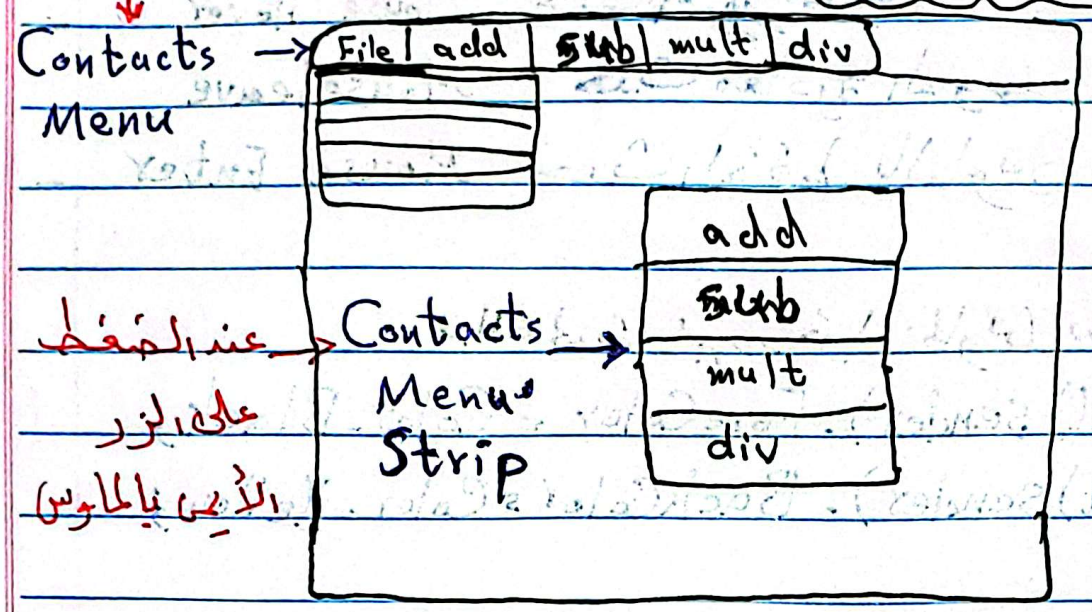

```
click()
{ Form1 f = new Form1();
  f.showdialog();
  if (f.filename != null)
   {string fname = f.filename);
      BitmapTemp = newBitmap (fname);
}
```

الواجب : .....

موضوع الدرس: المحاضرة ٢ ..... اليوم: السبت

التاريخ: ١٩/١٠/٢٠٢٤م ..... الموافق: ١٦/٤/١٤٤٦هـ

القوائم Menu وتوظيفها



- من قائمة الأدوات نضغط الزر الأيمن ونختار Choose Item + Dix List Box
- القائمة يمكنها أن تحتوي على أكثر من واحد.
- في حال كان لدينا أكثر من قائمة نذهب إلى الخصائص مثال ما هية تسمى Contacts Menu Strip وقد من يعمل أولاً بحيث تستطيع تنزل أكثر من واحد وتستطيع تبدل بينهم داخل البرنامج أو بالكود البرمجي
- اسم القائمة = This.Contacts Menu Strip
- بعدما تنزل القائمة تكتب لكل منهم وقا الكود كي يؤدي عمله.
- القوائم لها اختصارات كثيرة ولها اختصارات وسائل.
- القوائم تعتبر أزرة لكن عند الطلب «أزرة منبثقة»

الواجب: .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

- Mouse Hover حدث المرور للماوس.
- Mouseleave حدث مغادرة الماوس.
- Mouse Enter حدث الضغط بالماوس.

// دالة لحدث بمجرد دخول الماوس تعمل

```
{ ((label) sender).ForeColor = Color.Black;
  (label) sender).BackColor = Color.Yellow;
}
```

- عند عمل هذا الحدث لنفس الأداء لعدة أدوات نجعلها في دالة الكود نفس الكود فقط يتغير ال sender وهذا نوع من أنواع ال polymorphism

```
{ ((control) sender).ForeColor = Color.Black;
  ((control) sender).BackColor = Color.Red;
}
```

- الأصل في الموضوع أن كل أداة لديها خاصيتها.
- الأدوات كلها مشتقة من Control وال Control مشتق من ال Object.
- ال Control مؤشر يؤشر على أي أداة من الأدوات لأن كل الأدوات مشتقة منه.
- transparent خاصية الشفافية للون.

