الفصل الثانى

القواعد العامة للغات البرمجة
General Programming Concepts

Programming Concepts 2

فى هذا الفصل

نعرض مقدمة عن أساسيات لغات البرمجة والعناصر المشتركة بين اللغات مع عرض أمثلة بسيطة على ان يتم تفصيل ذلك فى الفصول التالية من خلال النقاط التالية:

- مقدمة
- البرنامج فى اللغات والمتغيرات والإدخال والإخراج
- جمل التحكم فى مسار البرنامج وجمل التكرار
- البرامج الفرعية Subprograms (Proc,Functions)
- التعامل مع المصفوفات فى لغات البرمجة
- مفاهيم البرمجة باستعمال اسلوب OOP
- خاصية التوريث فى مفاهيم البرمجة باستعمال اسلوب OOP
- التعامل مع أخطاء البرنامج Error Handling
- استعمال مفهوم الـ Visual Programming
- مقدمة سريعة عن أوامر قواعد البيانات SQL
- مفاهيم وطرق الاتصال والتعامل مع قواعد البيانات
- نظرية اعادة الاستخدام Reusability والمكتبات
- الرسم فى اللغات Graphics
- برمجة الشبكات Networking Programming
- مفهوم الـ Threading
- برمجة الإنترنت Web Programming

مقدمة

نحاول في هذا الفصل تجميع الخصائص العامة للغات البرمجة مع توضيح كل قاعدة وعنصر من عناصر اللغة مع عرض اختلافها من لغة لأخرى .

واذا قمت بقراءة هذا الفصل أكثر من مرة وبذلت فيه مجهود سوف يوفر عليك الكثير حيث أنه يجمع معظم خصائص اللغات التي سوف نتناولها في الفصول التالية بالتفصيل لذلك حاول قراءته أكثر من مرة .

البرنامج في اللغات

يأخذ البرنامج في معظم اللغات شكل وتركيب محدد وخاصة في اللغات الاولى التي كانت تكتب في بيئة نظام التشغيل DOS حيث كان البرنامج يكتب وحدة واحدة وذلك مثل لغة C .. وفيما يلي أمثلة لتركيب البرنامج في اللغات المختلفة.

1- البرنامج في لغة C

يأخذ البرنامج التركيب الموضح في السطور التالية

```
1:#include <stdio.h>
2: main()
3: {
4: printf("\n Welcome With First Program in C Language\n");
5:        }
```

في هذه السطور يظهر الهيكل الاساسي لبرنامج لغة C ويتكون مما يلي :

السطر #include "stdio.h" ويقوم بفتح (الاشتمال على) الملف stdio.h الذي يحتوى على تعريفات الدوال المهمة والمستعملة خلال البرنامج.

السطر main() فيه الدالة الرئيسية للبرنامج main() وبين أقواسها.

السطر { :هو قوس بداية البرنامج الذي يكتب بعد تعريفات المتغيرات وسطور البرنامج والقوس } لإغلاق الدالة الرئيسية.

لغة Java

برنامج من نوع تطبيق Java Application ، يأخذ التطبيق فى لغة Java الشكل التالى:

```
1  package mypackage3;
2  public class Ex1
3  {
4   public Ex1()
5   {
6   }
7  public static void main(String[] args)
8   {
9   new Ex1();
10    System.out.println("Welcome With First Java Program");
11  }
12 }
13
```

هذا هو الشكل الأساسي لبرنامج Java وفيه يبدأ البرنامج

بالأمر Package الذى يخبر البرنامج بوضع فصائل البرنامج فى حزمة (مجلد) بالاسم mypackage3 ثم يبدأ بفصيلة class بالاسم Ex1 ثم قوس لغة الـ C الشهير) ثم الدالة ()main التى تشابه الدالة الرئيسية فى برنامج ال C وداخل أقواس الدالة ()main تكتب السطور المطلوب تنفيذها ثم يغلق قوس الدالة ()main الذى فتح بالقوس وكذلك يغلق قوس الفصيلة class الذى فتح بالقوس {

هذا بالاضافة لكتابة الاوامر فى أحداث التطبيق مثل Button_clciked وغيره فى حالة التطبيق فى بيئة النوافذ Windows application وهذا هو التطبيق الغالب لذلك سوف نتناول الأحداث Events ومعناها وكيفية توظيفها.

البرنامج فى لغة C#

فى حالة كتابة برنامج فى بيئة نظام التشغيل DOS يقترب تركيب البرنامج فى لغة C# من نفس تركيب البرنامج فى لغة Java ويسمى Console Application ويأخذ الشكل الموضح فى السطور التالية:

```
using System;

namespace ConsoleApplication10
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            //
        }
    }
}
```

يبدأ بالأمر using الذى يخبر البرنامج بالإشارة إلى المكتبة system لإمكانية استعمال الفصائل المعرفة بها ثم يبدأ بفصيلة class بالاسم class1 ثم قوس لغة الـ C الشهير { ثم الدالة ()main التى تشابه الدالة الرئيسية فى برنامج الـ C وداخل أقواس الدالة ()main تكتب السطور المطلوب تنفيذها ثم يغلق قوس الدالة ()main الذى فتح بالقوس وكذلك يغلق قوس الفصيلة class الذى فتح بالقوس {

هذا بالإضافة لكتابة الأوامر فى أحداث التطبيق مثل Button_clicked وغيره فى حالة التطبيق فى بيئة النوافذ Windows application وهذا هو التطبيق الغالب لذلك سوف نتناول الاحداث Events ومعناها وكيفية توظيفها.

البرنامج فى لغة VB.Net

يأخذ البرنامج فى VB.NET نفس تركيب البرنامج فى لغة C# فى حالة كتابة برنامج فى بيئة نظام التشغيل DOS ويسمى Console Application هذا بالإضافة لاستعمال الأحداث Events فى حالة إعداد برامج تعمل فى بيئة Windows.

البرنامج فى Oracle PL/SQL

تحتوى Oracle على لغة برمجة خاصة بها تسمى PL/SQL ويسمى البرنامج فيها بلوك Block ويأخذ الشكل التالى :

```
DECLARE
/* Declare variables which will be used in SQL statements */
v_EName          VARCHAR2(30) ;
v_EmpNo          Number := 7788;
BEGIN
/* Select Data from  Emp table. */
SELECT Ename
INTO  v_EName
FROM emp
WHERE EmpNo = v_EmpNo;
END;
```

فى هذه السطور :

- فى السطر رقم 1 يبدأ البلوك بالكلمة المحجوزة DECLARE ومعناها إعلان وهى تسبق الاعلان عن المتغيرات المستعملة فى البرنامج.
- فى السطر رقم 3 و4 يتم الإعلان عن متغيرين الأول من النوع VARCHAR2 أى حرفى والثانى من النوع Number مع إعطائه قيمة ابتدائية.
- فى السطر رقم 5 تبدأ العمليات داخل البلوك بالكلمة المحجوزة BEGIN وهى تسبق العمليات المطلوب إجراتها فى البلوك.
- فى السطور 7 و 8 يتم كتابة الأوامر المطلوب تنفيذها.
- فى السطر رقم 11 يتم انهاء البلوك بالكلمة المحجوزة ;END.
- وبالتالى ملخص البلوك هو كلمة DECLARE تسبق الاعلان عن المتغيرات التى يتم الاعلان عنها بين DECLARE وbegin بين الكلمة المحجوزة BEGIN والكلمة المحجوزة ;END يتم كتابة سطور العمليات.

من خلال الفقرات السابقة نكون قد قمنابعرض شكل البرنامج فى أكثر من لغة حتى يكون لديك تصور عن ذلك وأن الفروق بسيطة:

المتغيرات Variables و الثوابت Constants فى اللغات

المتغير يستعمل لحفظ قيمة رقمية أو حرفية أو تاريخ أو غيره لاستعمالها أكثر من مرة داخل البرنامج ولا يخلو برنامج من استعمال المتغيرات فعن طريق المتغيرات تستطيع تكوين التعبيرات Expressions وتستطيع تمثيل بيانات البرامج و والمتغيرات تأخذ أنواع كثيرة وتختلف أنواع البيانات الى حد ما من لغة الى لغة ولكن بدرجة بسيطة ،ربما فقط فى المسميات ، فمثلا فى لغة VB النوع Integer هذا النوع فى جميع لغات الـ C (C,C++,Java,C#) يسمى int وهكذا ، مع ملاحظة استعمال نفس الأنواع تقريبا فى لغات الـC التى أطلق عليها C Like وهى C,C++,Java,C#,

الإعلان عن المتغيرات varible declaration

الإعلان عن المتغير هو إعطاء فكرة مسبقة لمترجم البرنامج بأن هناك متغير varible أو متغيرات بنوع معين ويجب الإعلان عن المتغير قبل استخدامه حسب اللغات كما يلى :

فى لغة VB.NET

يتم الإعلان عن المتغير بالصورة التالية :

```
Dim a1 As Integer
```

و في هذا السطر يتم استعمال كلمة DIM في الإعلان عن متغير بالاسم A1 من نوع INTEGER مع استعمال AS قبل النوع وكأنك تقول هناك متغير بالاسم a1 من النوع Integer.

