

ملخص منهج *Visual Basic . Net*



للفصل الثالث الإعدادي
الفصل الدراسي الأول
Mr: Mohammad Zyan

الفصل الأول حل المشكلات

المشكلة (Problem) :

- تعني هدف أو ناتج مطلوب الوصول إليه من خلال إتباع عدة خطوات بترتيب معين .

حل المشكلة (Problem Solving) :

- هو الوصول إلى هدف أو ناتج محدد مطلوب من خلال خطوات وأنشطة متتابعة ومعطيات محددة .

خطوات حل المشكلة :

أولاً تحديد المشكلة Problem Definition

- وفيها يتم تحديد (المخرجات المطلوبة – والمدخلات المتوفرة – وعمليات المعالجة الحسابية والمنطقية) أي معرفة ما هو المطلوب .

ثانياً إعداد خطوات الحل الخوارزمية Algorithm

- ٢- إعداد خطوات الحل الخوارزمية : Algorithm
س : ما المقصود بالخوارزمية ؟
ج : هي مجموعة من الإجراءات المرتبة ترتيباً منطقياً والتي يتم تنفيذها للوصول إلى هدف أو ناتج محدد من معطيات محددة .

س : علي ما يجب أن يكون شكلها ؟

ج : علي شكل سلسلة من الخطوات المتتالية المرتبة ترتيباً منطقياً بحيث إذا تتبعناها نصل إلى الحل .

س : لماذا تسمى خطوات الحل بالخوارزمية ؟

ج : تسمى بذلك نسبة لعالم الرياضيات ومؤسس علم الجبر محمد بن موسى الخوارزمي وهو عالم مسلم عراقي .

س : بما يتم تمثيل خطوات الحل الخوارزمية ؟

ج : يتم تمثيل خطوات الحل من خلال خرائط التدفق .

ثالثاً تصميم البرنامج علي الكمبيوتر Program Design

- وفيها يتم ترجمة خرائط التدفق إلى إحدى لغات البرمجة مثل (الكوبول – الجافا – الفيجوال بيزك دوت نت)

رابعاً اختبار صحة البرنامج وتصحيح أخطائه Program Testing

- وذلك عن طريق إدخال بيانات للبرنامج معروف نتائجها مسبقاً حتي نتمكن من مقارنة النتائج التي نحصل عليها بالنتائج الفعلية وبذلك يتم اكتشاف لأخطاء ونقوم بتصحيحها .

خامساً توثيق البرنامج Program Documentation

- وذلك عن طريق كتابة جميع الخطوات التي اتخذت لحل المشكلة من (مدخلات – مخرجات – أوامر البرنامج – تاريخ آخر تعديل للبرنامج – ومن شارك في عمل البرنامج) بهدف الاحتفاظ به موثق للرجوع إليه في اي وقت بهدف التصحيح .

خرائط التدفق Flow Chart

- هي تمثيل تخطيطي يعتمد علي الرسم بأشكال قياسية لتوضيح ترتيب العمليات اللازمة لحل مسألة أو مشكلة معينة .

فوائد استخدام خرائط التدفق " مميزات خرائط التدفق " :

- ✗ تيسر فهم المشكلة وتوضح للمبرمج ما يجب عمله .
- ✗ يكون من السهل كتابة البرنامج بعد رسم خريطة التدفق له .
- ✗ مفيدة في شرح البرنامج للآخرين .
- ✗ توفر توثيق أفضل للبرنامج وخصوصاً إذا كان البرنامج معقداً .

الأشكال المستخدمة في رسم الخرائط

١ رمز البداية أو النهاية Terminal

الوظيفة : يستخدم في بداية ونهاية خريطة التدفق .

٢ رمز الإدخال أو الإخراج Input / Output

الوظيفة : يستخدم في إدخال وإخراج البيانات .

٣ رمز المعالجة أو العملية Process

الوظيفة : يستخدم في المعالجة (إجراء العمليات الحسابية المختلفة مثل : الجمع - الطرح - الضرب - القسمة)

٤ رمز اتخاذ القرار Decision

الوظيفة : يستخدم عند الاختيار بين قيمتين أو أكثر بناءً على تحقيق شروط معينة .

٥ خطوط اتجاه Flow Lines

الوظيفة : تستخدم لتوضيح ترتيب تدفق خطوات الحل .