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- تحويل الـ Form إلى Mdi Parent في حال لديك عدة Forms .

```
{ Form11 f = new Form11();
  f.MdiParent = this;
  f.Show();
}
```

نختصرها

```
f.Show(this);
```

لـ أو بالـ Load ⇐ this.MdiParent = true;

- الاحتياطات :

تأكد من الـ TextBox يدخل بيانات صحيحة ⇐ عملية التحويل تكون صحيحة ⇐ بحيث : لا يكون فارغاً أن يكون المدخل رقم أو نعمل للأزرة بحيث لا يضغط الزر أساساً .

- احتياط يمنع إدخال الأحرف بحيث

إذا الحرف المدخل من ضمن الأرقام يمشي ⇐ يقبله ⇐ .

" " " " " الحروف لا يمشي .

مع العلم أن C# كلها تحتوي

- الحدث Keypress عند نقرة الحرف .

- sender هو الذي يستدعي الحدث .

- e بيانات الحدث نفسه .

- البيانات المرتبطة بالحدث مثل نقرة الزر Keypress .

الواجب : .....

- الحدث Keypress يعمل قبل أن يصل الحرف إلى TextBox
- الحدث Text_Change يعمل عندما يصل الحرف إلى TextBox.

- تغيير Form من داخل Form آخر:
- باستخدام خاصية Modifier الموجودة في قائمة الخصائص والتي تسمح بتغيير الـ Form من Private إلى Public بحيث تستطيع تغيير أي شيء من Form آخر.
- خاصية Modifier خاصية لا يمكن أن تغيرها برمجياً هي فقط لل editor.

e.KeyChar = '\0'; // تهيئة علة
الحرف الذي تم ضغطه
e.Handled = true; // عطلت علة
انتهى معاملته ويرجع قيمة صفر.

bool f = int.tryParse(textBox1.Text.Trim(), out x);
تعيد True أو False أن التحويل تم أم لا دا لي دالة تتبع الكلاس int
- دالة Convert تحدث مشاكل في بعض الأحيان.
تحول وتسند لهذا المتغير فهي لا تسند قيمة عن طريق اسم الدالة وإنما عن طريق متغير out.
out هنا تأتي بدل ref. بحيث أن هذا المتغير يعيد قيمة
in لا تتغير هي للإدخال فقط.

موضوع الدرس : تكملة المحاضرة ٢ اليوم : الأحد

التاريخ : ٢٠/١/٢٠٢٤م الموافق : ١٧/٤/١٤٤٦هـ

- أنواع البيانات :

int ← int 32

Long ← int 64

short ← int 16

float

Double

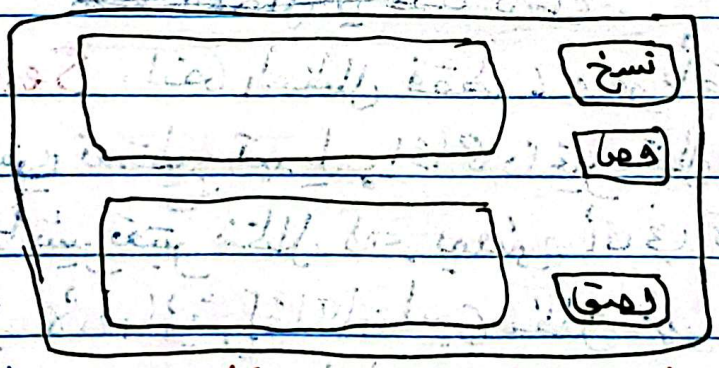
Decimal ← بعد الـ double من حيث السعة والدقة

Byte

char

string ← نوع بيانات primitive

DateTime



string temp; // متغير نصي عام لنسند له القيمة

// زر النسخ

{ temp = textBox1.SelectedText;

//Clipboard.SetText(textBox1.SelectedText);

}

الواجب :

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠ م الموافق : / / ١٤ هـ

```
// القص //
{ temp = textBox1.SelectedText;
  textBox1.SelectedText = null;
  // Clipboard.SetText (textBox1.SelectedText);
  // Clipboard.SetText = null;
}

// اللصق //
{ textBox2.Text = temp
  textBox2.SelectedText = temp; //
  textBox2.SelectedText = Clipboard.GetText();
}
```