فى لغات الـ C

كما ذكرنا لغات C Like هى لغات C,C++,Java,C# يتم الاعلان بطريقة واحدة وهى كما فى السطور.

```
int  a,b,c;
float k,l,m;
double a1,b2;
```

فى هذه السطور :

يتم كتابة نوع البيانات أولا وليكن int كما فى السطر الأول ثم كتابة أسماء المتغيرات وليكن

a,b,c ، وفى هذه الحالة يكون معنى السطر المتغيرات a,b,c من النوع int أى تقبل قيم صحيحية مثل 10,100,66

فى لغة Oracle PL/SQL

يتم الاعلان بكتابة المتغيرات فى جزء الإعلان داخل البلوك Block ويأخذ الشكل التالى :

```
DECLARE
/* Declare variables which will be used in SQL statements */
v_EName          VARCHAR2(30) ;
v_EmpNo          Number := 7788;
BEGIN
/* Select Data from  Emp table. */
SELECT Ename
INTO  v_EName
FROM emp
WHERE EmpNo = v_EmpNo;
END;
```

فى هذه السطور :

- فى السطر رقم 1 يبدأ البلوك بالكلمة المحجوزة DECLARE ومعناها إعلان وهى تسبق الاعلان عن المتغيرات المستعملة فى البرنامج.
- فى السطر رقم 3 و4 يتم الاعلان عن متغيرين الأول من النوع VARCHAR2 أى حرفى والثانى من النوع Number مع إعطائه قيمة ابتدائية.

التحويلات بين الأنواع فى اللغات

بالإضافة لإمكانية الإعلان عن المتغيرات بالمستويات المختلفة واستعمالها توجد امكانية التحويل من نوع إلى نوع إذا كنت بحاجة إلى ذلك وتوجد أكثر من طريقة.

فى لغة VB.NET

1- كلمات التحويل conversion keyword

توجد مجموعة من الكلمات التى توفرها Visual Basic .NET للتحويل إلى الأنواع المختلفة وهى تستعمل بسهولة كما فى الصورة التالية:

```
Dim K1 As Integer
Dim Q1 As Double
K1= 432
Q1= K1
Q1 = Math.Sqrt(Q1)
K1 = CInt(Q1)
```

- في هذه السطور تم الإعلان عن متغيرين الأول من النوع Integer والثاني من النوع Double.
- ثم تم استدعاء الدالة Sqrt لحساب الجزر التربيعي لقيمة المتغير Q1.
- ثم تم وضع قيمة المتغير Q1 في المتغير K1 ولكن بعد وضع المتغير Q1 كمعامل للكلمة CInt(Q1) التي تقوم بتحويله أولا إلى النوع Integer.
- كما يوجد جدول يعرض كلمات التحويل المتاحة في الكتاب الفني لميكروسوفت.

في لغة Java

يتم استعمال طريقتين من التحويل

1- استعمال طريقة Casting

وهو إما تحويل تلقائي كما في السطور التالية

```
byte a=20;
short b;
int c;
long d;
float e;
double f;
b=a;
c=b;
d=c;
E=d;
F=e;
```

أو باستعمال أوامر التحويل .

2- استعمال دوال التحويل في الفصائل

حيث تحتوى الفصائل classes على دوال للتحويل إلى أنواعها مثل الفصيلة Integer التي تحتوى على الدالة ()ParseInt و هكذا .

فى لغة #C

1- دوال التحويل conversion keyword

توجد مجموعة من الدوال الأعضاء بفصائل الأنواع توفرها C# للتحويل إلى الأنواع المختلفة وهى تستعمل بسهولة كما فى الصورة التالية:

```
Int k;
String q="144";
K=INT32.Parse(q)
```

فى هذه السطور تم الإعلان عن متغيرين الأول من النوع Integer والثانى من النوع string

ثم تم وضع قيمة المتغير q فى المتغير K ولكن بعد وضع المتغير q كمعامل للدالة Parse التى تقوم بتحويله أولا الى النوع Integer.

2- استعمال الفصيلة Convert Class

توجد فصيلة Class بالاسم Convert تحتوى على عدد من دوال التحويل التى يمكنك استعمالها لتحويل نوع الى نوع أخر .

دوال الإدخال و الإخراج I/O Functions

كما هو مشهور في جميع لغات البرمجة توجد جمل (دوال) الإدخال و الإخراج I/O Functions التى تستعمل في استقبال قيم من المستخدم و كذلك عرض الرسائل و القيم له .

و بالرغم من اختلاف ذلك شيئا ما مع برمجة النوافذ إلا أنه توجد دوال كما يلي :

فى لغة VB.NET

دوال الإخراج (عرض الرسائل)

الدالة MSGBOX

و تأخذ الصورتين:

```
1-MSGBOX MSG
2-MsgBox(prompt[, buttons] [, title] [, helpfile, context])
```

فى لغة #C

نفس الدوال الموجودة فى لغة VB.Net ويظهر ذلك من السطور

```
string v1, tit;
   v1 = textBox1.Text;
   tit = textBox2.Text;
   DialogResult r;
   MessageBoxButtons buttons = MessageBoxButtons.YesNo;
     r = MessageBox.Show(this, tit, v1,buttons,
   MessageBoxIcon.Question, MessageBoxDefaultButton.Button1,
   MessageBoxOptions.RightAlign);
```

بالاضافة لأمر طباعة المخرجات فى بيئة نظام التشغيل DOS كما فى السطر.

```
Console.WriteLine("This is The ConsoleApplication ");
```

فى لغة الـ Java

يتم طباعة المخرجات فى بيئة نظام التشغيل DOS كما فى السطر.

```
System.out.println("Welcome with first Java program..");
```

التكرار Looping

من العمليات المسموح بها فى جميع اللغات عملية التكرار وهى تكرار تنفيذ عملية معينه أو مجموع سطور أكثر من مرة وهى مفيدة بدرجة أنها لا يمكن الاستغناء عنها فى كثير من البرامج.

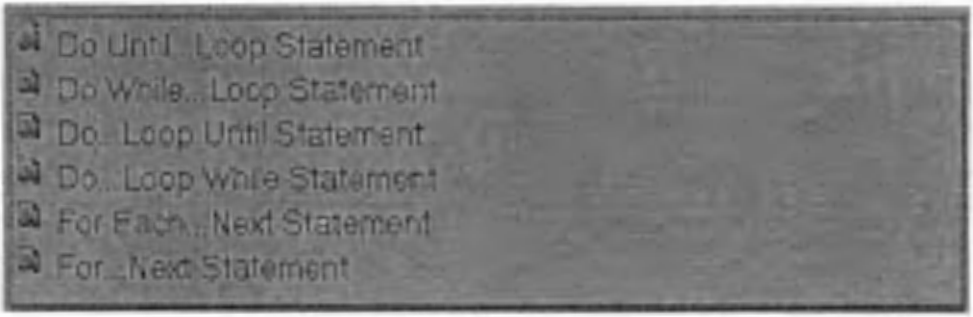
وينقسم التكرار Looping إلى نوعين:

- تكرار معلوم العدد: وفيه معروف عدد مرات التكرار وتستعمل له الجملة for بأشكالها المختلفة.
- تكرار غير معلوم: وأحياناً يسمى تكرار مشروط لأن التكرار يتوقف على شرط معين. والتكرار من أشهر الجمل في جميع لغات البرمجة حيث تسمح بتكرار عملية أو مجموعة سطور أكثر من مرة وكذلك جمل التي تحدد مسار البرنامج و تحقق رغبات المستخدم و هي جمل التحكم في مسار البرنامج حيث نتناول النقاط التالية:
  - جمل التكرار..... , FOR
  - جمل التكرارFor Each
  - الأشكال المختلفة للجملة FOR

- الجملة WHILE ...WEND
- الجملة While
- الجملة do ---- while
- Loop

فى لغة VB.NET

توجد الجمل التالية للتكرار:



جملة FOR

نستخدم جملة FOR لتكرار تنفيذ عملية عدد من المرات معلومة العدد كما فى السطور:

```
1: Dim i
2:       For i = 0 To 20
3:         MsgBox("I=" & i)
4:   Next i
```

و تأخذ أكثر من صورة يمكنك معرفتها بالرجوع لكتاب متخصص فى اللغة .

جملة DO.....WHILE

الجملة DO WHILE تستخدم لتكرار عملية عدد من المرات غير معلومة العدد و لكن يعتمد هذا التكرر على شرط WHILE حيث تأخذ الصورة العامة التالية :

```
Do [{While | Until} condition]
[statements]
[Exit Do]
[statements]
Loop
Or, you can use this syntax:
```

```
Do
[statements]
[Exit Do]
[statements]
Loop [{While | Until} condition]
```

في هذه الصورة :

يتم تكرار ما بين DO و الفرق الآخر LOOP طالما أن الشرط مكتوب أمام كلمة WHILE صحيح فإذا أصبح غير صحيح يتوقف التكرار .