Temp في منطقة التظليل حط

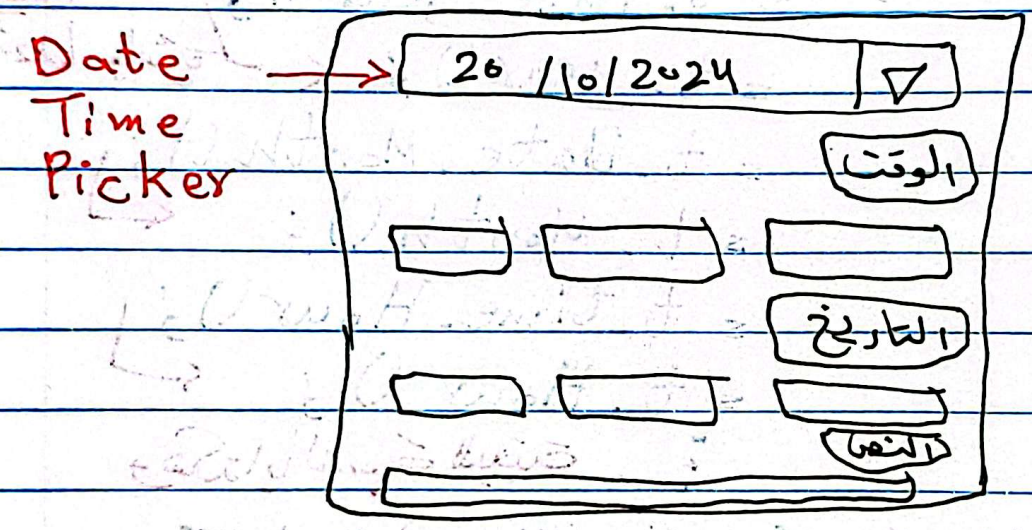
- Text تأخذ النص كامل.
- SelectedText النص المظلل فقط، وهو يأخذ ويعطي، يعني تستطيع تبديله إذا كان غير مظلل، منطقة التأشير تعتبر مظلل له يعني أنه في كل الحالات في تظليل لأن المؤشر إذا كان ليس مظلل فهو يظلل على منطقة مظلل والطول صفر Zero.
- نظام التشغيل لديه منطقة لتخزين النسخ واللصق، نصل له عن طريق كائن أو كلاس يمتلك دالة توصل له وتحفظ فيها القيمة ودالة تسحب منها القيمة كأنه temp.

الواجب : .....

وهذه المكتبة هي Clipboard . وهي تأخذ بايتات

- لماذا نستخدم حافظة نظام التشغيل ؟!
وذلك لو أردنا النسخ من برنامج إلى برنامج آخر.

- كود اللصق هو نفسه كود النسخ الفرق فقط أنه في اللصق نجعل النص المظلل يساوي null .




```
DateTime t = DateTime.Now ; // دالة تعيد الوقت الحالي
textBox1.Text = t.ToString () ;
DatTime.Value = DateTime.Now ; // نستخدمها مباشرة
```

لـ على هيئة خاصية وليس على هيئة دالة.

```
textBox1.Text = DateTime.Now.ToString () ;
```

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
DateTime t;
t = DateTime.Now;
TextBox2.Text = t.Time().ToString();
```

الوقت فقط

```
TextBox3.Text = t.Date.ToString();
              = t.Date.year();
              = t.year();
              = t.Date.Month();
              = t.Month();
              = t.time.Hour();
              = t.Hour();
```

يمكن نعملها مباشرة

وهكذا بالنسبة للبقية ...

- DateTime أداة تختص بالـ Time تطلع التاريخ لنفس اليوم أوتوماتيكياً مع إمكانية التغيير.
- التاريخ إما تكتبه كتابة أو تحدده يا ملوس.
- التاريخ له عدة صيغ ، معتمدة على الوقت سنة الخ ..
- ليس شرطاً استخدام الأداة هنالك أمر تستخدمه مباشرة اسمه Now يجيب التاريخ كاملاً على هيئة object من نوع DateTime
- التحدي يكمن في عمل العمليات الحسابية بالنسبة للزمان فارق الأيام ، حدود معينة - الخ ..

الواجب : .....

موضوع الدرس: المحاضرة ١٥ — اليوم: السبت

التاريخ: ٢٦/١٠/٢٠٢٤م — الموافق: ٢٣/٤/١٤٤٦هـ

```
void TT (int (*p)(int,int))
{ cout << p(10,20) ; }
int add (int a, int b)
{ return a+b ; }
void main ()
{ int *p, x = 20;
  int (*pf) (int ,int);
  pf = add //
  p = & x;
  int z = pf (10,20); // delegate
  z = add (1,2);
  TT (pf);
}
```

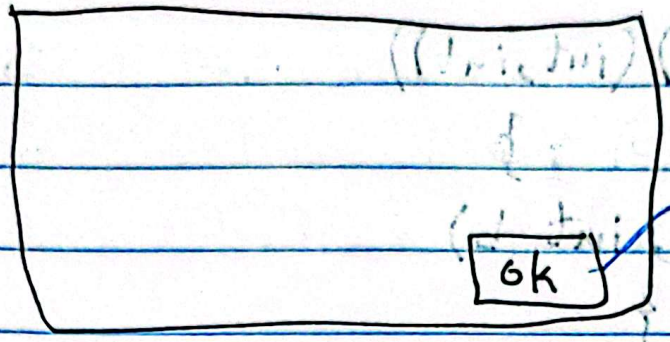
ما تعمل & مثل المتغير العادي

- pf في المثال السابق يعتبر delegate حيث يعتبر مؤشر دالة يؤشر على دالة معينة.
- الأحداث في C# عبارة عن delegate عندما يحصل حدث معين ينفذ الدالة.

الواجب:

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠ م الموافق: / / ١٤ هـ




```
void b1_click
(object sender,
EventArg e)
{ This.close(); }
```

تنبيه من الصفر

project → add class → myForm

```
using System.Windows.Forms;
class myForm : Form
{ private Button okButton;
  public myForm ()
  {okButton = new Button();
    okButton.Text = "OK";
    okButton.Location = new point (200,300);
    this.Controls.Add(okButton); //
    okButton.Name = "Button1"; //
    this.Text = "Example"; //
    Button1.click += b1_click; //
  }
}
```

هنيئه للفورم بهيئة مرئية.

اسم للأداة

الفورم نفسه

إضافة الحدث click

الواجب: .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- لكي نبني الفورم من الصفر :

نعرف الأهداف كلها، نحتاج لكل هدف new، نفعل خصائص لكل هدف، أهم الخصائص هي Size, Location، وترجع نقل له Add وترجع تعد خصائص الفورم وهذا كله داخل الـ Constructor أو في Load.

إضافة الأحداث يمكن أن يكون في Load.

- لو أردنا فورم جمع رقمين :

نعرف 3 TextBox, 3 label, 2Button

وبعدين نقل له new ونعطي خصائص كل واحد، تربيط الأحداث حقك وأنت حافظ الهيدر حق الأحداث.

لما بدي تربيط الأحداث تكتب فقط اسم الدالة.

- دالة intilize Compannende () :

هي الدالة التي تنشأ الأهداف وتعطي له الخصائص اللي أنت تسحبها يحولها إلى أهداف.

أنت تسحب هدف يعرفه ويفعل له new.

Controls ⇐ عبارة عن مصفوفة فيها بوينترات.

- إذا أردنا نعرف أدوات لداخل panel :

This.Panel.Controls ();

This.Controls.Add(panel);

- عشان فورم نحبي نفتح عن طريق الزر الأيمن ← view code

الواجب : .....

```
void Start ()
{ textBox1 = new textBox ();
  textBox2 = new textBox ();
  label1 = new label ();
  label1.Location = new Point (10,10);
  label1.Size = new Size (100,10);
  label1.text = "العدد الأول";
  textBox1.Click += button1_Click;
  Controls.Add (Button1);
  :
}
```

- item[] هذه تعطي ولا تأخذ أي لا تستطيع تعطيها يساوي نهائياً
null = item[0] هذا خطأ تعطيها فقط Clear, Remove, Add.

- حذف البيانات من داخل TextBox :-
دوران وباستخدام مصفوفة المؤشرات Controls احذف Text

```
for (int i=0 ; i< this.Controls.Count ; i++)
    this.Controls[i] ;
```

- لكن هذا يمسح في كل الأدوات، إذا أردنا أداة معينة مثل TextBox نضيف الشرط