و تركيبها كما في السطور التالية:

```
Do Until False

Loop
```

```
Do While True

Loop
```

```
Do

Loop Until True
```

```
Do

Loop While True
```

```
For Each Item As String In CollectionObject

Next
```

```
For index As Integer = 1 To 10

Next
```

```
    While True

End While
```

في لغات الـ C

تتقارب جمل التكرار بشكل كبير جدا في لغات الـ C like وهي كما أشرنا اللغات C,C++,C#,Java كما يلي:

الجملة for

وتأخذ الصورة العامة التالية :

```
for ([initializers]; [expression]; [iterators]) statement
```

و هي تستعمل في تكرار تنفيذ عملية معينة عدد من المرات .

فمثلا لطباعة كلمة C# 20 مرة سوف نكتب 20 سطر لتحقيق ذلك بدون استعمال جمل التكرار في حين تستعمل جملة تكرار مع سطر واحد لتنفيذ ذلك.

و يتضح ذلك من السطر التالي :

```
for (int i = 1; i <= 5; i++)
Console.WriteLine("C#");
```

في هذه السطور تم استعمال جملة التكرار FOR.

لتنفيذ السطر رقم 2 عدد 10 مرات و هذا ما يوفر علينا كتابة سطر 2 عدد 10مرات .

جملة DO.....WHILE

الجملة DO WHILE تستخدم لتكرار عملية عدد من المرات غير معلومة العدد و لكن يعتمد هذا التكرر على شرط WHILE حيث تأخذ الصورة العامة التالية :

```
do statement while (expression);
```

في هذه الصورة:

يتم تكرار ما بين DO و الفرق الآخر LOOP طالما أن الشرط مكتوب أمام كلمة WHILE صحيح فإذا أصبح غير صحيح يتوقف التكرار .

الجملة while

تستعمل هذه الجملة لتكرار تنفيذ مجموعة من السطور عدد من المرات غير محدد مسبقا ولكن يتوقف على شرط ولذلك يفضل استعمالها في حالة عدم معرفة عدد مرات التكرار وتاخذ الجملة التركيب التالي:

```
while (expression) statement
```

كما توجد foreach في لغة C# وتأخذ الشكل التالي

الجملة foreach

تستخدم هذه الجملة لتكرار اجراء عملية على عناصر مصفوفةArray أو مجموعة Collection وتأخذ الصيغة التالية:

```
foreach (type identifier in expression) statement
```

فى لغة Oracle PL/SQL

أوامر التكرار LOOPS

جمل التكرار هى أشهر جمل فى لغات البرمجة ،وهى مجموعة من الجمل التى تستعمل لتكرار تنفيذ عملية أكثر من مرة ، ولتقريب ذلك إلى ذهنك تصور لو أنك تريد التعويض فى المعادلة التالية:

```
Y=2*x
```

المطلوب التعويض فى هذه المعادلة فى قيمة x من القيمة 0 إلى القيمة 1000 وحساب قيمة y المقابلة لها ،

تخيل لو أنك استعملت الطريقة التقليدية وقمت بالتعويض بهذه القيم بنفسك بالتأكيد سوف يستغرق ذلك منك جهد كبيرا.

لذلك يوجد بديل فى جميع اللغات هو جمل التكرار LOOPS

ويوجد أكثر من أمر تكرار تنفيذ العمليات وهي :

التكرار LOOP---if condition –Exit

التكرار LOOP----Exit when

التكرار while----loop

التكرار For ----loop

التكرار LOOP---if condition –Exit

وفيه يتم تكرار الأوامر التي بينهما بدون توقف ولإيقاف هذا التكرار لابد من إضافة شرط داخل التكرار كما يلى:

```
LOOP                                    -- delimiter
  statement1;                           -- statements
  . . .
 IF condition THEN                      -- EXIT statement
  EXIT;
END IF;
END LOOP;                               -- delimiter
```

جملة التكرار LOOP----EXIT WHEN

يتم استعمال الامر EXIT WHEN بدلا من استعمال IF وهى أسهل وتأخذ الصورة العامة التالية:

```
LOOP                                    -- delimiter
  statement1;                           -- statements
  . . .
  EXIT [WHEN condition];                -- EXIT statement
END LOOP;                               -- delimiter
```

فى هذه الصيغة تم استبدال جملة الشرط IF وأمر الخروج EXIT بالامر EXIT WHEN

التكرار WHILE...LOOP

فى هذه الجملة يتم إضافة شرط التكرار مع While لتكوين تكرار while بدلا من تحديد شرط باستعمال جملة IF أو EXIT WHEN وبالتالى يختلف تركيب التكرار وذلك كما بالشكل:

```
WHILE condition LOOP  <----------  Condition is evaluated at the
  statement1;                      beginning of each iteration.
  statement2;
  . . .
END LOOP;
```

جملة التكرار FOR

هو من أشهر الطرق في جميع لغات البرمجة حيث يستعمل التكرار تنفيذ عملية عدد من المرات معروفة ، ويأخذ الشكل التالى:

```
FOR counter IN [REVERSE]
    lower_bound..upper_bound LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

- ويمكن تغير قيم التكرار بالعكس أي بدلاً من تكرارها من القيمة 1 إلى 1 ويتم تكرارها من 1 إلى 1 وذلك بإضافة كلمة REVERSE كما بالشكل:

FOR MY_VAR IN REVERSE 1—1 LOOP

```
|
END LOOP;
```

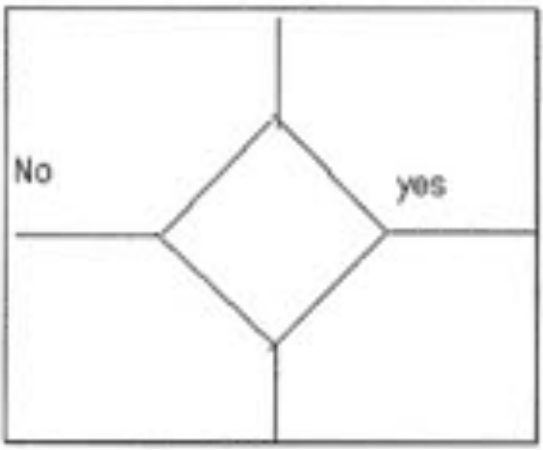
جمل التحكم في مسار البرنامج

في هذه الفقرة نتناول الجمل التالية:

- جملة ..... if
- جملة ....Else..if
- جملة ....End if ..... if ....else if
- جملة select case....end select
- جملة GOTO

هذه الجمل هي التي تحدد مسار البرنامج هل يسير في الخط الطبيعي أم يتفرع إلى دالة فرعية و هي التي تحقق رغبات المستخدم حيث لا يمكن أن يكون هناك برنامج بدون جمل تحكم .

و يتضح ذلك من خريطة التدفق الموضحة في الشكل (12-7):



الشكل (12-7)

في هذا الشكل نلاحظ وجود جملة شرط تحدد مسار البرنامج فإذا كان yes يتجه التنفيذ إلي اتجاه معين و إذا كان NO اتجه اتجاه آخر .

و فيما يلي نشرح جمل التحكم :

فى لغة VB.NET

جملة IF

تستخدم جملة If كما هي بدون else إذا كان هناك اختبار تريد اختباره إذا تحقق تنفذ جمل معينة و إلا يسير البرنامج طبيعي و لا ينفذ هذه الجمل و يأخذ ذلك الصورة البسيطة التالية

```
if ... condation ... then
Statements 1
Statements 2
End If
```

في هذه الصورة إذا كان الشرط condation صحيح تنفذ الجمل , statements 2 statements 1 و ينهي ذلك كلمة end If . وهذا صور أخر سوف نتناولها بالتفصيل

فى لغات الـ C-Like

تتشابه الجمل فى هذه اللغات بدرجة عالية جدا كما يلى

جملة IF

تستخدم جملة If كما هي بدون else إذا كان هناك اختبار تريد اختباره إذا تحقق تنفذ جمل

معينة و إلا يسير البرنامج طبيعي و لا ينفذ هذه الجمل و يأخذ ذلك الصور العامة التالية :

```
if (expression)
   statement1
[else
   statement2]
```

في هذه الصورة إذا كان الشرط condation صحيح تنفذ الجمل , statements 2 statements 1 و ينهي القوس)

جملة if else

تستخدم جملة else .. if إذا كان هناك أكثر من احتمال نريد اختياره حيث تأخذ الشكل التالي :