```
if (Controls[i] is TextBox)
    this.Controls[i].Text = "";
```

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

- هذه المصفوفة تفيدنا في أشياء كثيرة حيث لو أردنا مثلاً تطبيق حدث معين على كل الأدوات أو أردنا مثلاً الماكد أن جميع Text Box فاضية أو سه إلخ.
- Object يؤثر على كل الكائنات
- Controls يؤثر على الأدوات الموجودة على الفورم.
- وهذا هو Polymorphism ← تعدد الأشكال ← وهو أن مصفوفة واحدة تؤثر على Text Box ومرة على label ومرة على Button أي دوران واحد يؤدي أغراض متعددة
- وهذا هو ال Generic نتيجة ال polymorphism.
- Generic : ميزة موجودة في لغات البرمجة مثل C++ تتيح لك إنشاء كود مرن وقابل للإعادة الاستخدام من خلال العمل مع أنواع بيانات مختلفة دون الحاجة إلى إعادة كتابة نفس الكود لأنواع متعددة ← تركز على النوع ←.
- Polymorphism : مفهوم يسمح للكائنات المختلفة بأن تعامل بنفس الطريقة من خلال واجهة موحدة مثل الوريث والواجهات. ← تركز على السلوك ←.
- ال Overloading يعتبر أيضاً Generic لأنه يتعامل مع نوع آخر من ال Generic وهو أن نعرف مصفوفة من نوع Template ← يؤدي إلى Generic ← عمومية المصفوفة ←.
- ال polymorphism معناه أن الهدف ممكن يؤشر على نوع مختلف من نوعه

الواجب : .....

- حلقة الدوران foreach :

* لكل ء في حال المتابعات كالمصفوفات.
foreach لكل قيمة في المصفوفة.

```
foreach (Control c in this.Controls)
    if (Controls[i] is TextBox)
        c.Text = "";
```

c تستبدل بكل دوران

```
foreach (Control c int this.Controls)
    if (c is TextBox)
        Controls[i].Text = "";
foreach (TextBox c in this.Controls.ofType
                <TextBox>)
    c.Text = "";
```

تعطي مصفوفة خاصة من النوع المرسل لها.

في هذا الدوران سوف تعطي مصفوفة من نوع TextBox الموجود بالـ Control. * تشتق الـ TextBox لحالها ء.

ففي الدوران السابق ما احتاج if لأنه استخلص الـ TextBox فقط.

This.active الفورم الفعال في نفس اللحظة.

This.active.mdichildren في أي منطقة داخل الكود معناه آخر واحد أنت موجود عليه.

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠ م الموافق : / / ١٤ هـ

- Close all
- Tool bar
- Tool Strip
- Save dialoge
- Open dialoge
- Arang Icons
- Envirome nt
- Special Folder
- Personal
- medilayout
- title vertical
- title horizental
- casecade
- Task bar
- layout

- Application ← الكلاس التي تشرف على المشروع كامل.
- OpenForm خاصية تعرف كم عدد الفورمات المفتوحة. هي مصفوفة مثل الـ Controls واخلها مؤشرات لكل النماذج المفتوحة « حق المشروع كامل ».
- Controls عبارة عن مصفوفة تحط فيها مؤشرات للأهداف التي داخل الفورم الحالي.
- الفورم يتبعه خاصية على هيئة مصفوفة اسمها Controls تأخذ عناوين كل الأهداف، إذا كان على شكل meli تضاف خاصية أخرى على هيئة مصفوفة اسمها melichiled وتحط فيها عناوين كل النماذج، ممكن نعمل دوران عليهم ونغلقهم أو نغير الـ Text أو الأبعاد ... إلخ.
- أثناء الحذف كيف تطلع رسالة هل تريد الحذف أم لا ؟!

MessageBox.show ("", "", MessageBoxButton.OkCancel);

وعند الضغط على Yes أو No Show Diabage

الواجب : .....

موضوع الدرس : تكملة المحاضرة 10 اليوم : الأحد

التاريخ : ٢٧/١٠/٢٠٢٤م الموافق: ٢٤/٤/١٤٤٦هـ

```
{ Button B1 = new Button();
  B1.Text = "Exit";
  B1.Click += الدالة (sender, e) => { }
  this.Controls.Add(B1);
}
// Exit
{ Application.Exit();}
class Program
{ public static void ClearForm (Form F)
  { foreach (Control c in F.Controles)
        c.Text = "";  }
}
int xTop = 50;
TextBox[] T = new TextBox[100];
int count = 0;          // Global
{ if (count < 100)
  { TextBox T[count] = new TextBox();
    T[count].Text = "";
    // r.nextdouble() * 1000;
    T[count].Top += 30 + Count;
```

الواجب :

موضوع الدرس: اليوم جمعة

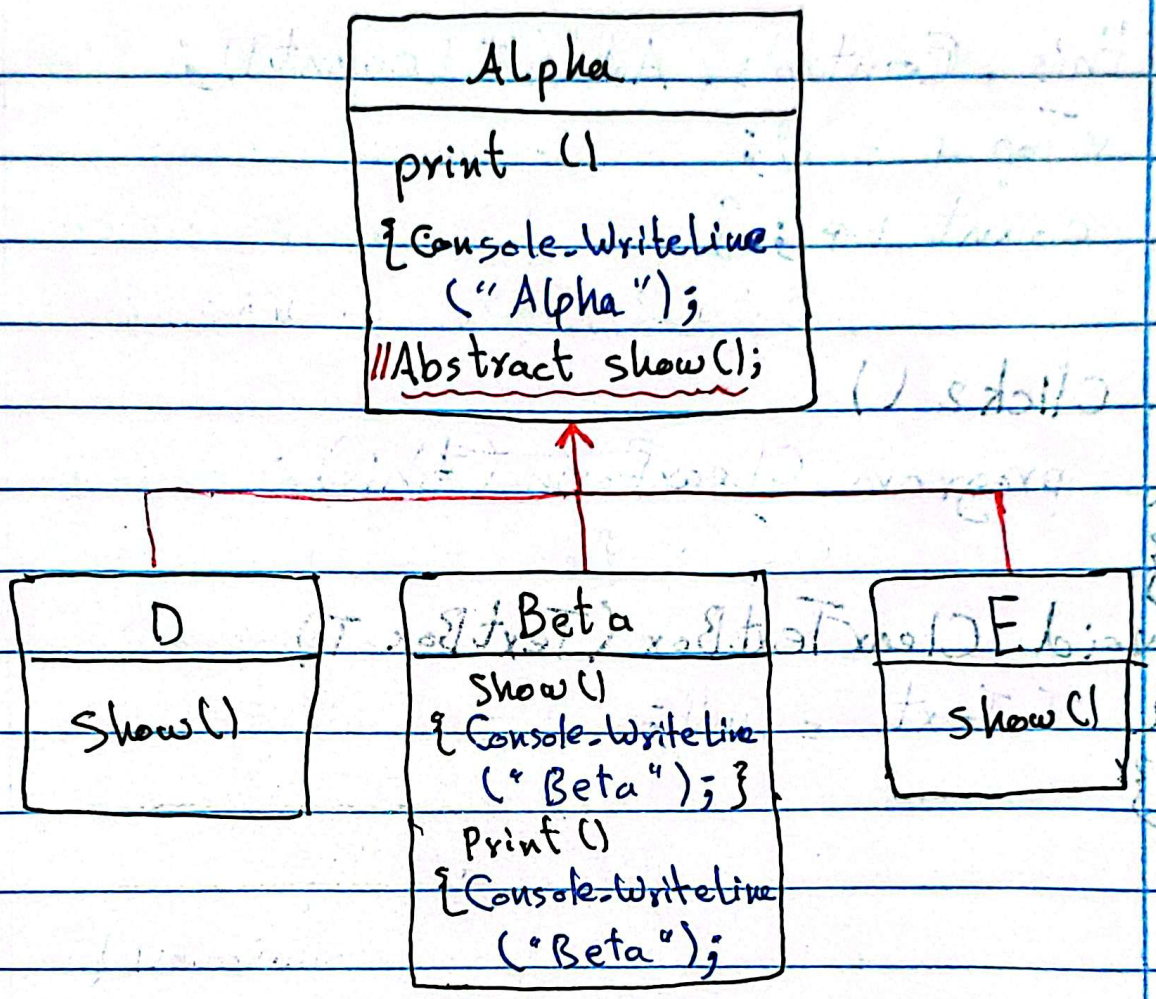
التاريخ: ١٠/٢/٢٠٢٠م

```
this.Controls.Add(T[count]);
xTop += 39;
count ++; }
}
Click2 ()
{ program.ClearForm(this);
}
void ClearTextBox(TextBox T)
{ T.Text = null;
}
```

الواجب:

موضوع الدرس: المحاضرة 11 اليوم: السبت

التاريخ: ٢/١٢/٢٠٢٤م الموافق: ٣٠/٤/١٤٤٦هـ