```
if (expression)
   statement1
[else
   statement2]
```

في هذه الصورة يتم اختيار الشرط CONDATION بجملة IF فإذا كان الشرط صحيح تنفذ الجملتين STATEMENT1, STATEMENT2 و إلا (إذا كان الشرط غير صحيح) تنفذ الجمل التي بعد ELSE و هي في الصورة STATEMENT3, STATEMENT4.

تركيب if elseif

يستعمل هذا التركيب عندما يكون هناك أكثر من احتمالين (شرطين) نريد اختيارهما و يأخذ الصورة :

```
if CONDATION
else if CONDATION THEN
else if CONDATION THEN
else
```

في هذه الصورة

إذا تحقق الشرط الأول يختبر الشرط الثاني وهكذا

لاحظ المثال الموجود بالمسار EX12 باسم NESTED_IF

تركيب switch...CASE

يستعمل هذا التركيب محل التركيب السابق و هو اختيار أكثر من حالتين و هو أسهل في التركيب و الاستعمال عن التركيب السابق .

و يأخذ الصورة العامة التالية:

```
switch (expression)
{
   case constant-expression:
      statement
      jump-statement
   [default:
      statement
      jump-statement]
}
```

في هذه الصورة

وهي الصورة التقليدية والتي شرحناها في الصورة السابقة.

جمل الانتقال

توجد مجموعة من الاوامر التي تؤدي الى الانتقال الى سطر معين وهي كما يلي:

- goto
- break
- continue
- default
- return

جملة GOTO

تستخدم GOTO لتغير مسار البرنامج وتوجه تنفيذ البرنامج الى خطوة معينة وان كانت غير مستحبة في عالم البرمجة وتأخذ الشكل التالي:

```
goto identifier;
goto case constant-expression;
```

في لغة Oracle PL/SQL

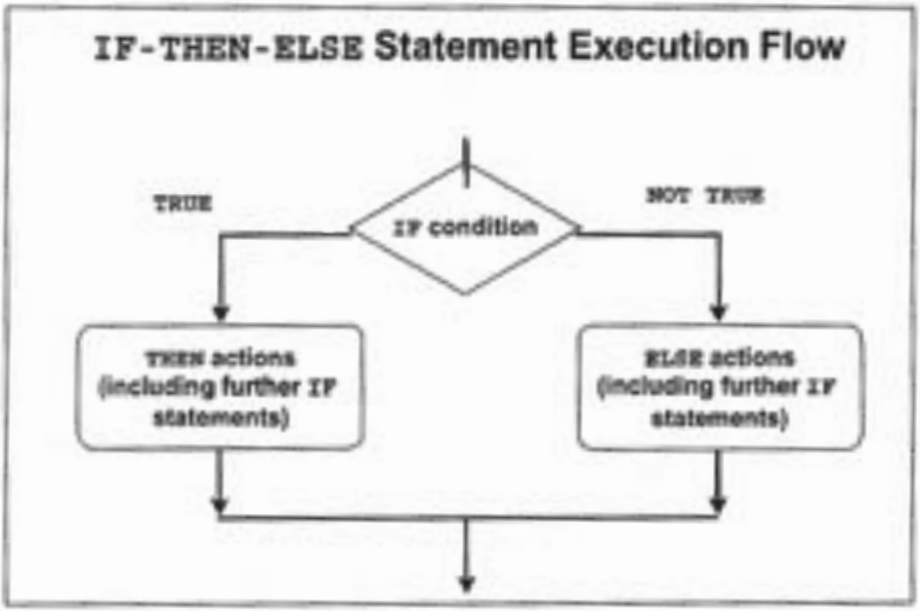
1. الأمر IF

وتأخذ جملة IF الشكل العام

```
IF condition THEN
  statements;
[ELSIF condition THEN
  statements;]
[ELSE
  statements;]
END IF;
```

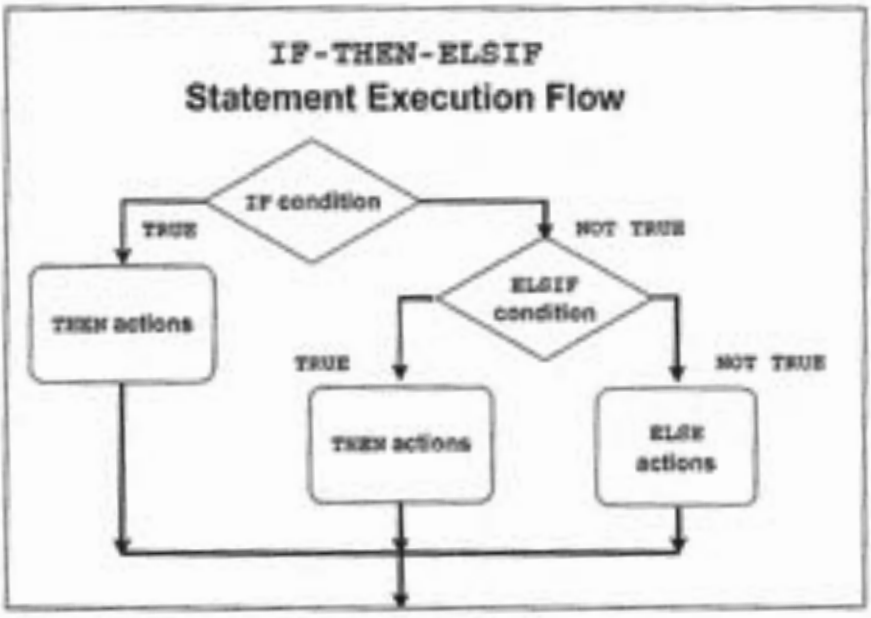
2. الصيغة IF ELSE

وتأخذ الصورة



الامر IF ELSIF

يستعمل هذا الامر في حالة تعدد الشروط بمعنى انك لديك أكثر من اختيار ويتضح ذلك من الشكل 6



معني الدالة و البرنامج الفرعي FUNCTION & SUB

إذا سبق لك دراسة لغة من لغات البرمجة مثل لغة C و غيرها فأنت تعرف أن أي برنامج يتكون من برنامج رئيسي و برامج فرعية و دوال يتم استدعائها من البرنامج الرئيسي أي أن الدالة و البرنامج الفرعي هما مجموعة من الأوامر التي تكتب داخل اطار اسم برنامج فرعي يسمي دالة FUNCTION أو SUB يتم إنشائه مرة واحدة و يمكن استدعائة أكثر من مرة مما يوفر اعادة كتابة هذه الأوامر أكثر من مرة.

الفرق بين الدالة FUNCTION و البرنامج الفرعي Procedure

الفرق الأساسي بين ال FUNCTION و Procedure (الإجراء) هو أن ال FUNCTION لابد ان تعيد قيمة عند استدعائها مثل دالة حساب AVGأو SIN أو أي دالة من دوال اللغة. بينما يقوم البرنامج الفرعي Procedure (الإجراء) بتنفيذ مجموعة من السطور دون أن تعيد قيمة بينما ينفذ الجمل لتحقيق حدث و سوف يتضح ذلك عند التعامل معها .

التعامل مع البرنامج الفرعي

يتم التعامل مع البرنامج الفرعي SUB أو الدالة خلال مرحلتين هما :

أ) إنشاء البرنامج الفرعي (الإجراء)

ب) استدعاء البرنامج الفرعي (الإجراء)

فى VB.NET

ينشأ البرنامج الفرعى SUB (الإجراء) بالشكل

```
Sub s1()
MSGBOX " THIS LINE CALLED FROM SUB S1 "
End Sub
```

فى هذه السطور يتم انشاء برنامج فرعى من النوع Sub بالاسم S1 يعرض رسالة

ويتم استدعاء هذا البرنامج الفرعى بكتابة اسمه فقط بالشكل

```
S1
```

برنامج فرعي من نوع function (الدالة)

يأخذ البرنامج الفرعي من نوع function نفس خطوات الإنشاء و لكنه يختلف عن النوع sub في أنها لابد أن تعيد قيمة مثل الدوال الجاهزة في اللغة مثل دالة abs و دالة sin و غيرها من الدوال .

وتكتب سطورها كما فى الشكل

```
Function f1(ByVal N1 As Integer, ByVal N2 As Integer)
    Dim s1 As Integer
    s1 = N1 + N2
    Return s1
End Function
```

فى هذه السطر يتم انشاء دالة function بالاسم f1

ويتم استدعاء الدالة f1 بالشكل

```
R=F1 (5,7)
MSG BOX " RETURENED VALUE IS : " & R
```

تلاحظ إرسال معاملات لاننا أنشأنا الدالة بمعاملات وكذلك ربد من وضع القيمة المرتجعة فى متغير وفى هذه الحالة المتغير R

فى لغات C-Like

كما ذكرنا لغات C-Like هى اللغات التى تلتزم بقواعد لغة C فى كتابة الاوامر وهى C,C++,C#,Java تنشأ الدالة function كما فى السطور.