```
{ Beta b = new Beta();
  Alpha a = b;   // الأصل بالموضوع أنه مؤشر أي أنه لم يحل
                 // بقواعد C العادية، بحيث أن C تأخذ مؤشر.
  a.show();      // في قوانين C++ لا يمكن إلا إذا كانت موجودة
                 بكلاس Alpha بحيث يأخذ حقه ذو إذا كان virtual.
  a.print();     // تمشي وتأخذ دالة الكلاس Alpha بالنسبة
                 لـ C العادية، في C++ يلتقي هذا الكلام بحيث يذهب لكلاس Alpha
                 إذا هو موجود ينزل للكلاس المؤشر عليه "Beta" ويأخذ الدالة منه
}
```

الواجب:

- القاعدة تقول: امشي من تحت لفوق أول ما يحصلها يأخذها
<< تنزل بالاشتقاق أساساً >> أي هلها تحت بس باسم
الدالة اللي فوق تنزل.

- في حالة الدالة هلها تحت وما هلها فوق لدينا حلين:

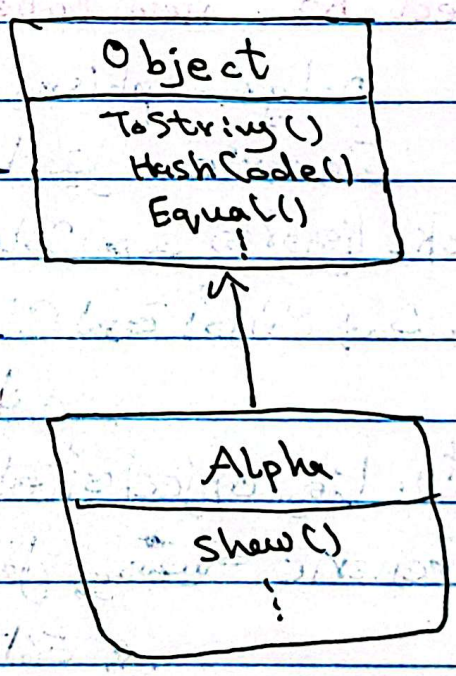
1- لا تأخذه على هيئة Alpha وإنما على هيئة Beta << تحويل نوع >>
كأنما قلت له روح أتأكد من Beta // ((Beta) a).Show();
تأكد من Alpha << البوينترات هنا سهلة التحويل >>

2- قاعدة: كل كلاس عندما يتم إنشائها يتم عمل اشتقاق أتوماتيكياً من
كلاس اسمه object.

Object b = new Alpha();

- أي object من b يمكن أن تكون هدف
من B أو من Alpha

- أي حاجة في base class تقدر
تعمل لها (override)، إذا لم توجد في
Subclass يعمل
وإذا مش موجودة يأخذ اللي فوق،
يعني << موجودة موجودة >>



موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

لنفترض أن Alpha هي TextBox هل نستطيع أن نقول ←

صحيح أن b مؤشر عليه لكن لا نستطيع عملها بهذا b.Text ;

الشكل، نعمل لها قول النوع ← ((textBox) b).Text ;

لماذا لا نستطيع قول b.Text ؟ لأنها بال subclass وليست

بال Parent، فهو يبحث بال b هل هي موجودة؟ لو كانت!

مش موجودة ما تمشي معاه. لكن دوال b يمشين «وما يعرف يأثر

إلا بالذي معاه فقط».

- فنعمله قول نوع عشان يقارن القالب حق textBox

صلاس object يأثر على كل الأدوات والكلاسات.

نفس النوع يؤشر عليه object b2 = new Label(); //

نوعين مختلفين ، وهذا هو polymorphism

- عندما تعمل دوال بأكثر من هيئة هذا يعتبر Generic

لو لا ال Generic لما اشتغلت الأحداث ← Click, Keypress

لذلك تستطيع ربط حدث مثل Click لعدة أدوات بحدث واحد

لماذا؟ لأن ال object يأثر عليهم.

ومعنى هذا أنه يمكن أن ترسل للدالة أي object وهذا ال object

قد يكون أي أداة لذا الدالة تشتغل بوضعية Generic.

«الدالة تكون مهيأة لأنواع مختلفة».

∴ الدوال بوضع overloading لأنها تتعامل مع label, button

TextBox وهي دالة واحدة.

Generic نفس الدالة لأنواع مختلفة

الواجب : .....

موضوع الدرس : ..... اليوم : .....

التاريخ : / / ٢٠م الموافق : / / ١٤هـ

```
ChText (b);
ChText (b2);
ChText (new TextBox());
ChText (object b)
{ if (b is TextBox)
      ((TextBox)b).Text = "";
  else if (b is Label)
      ((Label)b).Text = "";
```

هنا أرسلنا للدالة أنواع مختلفة

هنا بدأنا بالتعرف ولماذا أ مباشرة لأن الأنواع مختلفة .

```
Alpha a1 = new Beta();
Alpha a2 = new D();
Alpha a3 = new E();
foo(a1);
foo(a2);
foo(a3);
void foo (Alpha a)
{ a.show(); }
Beta b1 = new Beta();
foo(b);
```

تشتغل في حال show كانت موجودة في Alpha

تمشي لأنه ورث من كلاس Alpha الدالة

الواجب : .....

موضوع الدرس: ..... اليوم: .....

التاريخ: / / ٢٠م الموافق: / / ١٤هـ

- كلمة virtual تسمى Abstract
- أي كلاس تشتق ترجعها pure أعلى Abstract
- كلاس من نوع pure لا أستطيع أكون منه object
يجوز في حالة واحدة فقط أن تؤشر على subclass أي تمنح
منها مؤشرات حتى لو فيها حقول أو دوال. ينزلوا بالاشتقاق
فقط. أما أنا فيها تستخدمه لا.
- إذا عملت في أي حقل أو data member في Abstract، الكلاس
هنا يعتبر Abstract.
- interface :
في ال C++ الكلاس من نوع Abstract كما يعني Interface
في ال C# أضافوا كلمة interface وهي عبارة عن
Abstract class كل ما فيه Abstract public، عملوا بدالها
كلمة interface وتم استثناء كتابة Abstract و public
أي تتحول أتوماتيكياً أثناء التنفيذ.
- كل ما في ال interface هو Segnature public.
تستخدمه في حال كان لدينا عوامل مشتركة عدد دوال متشابهة
لكن عملها يختلف.

```
Abstract class Alpha | interface class Alpha
{ public abstract void show(); | void show();
  ~ ~ ~ ~ print(); | void print();
}                    | }
```

الواجب: .....

- لأن كلاس Alpha يؤثر على أي كلاس فيها هذه النوعين بحيث تستخدمه مباشرة بدون تحويل ولا شيء
- تمهيد بين مبدأ generic ومبدأ polymorphism
- interface هو pure حتى لا data تعتبر constant

string T; ← object

textBox1.Text.Trim().CompareTo(textBox2.Text.Trim());
↳ مقارنة النصوص

موضوع الدرس: المحاضرة 12 — اليوم: السبت

التاريخ: ١٦/١١/٢٠٢٤م — الموافق: ١٤/٥/١٤٤٦هـ

Focuse() — حدث انتقال التركيز

Back Color Change() — حدث تغير لون الخلفية

Enter() — حدث فور ما يدخل الـ cursor

Key down — حدث يقسم Keypress الى قسمين

up و down ← down عند الاستمرار في الضغط

up عند الانتهاء من الضغط

Leave — حدث المغادرة

closed — بعد الإغلاق

closing — أثناء الإغلاق

← يمكن أنها توقف الإغلاق e.Cancel = true;

- متى تشتغل الـ Enter حق الـ Form ؟

عندما يكون تعمل له focuse() يشتغل

- في بعض الأحداث الكائن e ما تحمل داخله شي سوى الدوال التابعة للكائن object مثل ← e.Equal()

وفي بعض الأحداث تقدر تستفيد منه مثلاً في الـ Keypress تقدر تطلع منه الحرف الذي تم ضغطه من لوحة المفاتيح.

```
{ object o1, o2;
  int x = 100;
  o1 = x;   o2 = x;
```

الواجب:

```
o1.Equals(o2); //
o1.GetType(); //
}
```

هل يساوي ؟!

بيبين نوعه ، هنا بيرجع System.int

لو كان TextBox بيرجع System.Windows.form.TextBox

هذه الدالة يمكن أن تستخدمها بالمقارنة

دالة Equals() ترجع true و false ، وعند المقارنة يقارن بالعنوان وليس بالمحتوى ، في المثال السابق الاثنين بنفس العنوان لذلك بيعيد true .

```
{ object o1, o2;
int x = 100, y = 100;
o1 = x; o2 = y;
o1.Equal(o2);
o1.GetHashCode(); //
}
```

خاصية تستخدم للحصول على قيمة عددية صحيحة تمثل رمز التجزئة لكائن معين

- Font :

- التعامل مع الخطوط . وهي عبارة عن List

- FontStyle موجودة على شكل enum datatype

// حدث Load

```
{ for (int i=0; i< FontFamily.Families.Count-1;
        i++)
    ComboBox1.Items.Add(FontFamily.Families[i]
        .Name);
}
```

// حدث التغير

```
{ string temp = comboBox1.SelectedItem.ToString();
  Label1.Font = new Font (temp , 10 , FontStyle.Bold);
}
```

size

رسيك ما كل سه

- هناك أحداث وليست كثيرة لا تعمل إلا إذا عملت flag قبلها .

this.KeyPreview = true; ← Flag

خاصية على هيئة flag تبع الـ Form . هو أساساً يشتغل لكن إذا وجده يبدأ يفحصه

- خاصية validate تسمى التحقق . ← التحقق من شيء معين داخل الـ Form ⇐