```
        void FirstFunc()
        {
        MessageBox.Show ("First Function.....");
        }
```

فى هذه السطور تم انشاء دالة بالاسم FirstFunc وتنحصر سطور الدالة بين الاقواس {} وكلمة void تعبر عن نوع القيمة المرتجعة وفى هذه الحالة void تعنى ليس هناك قيمة مرتجعة وبالتالى تقابل هذا النوع من الدوال نوع الإجراء Procedure فى لغة VB.NET ولكن إذا كانت الدالة تعيد قيمة يحدد نوع هذه القيمة كنوع للدالة ، ويتم استدعاء الدالة بكتابة اسمها ومعاملاتها إذا كان لها معاملات كما فى الشكل.

```
        private void button1_Click(object sender, System.EventArgs e)
        {
        FirstFunc();
        }
```

فى Oracle PL/SQL

يتم إنشاء الإجراء الفرعى Procedure كما فى السطور

```
    PROCEDURE DEPT_SEARCH IS
    BEGIN
      SELECT DNAME INTO :TEXT_ITEM6  FROM DEPT WHERE
DEPTNO=:TEXT_ITEM5;
    END;
```

تلاحظ أنها بدئت بالكلمة PROCEDURE ثم اسم الإجراء وتنحصر مكونات الإجراء بين الكلمتين BEGIN و END وينفذ الإجراء بكتابة اسمه كما فى السطر

DEPT_SEARCH;

إنشاء دالة CREATING PL/SQL FUNCTION

يتم ذلك كما فى السطور

```
1: Create function MySum (No1 Number,No2 Number )
2: RETURN NUMBER
3: IS
4: S NUMBER;
5: BEGIN
6: s=No1+No2;
7:  9: RETURN s;
1 : END;
```

المصفوفات Arrayes

في هذا الدرس نشرح موضوع المصفوفات Arrays بأنواعها وكيفية تمثيلها واستخدامها و ذلك من خلال النقاط التالية :

- **معنى المصفوفات و استخدامها .**
- **أنواع المصفوفات Arrays types.**
- **مصفوفة بعد واحد One Dimension.**
- **مصفوفة بعدين Two Dimension .**
- **خصائص المصفوفات**

معنى المصفوفة

كما تعلم من الرياضيات أن المصفوفة Array عبارة عن مجموعة من القيم (العناصر) من نفس النوع و تأخذ أسم و تأخذ عدد للعناصر .

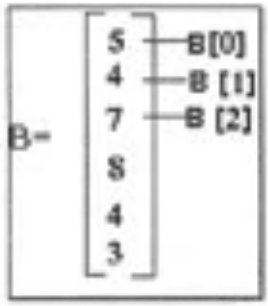
والمطلوب كيفية تمثيلها و استعمالها وكيفية التعامل مع عناصرها و للمصفوفة تطبيقات كثيرة لا تصلح إلا بها مثل عمليات البحث عن قيمة ضمن مجموعة قيم Search

ومثل عمليات ترتيب مجموعة من القيم سواء تصاعدى أو تنازلى Sorting ومثل ايجاد اكبر قيمة ضمن مجموعة قيم Max وايجاد أقل قيمة Min والكثير من التطبيقات وخاصة فى مجال الدرسات العليا والابحاث

أنواع المصفوفات :

(أ) مصفوفة بعد واحد One Dimension كما بالشكل (9.1)

A= [5 4 7 8 4 3]



الشكل (1-9)

- المصفوفة A مصفوفة صف One Row وهي ذات بعد واحد حيث لا يوجد صفوف وأعمدة بل هي صف واحد فقط .
- المصفوفة B مصفوفة عمود واحد One Column لذلك يسمي كلا منهما مصفوفة البعد الواحد و يتم ترتيب العناصر بأسم المصفوفة و ترتيب العنصر مثل A[1] ، B[2] وهكذا.

(ب) مصفوفة متعددة الأبعاد :

وفيها يوجد أكثر من صف Row وأكثر من عمود Column كما بالشكل (2-9) .

$$C=\begin{bmatrix} 5 & 70 & 10 \\ 20 & 40 & 5 \\ 30 & 40 & 7 \\ 5 & 30 & 5 \\ 9 & 40 & 6 \\ 30 & 10 & 7 \end{bmatrix}$$

الشكل (2-9)

مصفوفة البعد الواحد One Dimension :

و يتم التعامل مع هذه المصفوفة بالخطوات التالية :

1- الإعلان عن المصفوفة

2- استعمال عناصر المصفوفة بملئها بالقيم أو استعمال القيم أو طباعة القيم

في لغة VB.NET

الإعلان عن المصفوفة

- يتم الإعلان عن المصفوفة كما يتم الإعلان عن المتغيرات و بنفس درجات الإعلان فيمكن الإعلان عنها و كأنها محلية Local داخل برنامج فرعي Sub أو على مستوي form أو على مستوي التطبيق كما يحدث مع المتغيرات و ذلك حسب المكان و بالصورة التالية :

```
Dim a(10) As Integer
```

أو يتم استعمال الصورة الحديثة التالية

```
Dim Arr As Integer()=New Integer(10) {}
```

وطالما وجدت مصفوفة لابد من وجود كود التكرار for للتعامل مع عناصر المصفوفة كما في السطور:

```
1. Dim a(10) As Integer
2. Dim i As Integer
3. For i = 0 To 10
4. a(i) = i
5. Next
6. For i = 0 To 10
7. Console.WriteLine(a(i))
8. Next
```

في هذه السطور يتم الاعلان عن مصفوفة ثم استعمال for لملئ عناصر المصفوفة ثم استعمال for لطباعة عناصر المصفوفة.

في لغات C-Like

يتم الاعلان عن المصفوفة كما في السطور:

```
int[] myArray = new int [5];
```

في هذه السطور تم الاعلان عن مصفوفة بالاسم myArray من النوع int وعدد عناصرها 5 ويتم التعامل مع المصفوفة بنفس الطريقة السابقة ، ألا إن لغات C-Like تستعمل الأقواس {} مكان الكلمات next لتحديد بداية ونهاية البلوك.

مفاهيم البرمجة بواسطة الأهداف OOP

تعتبر مفاهيم البرمجة بواسطة الأهداف OOP من أهم مفاهيم البرمجة الحديثة حيث أصبحت معظم لغات البرمجة تطبقه وبناء الفصائل classes هو أساس البرمجة بواسطة الأهداف OOP وفي هذا الفصل نشرح الخطوات العملية لبناء الفصائل وذلك من خلال النقاط التالية :

- مفاهيم البرمجة بواسطة الأهداف OOP
- انشاء واستعمال فصائل Classes
- أضافة دوال أعضاء adding methods to class
- فصائل بدون دوال بناء classes without constructors
- دوال البناء في الفصيلة constructors
- استعمال أداة designer في التعامل مع الفصائل classes
- إنشاء خصائص للفصيلة properties
- إضافة خصائص قراءة أو كتابة فقط read-only and write- only

مفاهيم البرمجة بواسطة الأهداف Object Oriented Programming

تعتبر البرمجة بواسطة الأهداف (object oriented programming) من الاتجاهات الحديثة للبرمجة ومن أول اللغات التي طبقت هذا المفهوم لغة C++ وكذلك من اللغات التي تلتزم بمفهوم OOP لغة JAVA ولغة VB.NET زادت من اهتمامها بمفهوم OOP بل وتعتبر ذلك من التعديلات المهمة والتي زادت من قوة VB. حيث نتناول النقاط التالية :

- معنى البرمجة بواسطة الأهداف Object Orient Programming
- معنى الفصيلة class
- ما هي دالة البناء و دوال الهدم constructors & destructors
- ما هي خاصية التوريث inheritence
- ما هي خاصية over loading

معنى البرمجة بواسطة الأهداف OOP(Object oriented programming

تبني فكرة البرمجة بواسطة الأهداف (OOP) على استعمال الهدف (Object) كوحدة برمجة بمعنى انه بدلاً من استعمال الدوال والأوامر لبناء البرنامج مما يضطر المبرمج لإعادة كتابة الأوامر كل مرة لتحقيق فكرة معينة وهذه كانت فكرة البرمجة التقليدية .

ولكن أتت البرمجة بواسطة الأهداف (OOP) لتجعل وحدة بناء البرمجة كبيرة وهي هدف Object أو فصيلة class وبالتالي يتم إعداد مجموعة من الفصائل العامة classes التي تلبي معظم متطلبات إعداد برنامج والتي يحتاجها المبرمج حتى ان بعض المبرمجين يشبه البرمجة بواسطة الأهداف بالبناء باستعمال المباني الجاهزة والبرمجة بالطريقة التقليدية القديمة تشابه البناء باستعمال الأدوات الأولية وبالتالي الفرق بينهما في السرعة كبيرة جدا

معنى الفصيلة class

الفصيلة class هي أساس البرمجة بواسطة الأهداف (OOP) وهي التي يبنى عليها البرنامج وأخذت فكرة الفصيلة CLASS من الواقع فكل عنصر من عناصر الحياة عبارة عن فصيلة class ، فأنت تستطيع أن تطلق على جميع السيارات إنها من فصيلة CAR مع بعض الاختلافات ،ويمكنك ان تطلق على الطيور فصيلة Bird أي طائر وهكذا تنتمي جميع العناصر الى فصائل classes ، وكل فصيلة تستطيع تمثيلها بعنصرين هما البيانات والدوال (data, methods)، فمثلاً فصيلة الموظف Employee بياناتها هي بيانات الموظف العامة مثل كود الموظف ، اسم الموظف ،عنوان الموظف ،تليفون الموظف ، وباقي بياناته ، وكذلك الدوال (methods) هي دوال تحقيق العمليات التي يمكن أن تتم على الموظف مثل : عملية إضافة موظف جديد ، وحذف موظف موجود ،تعديل بيانات موظف وجميع العناصر يمكن تمثيلها بهذه الطريقة.

ما هي خاصية التوريث inheritance

معنى خاصية التوريث (inheritance) هو توريث فصيلة Base class قديمة موجودة بالفعل لفصيلة class جديدة عند إنشائها بحيث يضاف تركيب الفصيلة Base class القديمة من بيانات و دوال إلى الفصيلة الجديدة new class ثم تكمل عليها في الفصيلة الجديدة.

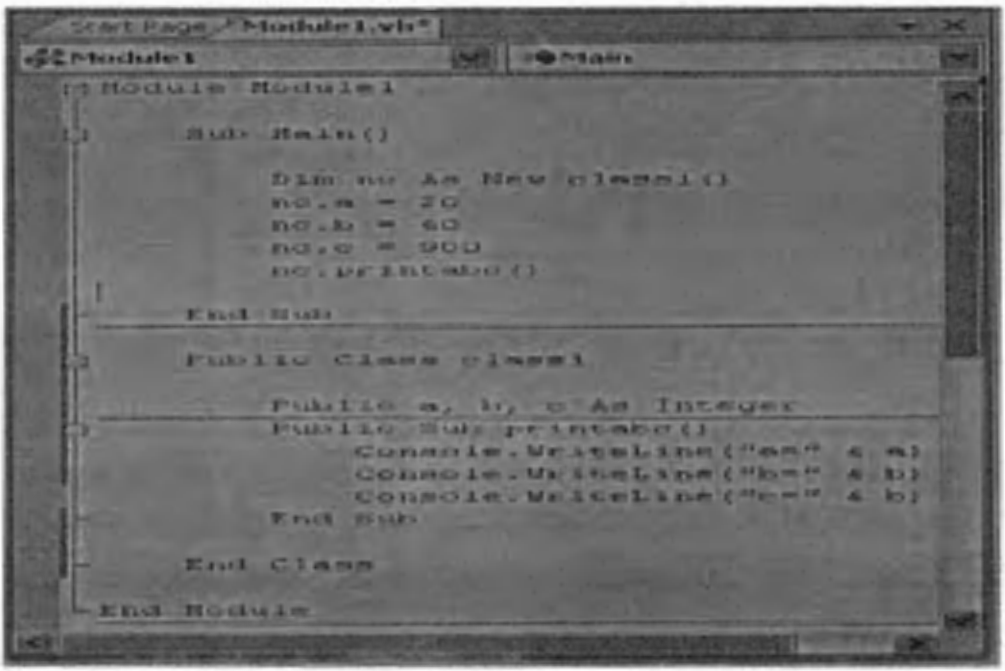
ما هي الخاصية Overloading

معنى خاصية **overloading** : هو إمكانية إنشاء اكثر من دالة Method بنفس الاسم مع تغيير عدد المعاملات Parameters مثلاً ،وهذا يفيد بإنشاء اكثر من دالة لنفس الوظيفة بمعاملات مختلفة بنفس الاسم ، مثل أنشاء دالتين بالاسم ( )find cust إحداهما تأخذ معامل رقمي هو كود الموظف ،والثانية تأخذ عبارة حرفية هي اسم الموظف وفي الحالتين تبحث الدالة ،و بالتالي يشعر المبرمج كأنها دالة واحدة ولكنها دالتين يتم استدعاء كل واحدة تلقائياً حسب المعامل المرسل لها.

ومن أهم ما تحب توضيحه في هذا الفصل شكل البرنامج باستعمال أسلوب OOP

في لغة VB.NET

يأخذ الشكل التالي



في هذا الشكل تم إنشاء فصيلة class باستعمال كلمة class وهي الكلمة المستعملة في جميع اللغات لإنشاء فصيلة.

ثم يلى ذلك أسم الفصيلة وهى فى هذه الحالة class1 ثم يلى ذلك محتويات الفصيلة من متغيرات ودوال وهو تركيب أى فصيلة فى أى لغة (متغيرات variables و دوال methods)

ثم يتم التعامل مع الفصيلة بتعريف متغير هدف object من الفصيلة كما فى السطور حيث تم تعريفه بالاسم nc

ثم التعامل مع مكونات الفصيلة من خلال متغير الهدف.

فى لغات C-Like

تأخذ فكرة انشاء الفصيلة class نفس الفكرة السابقة مع استعمال الاقواس كما هو معتاد كما فى الشكل:

تلاحظ أنها نفس القواعد السابقة.

مع ملاحظة ان القواعد واحدة والاختلافات بسيطة وبالتالى يمكنك تعلم أكثر من لغة حيث أن القواعد الاساسية واحدة.

معالجة الاخطاء Error Handling

في هذا الفصل نعرض موضوع الأخطاء الوارد حدوثها مع المبرمج ومع مستخدم البرنامج وذلك من خلال النقاط التالية:

- الأنواع المختلفة للأخطاء : Bugs
- الخطأ الهجائي Syntax error
- الخطأ أثناء التشغيل Runtime Error
- ON ERROR GOTO-
- ON ERROR RESUME NEXT-
- الخطأ المنطقي Logic Error
- طرق حل الأخطاء :
- طرق تلافي خطأ أثناء التشغيل
- طرق تتبع الخطأ المنطقي ومعرفة سببه
- استعمال الشاشات المختلفة
- استعمال التركيب الجديد Try.....Catch

عند القيام باعداد برنامج يصادفك أخطاء ما أثناء الإعداد أو بعد توزيع البرنامج للاستخدام ، ومن عوامل نجاح البرنامج عدم ظهور أي أخطاء ، وفيما يلي توضح أنواع الأخطاء وطرق تلافيها .

أنواع الأخطاء :

1- الخطأ الهجائي Syntax Error

وهو أبسط أنواع الأخطاء حيث يظهر عند كتابة نصوص البرنامج وهو خطأ في الحروف الهجائية للأوامر المعرفة للغة أو خطأ في قواعد اللغة فمثلاً عند كتابة جملة For بدون Next أو العكس.

- وعند كتابة جملة If بدون End If.
- عند كتابة أمر MSGBOX ناقص حرف وهكذا.
- وهذا الخطأ يظهر سريعاً ويسهل علاجه.

النوع الثانى : الخطأ أثناء التشغيل : Runtime Error

لا يظهر هذا الخطأ أثناء أعداد البرنامج الا إذا قمت بتجربة البرنامج بينما يظهر للمستخدم نتيجة قيام المبرمج بكتابة أوامر تحقق عملية غير محققة أثناء التنفيذ فمثلاً:

أ- عند صدور أمر فتح ملف من مشغل أقراص (A,B,...) والمشغل غير جاهز تظهر رسالة خطأ RunTime Error ويقطع البرنامج ويعود الى بيئة التشغيل.

ب- عند صدور أمر فتح قاعدة بيانات غير موجودة وهكذا ويعتبر هذا النوع من الأخطاء التى تعيب البرنامج فعندما يظهر للمستخدم الخطأ يقطع البرنامج ويخرج وتعتبر هذه عيوب فى البرنامج ، وبالتالى يجب تلافى هذا الخطأ والا يتقطع العمل فى البرنامج فمثلاً يمكن التأكد من وجود القرص قبل تنفيذ أمر فتح الملف أو وجود قاعدة البيانات أو غيره ومن أشهر هذه الأخطاء القسمة على القيمة صفر Divid By Zero حيث تعطى خطأ Over Flow ولتوضيح ذلك تابع المثال التالى:

طرق علاج الخطأ

توجد أكثر من طريقة لعلاج هذا النوع من الأخطاء حسب أهمية العملية وهى :

1- استعمال الجملة ON Error Resume Next

2- استعمال الجملة ON Error GOTO

3- استعمال مصفوفة الأخطاء

4- استعمال التركيب الجديد Try.....Catch

استعمال التركيب الجديد Try.....Catch

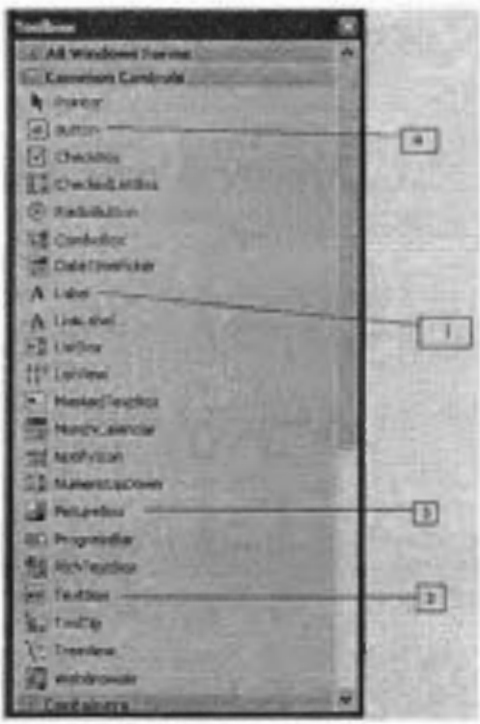
هذا الأسلوب يعتبر جديد فى الاصدار Vb.Net ولكنه كان موجود فى اللغة VC++ وكذلك لغة Java وياخذ الشكل.

```
Try
Statment1
Statment2

Catch Exception1
Action1
```

```
Catch Exception2
Action2

Catch Exception3
Action3
End Try
```

- فى هذه السطور نبدأ بالكلمة Try بمعنى حاول تنفيذ الجمل التالية وهى Statment1,Statment2 فاذا تم تنفيذهم بنجاح اذهب الى End Try وتابع البرنامج
- واذا ظهرت مشاكل اذهب الى مجموعة جمل Catch (أى امسك) التى تتعامل مع الخطأ الناتج.
- فاذا كان الخطأ الناتج من النوع Exception1 نفذ Action1 و كان الخطأ الناتج من النوع Exception2 نفذ Action2

استعمال مفهوم الـ Visual Programming



من المفاهيم الحديثة للغات البرمجة Visual Programming وفيه يتم استعمال أدوات مرئية تسهل البرمجة كثير وتترك للمبرمج التركيز فى منطق البرنامج حيث توفر جميع اللغات بمربع الأدوات ToolBox والشكل يعرض مربع الأدوات و على كل أداة رقم يشير إليها. وهذه هى الأدوات.

مقدمة سريعة عن أوامر قواعد البيانات SQL

فى الفصل الأول قدمنا فكرة عن لغة قواعد البيانات SQL وسوف نتناولها بتفصيل أكثر فى فصل منفصل.

مفاهيم وطرق الاتصال والتعامل مع قواعد البيانات

أصبح التعامل مع قواعد البيانات من الموضوعات المهمة جدا وخاصة فى الوطن العربى ، ولا يمكن تعلم البرمجة فى الوطن العربى بدون تعلم التعامل مع قواعد البيانات وإعداد تطبيقات قواعد البيانات.

لذلك عليك بتعلم أساسيات قواعد البيانات وأوامر SQL وبعض أدوات قواعد البيانات مثل Access,SQL Serverm,Oracle,MySQL.

فى لغة VB.NET

استعمال أوامر الاتصال المباشر Direct Connection

يوفر VB.Net مجموعة من الفصائل Classes التى توفر امكانية استعمال أوامر الاتصال المباشر Direct Connection بقاعدة البيانات حيث تستطيع فتح قاعدة البيانات واصدار الاوامر المختلفة لها مثل الفصيلة Connection class والفصيلة Command class وفصائل أخرى سوف نتناولها بالتفصيل.

تعريفات

تحقيق عمليات قواعد البيانات

من الضرورى تحقيق العمليات الاساسية لقواعد البيانات وهى: الاضافة Insert والحذف Delete والتعديل Update والبحث ويتم ذلك بسهولة بتغيير جملة الـ SQL التى تمثل العملية المطلوبة ، وقد تم شرح جمل SQL المسؤلة عن هذه العمليات.

أهمية التقارير ومولد التقارير Crystal Report

تعتبر التقارير من أهم أجزاء برنامج قواعد البيانات ، فهي الصورة التي يراها المستفيد من البرنامج والتي تخرج فيها مخرجات البرنامج ، فمثلاً عند تصميم برنامج لمتابعة حسابات العملاء لابد من توفير مجموعة من التقارير منها :

- تقرير بحسابات العملاء خلال فترة.
- تقرير بحسابات عميل خلال فترة.
- تقرير بأرصدة العملاء وتقرير برصيد عميل معين ، وهكذا .

وهذه التقارير هي التي تعطي قيمة للبرنامج ، لأن البرنامج بدون تقارير يعتبر بلا فائدة لأنه ينفذ العمليات ولا يعطي النتائج ، وبالتالي يصبح وسيلة لحفظ البيانات فقط .

وكانت عملية تصميم التقارير قبل برمجة النوافذ تتطلب جهد من المبرمج ، حيث يقوم المبرمج بتصميم التقرير باستخدام أوامر البرمجة كما يهتم بضبط الهوامش وغيرها من العمليات الصعبة ، ولكن مع ظهور النوافذ ظهرت برامج منفصلة لتصميم التقارير ثم ربطها بالبرنامج ، ومن هذه البرامج برنامج مولد التقرير المسمى Crystal Report الذي يوفر معظم العمليات المطلوبة لتنظيم التقارير مع إمكانية ربطه بالبرنامج والتقرير الذي يتم تصميمه باستخدام Crystal Report يمكن استدعائه من برنامج مصمم باستخدام لغة VC++ أو باستخدام لغة Visual Basic مع مراعاة الاصدار فإذا سبق لك كتابة برامج باستخدام لغة VB فبالطبع تعرف كيفية استعمال Crystal Report في تصميم التقارير المطلوبة .

خطوات تصميم برنامج به تقارير

1. تصميم البرنامج المطلوب وفي هذه الخطوة يتم تصميم البرنامج وتحقيق عملياته وترك عملية التقارير لآخر خطوة .
2. تصميم التقارير المطلوبة وحفظها في فهرس البرنامج .
3. إضافة أداة استدعاء التقارير إلى البرنامج Crystal Report Control
4. ربط أداة التقارير بالتقرير المطلوب إظهاره مع ضبط الخصائص .
5. كتابة أوامر استدعاء التقرير في البرنامج.

نظرية اعادة الاستخدام Reusability والمكتبات في Visual Studio.Net

المكتبة class Lib

من المفاهيم المتقدمة هو إمكانية إنشاء مكتبة فصائل class Lib تحتوي على مجموعة من الفصائل classes كثيرة الاستعمال بالنسبة لك أو بالنسبة لفريق العمل ووضعها في مكتبة تترجم إلى ملف بالامتداد dll (Dynamic Link Lib) ثم استعمال هذه المكتبة من قبل

أى برنامج جديد لك أو لفريق البرمجة وهذا الإسلوب يوفر الكثير حيث يوفر إعادة كتابة هذه الفصائل classes أكثر من مرة ، كما يوفر إعادة ترجمة هذه الفصائل أكثر من مرة وتصحيح الأخطاء حيث تقوم بترجمتها مرة واحدة وتصحيح الاخطاء مرة واحدة ثم استعمالها كما تشاء كما أنها توفر صورة غير نصية من المكتبة DLL بحيث يمكن استعمالها دون الإطلاع على نص الأوامر أو التعديل فيها.

موضوع الرسم Graphics

من الموضوعات الجذابة والتى تضيف للبرنامج الكثير من المؤثرات التى قد تحتاجها فى بعض التطبيقات وتوفر معظم اللغات أدوات ودوال للتعامل مع عناصر الرسم.

فى Visual Studio.Net

التعامل مع الأداة Picture Box

يمكن عرض صورة فى الأداة Picture Box ولكن فى وقت التصميم وذلك بتوقيع أداة Picture Box على الـ Form ثم عرض الخصائص F4 واختيار الخاصية Image ثم اختيار الصورة المطلوبة ، ويمكن عرض صورة فى الأداة Picture Box باستخدام أوامر البرمجة وذلك بكتابة الصيغة التالية:

```
PictureBox1.image= New BitMap(filename)
PictureBox1.image= New Metafile(filename)
```

ونستخدم الدالة ()BitMap إذا كنا نتعامل مع صور بالامتداد bmp.

ما هى الفصيلة Graphics

توفر هذه الفصيلة إمكانيات الرسم Graphics من إنشاء عنصر رسم Graphics Object مع توفير دوال الرسم المطلوبة.

ويتم استعمال الفصيلة Graphics بتعريف هدف Object منها كما فى السطور التالية:

```
1. Dim MyGraphics as Graphics
2. MyGraphics=Me.CreateGraphics
```

- فى السطر رقم 1 تم الإعلان عن المتغير MyGraphics كهدف Object من نوع الفصيلة Graphics.

- فى السطر رقم 2 تم إنشاء هدف الرسم Graphics Object مع المتغير Me الذى يشير إلى الـ Form والإشارة إلى هذا الهدف بالمتغير المعرف فى السطر رقم 1 MyGraphics وبهذا عند استدعاء دوال الرسم مع الهدف MyGraphics يتم الرسم على الـ Form ولكن للرسم على أداة اخرى غير الـ Form يتم إنشاء عنصر الرسم مع الأداة الأخرى وسوف نوضح ذلك.

استعمال دوال الرسم Graphics Methods

بعد إنشاء هدف Object رسم يتم استعماله فى استدعاء دوال الرسم.

برمجة الشبكات Networking Programming

ما المقصود ببرمجة الشبكات NetWork Programming

المقصود بذلك اعداد تطبيقات تعمل من خلال الشبكة وأشهرها هو تطبيقات تبادل الرسائل Chating المشهورة وإمكانية إرسال الملفات من جهاز إلى جهاز أخر.

وفى هذا الفصل نتناول فكرة إعداد تطبيقات للشبكات NetWork Programming بأكثر من طريقة وفيها نقوم بإعداد نوعين من التطبيقات.

الأول: تطبيق خادم Server وهو التطبيق الذى يستقبل الاتصال من الجهاز المرسل كما يستقبل منه الرسائل.

الثاني: تطبيق مرسل Client وهو التطبيق الذى يقوم بالاتصال بالجهاز الخادم Server ويرسل و يستقبل منه الرسائل.

وتوفر اللغات إمكانية تحقيق تطبيقات الشبكات كما سنتناول فى فصول لاحقة.

مفهوم الـ Threading

ما هو Threading

هى طريقة تسمح بتنفيذ أكثر من عملية Task فى وقت واحد كل عملية تنفذ فى Thread مع إمكانية التحكم فى هذه العمليات مثل إيقاف الـ Thread مؤقتا أو نهائيا.

ومثال لذلك تطبيق الإنترنت الذى يأتيه أكثر من طلب إذا أمكن التعامل مع أكثر من طلب فى نفس الوقت زادت كفاءة التطبيق ويتم ذلك بإنشاء أكثر من Thread باستعمال الفصيلة thread.

ما هى الفصيلة Thread

هى الفصيلة المسؤلة عن تعريف Thread جديد مع توفير الاعضاء (الدوال Methods والخصائص Properties) التى تتحكم فى جميع عمليات الأت Thread مثل بدء تشغيل الـ Thread (الدالة Start()) وايقاف التشغيل Stop() والكثير من العمليات.

برمجة الانترنت Web Programming

ما هي برمجة وتطوير الإنترنت WEB Developing & programming

المقصود بتطوير وبرمجة الإنترنت هو إعداد المواقع و التطبيقات التي تعمل علي الإنترنت .

المواقع Web Sites

الموقع قد يكون بسيط عبارة عن صفحة رئيسية ومجموعة من الصفحات المرتبطة بها وذلك مثل كثير من المواقع ، والغرض منها هو التعرف بالشركة أو الجهة أو الهيئة وعرض أنشطتها ،وبياناتها و كيفية الاتصال بها .

وهناك الكثير من الأدوات التي تستعمل لذلك سوف نعرض لك بعضها .

تطبيق الإنترنت Web Application

تطبيق الإنترنت (web Application) هو تطبيق يعمل من خلال شبكة الإنترنت وذلك لتحقيق وظيفة معينة مثل التطبيق الموجود بموقع شركة التوظيف ، وكذلك تسجيل بيانات الزائرين الذين يريدون وظائف ، ويوفر كذلك إمكانية البحث يستطيع صاحب عمل البحث عن موظفين ، وكذلك يستطيع الزائر البحث عن وظيفة ، وكذلك توجد تطبيق أخرى للإنترنت مثل تطبيقات التجارة الإلكترونية التى تقوم بتوفير قائمة بالشركات المنتجات وتوفير إمكانية اختيار المنتجات وإعداد فاتورة شراء وكثير من التطبيقات الأخرى.

في الفصل السابق عرضنا لك أساسيات لغات البرمجة وشرحنا كم هي متشابه كما عرضنا عناصر لغات البرمجة باختصار.

و في هذا الدرس نبدأ بشرح هذه الأوامر و قواعدها حتى يكون لديك حصيلة من الأوامر و قواعد اللغات التي تساعدك في كتابة ما تريده من أوامر و تحقيق متطلبات البرنامج ونتابع هذه القواعد في الفصول التالية كما يتضح كيف يمكنك الانتقال من لغة الى أخرى بسهولة طالما تعلمت القواعد العامة للغات.

المتغيرات Variables و الثوابت Constants

المتغير يستعمل لحفظ قيمة رقمية أو حرفية أو تاريخ أو غيره لاستعمالها أكثر من مرة داخل البرنامج ولا يخلو برنامج من استعمال المتغيرات فعن طريق المتغيرات تستطيع تكوين التعبيرات Expressions وتستطيع تمثيل بيانات البرامج و والمتغيرات تأخذ أنواع كثيرة.

في VB.NET

لتوضيح بعض هذه الأنواع راجع الجدول التالي:

Integer	متغير يخزن رقم صحيح ولا يأخذ علامة عشرية
Double	متغير يقبل رقم حقيقي يقبل العلامة العشرية
String	متغير يخزن عبارة حرفية
Date	متغير يقبل تاريخ
Boolean	متغير يأخذ احد قمتين True أو False
Byte	متغير يأخذ Byte واحدة
VariantType	يحدد النوع حسب القيمة المدخلة في المتغير

الإعلان عن المتغيرات varible declaration

الإعلان عن المتغير هو إعطاء فكرة مسبقة لمترجم البرنامج بأن هناك متغير varible أو متغيرات بنوع معين ، وكان الإصدار 6.0 لا يفرض عليك الإعلان إلا إذا أردت أن تغير درجة الإعلان كما سيلي أو إذا أردت أن تجبر المترجم أن يبحث عن الإعلان تضبط ذلك من القائمة الرئيسية بينما الاصدار VB.Net يفرض عليك الإعلان عن المتغيرات قبل استعمالها.

و يتم الإعلان عن المتغير بالصورة التالية :

```
Dim a1 As Integer
```

و في هذا السطر يتم استعمال كلمة DIM في الإعلان عن متغير بالاسم A1 من نوع INTEGER مع استعمال AS قبل النوع وكأنك تقول هناك متغير بالاسم a1 من النوع Integer

وبنفس الإسلوب تستطيع الإعلان عن متغيرات من باقي الأنواع التي أشرنا إليها درجات مستوي و أشكال الإعلان .

يأخذ الإعلان عن المتغير درجه معينه تعتمد علي مكان الإعلان وهذه الدرجة تحدد هل يأخذ الإعلان أي الدرجات التالية :

1. هل هو بالمستوى LOCAL أي خاص بالدالة أو الإجراء الذى بداخله المتغير مثل البرنامج الفرعى ()Sub الخاص بزر أمر معين Button.
2. علي مستوي الـ FORM و يسمي PUBLIC وبالتالى يتم التعرف على هذا المتغير داخل جميع دوال وإجراءات هذا الـ Form واذا تم التغيير فيه فى اجراء على المستوى الـ Form يتأثر بهذا التغيير أى إجراء أخر حيث يرى المتغير كأنه معرف لدى جميع إجراءات الـForm بنفس الاسم بنفس النوع.
3. علي مستوي البرنامج GLOBAL ويتم التعرف عل المتغير داخل جميع إجراءات البرنامج.
4. مستوى الفصيلة class وسوف نتناول ذلك فى فصل البرمجة بالأهداف OOP.

وكذلك يمكن للموقع أن يحتوي علي نموذج أو أكثر من التطبيقات وليس تطبيق متكامل فمثلاً شاشة (نموذج) تسجيل البيانات Registration from في أي موقع ويتم بإعداد برامج الإنترنت وكذلك نموذج تطبيقات الزائر feed back وغيرها.

طرق وأدوات إعداد مواقع انترنت web site tools

كما أشرنا في الفقرة السابقة فأن متطلبات إعداد موقع web site تحتاج مجهود وخبرة أقل من متطلبات إعداد تطبيق إنترنت web application وهناك الكثير من الأدوات التي تساعد في إعداد موقع منها ما يلي:

1- استعمال أوامر وعلامات HTML

2- استعمال برامج جاهزة مثل Front Page وبرنامج Dream weaver

3- استعمال برامج حركة مثل Flash

4- استعمال برامج مساعدة مثل Amfy, Cool 3D, XARA 3D, SWICH

5- إضافة برامج JAVA والتي تسمي APPLETS

6- تحميل المواقع علي الإنترنت UPLOAD

هذه بعض أدوات ومتطلبات إعداد مواقع بعضها يجب أن تتعرف علي أساسياته والأخر اختياري يمكنك معرفته أو استبداله وفيما بعد سوف نتناول شرح أساسيات بعض هذه الأدوات.