ملاحظات هامة

- يجب أن تبدأ خريطة التدفق برمز البداية ويكتب فيه كلمة (Start) وتنتهي برمز النهاية وتكتب فيه كلمة (End) .
- المتغير (Variable) يعني مخزن بالذاكرة يحتوي على قيمة تتغير طوال سير البرنامج .
- أي عملية حسابية توضع في رمز المعالجة .
- رمز الإدخال أو الإخراج يستخدم للتعبير عن إدخال القيم باستخدام كلمة (Enter) أو (Read أو Get أو Input) بينما يتم التعبير عن إخراج النتائج باستخدام كلمة (Print أو Output) .
- رسم خريطة التدفق يكون من أعلى لأسفل ومن اليسار إلى اليمين .
- يمكن استخدام التفرع في خرائط التدفق فعندما نجد شرط (سؤال) في المسألة نقوم باستخدام رمز القرار  ويجب أن يخرج من شكل اتخاذ القرار خطين اتجاه علي الأقل أو ثلاثة خطوط .
- التكرار : يقصد به إعادة تنفيذ أمر أو مجموعة من الأوامر حتي يتحقق شرط معين .
- يتم استخدام الحلقات التكرارية في خرائط التدفق عندما نريد تكرار أمر أو مجموعة الأوامر حتي يتحقق شرط معين داخل المسألة أو المشكلة .
- عند الشروع في حل أي مسألة باستخدام خرائط التدفق يجب تحديد ثلاثة عناصر :
 - ١- تحديد (تعريف المشكلة) .
 - ٢- خطوات الحل .
 - ٣- رسم الخريطة .
- أ- المخرجات (المطلوب) .
- ب- المدخلات (المعطيات) .
- ج- الحل (العملية الحسابية) .

الفصل الثاني مقدمة للغة Visual Basic . Net

معلومات عامة

- لغة *visual Basic . Net* هي لغة برمجة متوفرة ضمن حزمة الفيجوال أستوديو دوت نت (*Visual Studio .Net*).
- الفيجوال أستوديو دوت نت (*Visual Studio .Net*) : يحتوي علي لغات برمجية اخري مثل : *C++* و *C#* و *J#*.
- تمتاز لغة *visual Basic . Net* بانها كائنية التوجه (*Object Oriented*) أي تستخدم لإنشاء تطبيقات النوافذ (*Windows Applications*) أو تطبيقات الويب (*Web Applications*) موجهة بالحدث (*Event Driven*).

ملاحظة

- معنى تطبيق النوافذ *Windows* موجهة بالأحداث *Event Driven* :
- تعني أنه بناء على تصرف المستخدم (الحدث الذي يفعله) مثل النقر *Click* أو الضغط على مفتاح معين من لوحة المفاتيح يتم تنفيذ عمل معين .

تعريف لغة البرمجة :

- هي مجموعة من الأوامر والتعليمات تكتب وفقا لقواعد معينة حسب كل لغة برمجة ويتم ترجمتها إلي لغة الآلة لتنفيذها.
- أو هي لغات تكتب بالحروف الإنجليزية ولكن بقواعد مختلفة تختلف من لغة إلي أخرى ولكل لغة مترجم خاص يقوم بتحويلها إلي لغة الآلة .

لغة الآلة :

- هي لغة جهاز الكمبيوتر حيث لا يفهم ولا ينفذ إلا هذه اللغة وهذه اللغة تتكون من رقمين فقط هما (صفر ، ١) .

تطبيق النوافذ *Windows Applications*

- هو تطبيق له واجهة مستخدم رسومية *GVI* ويعمل من خلال نافذة مثل برنامج الرسام والمفكرة والحاسبة وهذه النوافذ لها خصائص مشتركة من حيث الشكل واحتوائها على أزرار التكبير والتصغير والإغلاق وأشرطة القوائم وأعمدة الإزاحة .
- (*GVI*) اختصار لـ (*Graphical User Internet*) .

المقصود بأن لغة *VB.Net* كائنية التوجه :

- أي أن كل شيء في الفيجوال بيزك دوت نت يعتمد علي الكائنات *Object* مثل : الزر *Button* وصندوق النص *TextBox* والقائمة المنسدلة *ComboBox*..... إلخ .
- لكل كائن في لغة *visual Basic . Net* مايلي :

- وهي تصف الكائن وتحدد سماته مثل الخاصية : *Text - Name* .
- الأفعال التي تقع علي الكائن ويستجيب لها مثل *Click - Key Press* .

1 خصائص *Properties*

2 أحداث *Events*

يقصد بها ما يمكن ان يفعله الكائن لحدوث سلوك معين أي الفعل المصاحب

للكائن مثل *Close – Clear – Open* .

تعريف الكائن *Object* :

• هو وحدة البناء الأساسية في لغات البرمجة كائنية التوجه ، والكائن (يتم إنشاؤه) من تصنيف *Class* معين .

تعريف التصنيف *Class* :

• هو المخطط *Blueprint* الذي يتم إنشاء الكائن منه ويحدد به العديد من العناصر مثل الخصائص *Properties* والوسائل أو وظائف *Methods* و الأحداث *Events* التي سوف يأخذها الكائن *Object* المستمد من هذا التصنيف *Class*.

الفرق بين الكائن والتصنيف

- ١ - التصنيف (*Class*) يحتوي علي تعريف الكائن (*Object*) .
- ٢ - الكائن (*Object*) ليس له وجود إلا عند عمل نسخة منه (*Instance*) من التصنيف الخاص به .
- ٣ - يمكن إنشاء أكثر من كائن من نفس التصنيف .
- ٤ - في لغة *VB.Net* يتم حجز مساحة للکائن في ذاكرة الكمبيوتر عند إنشاؤه .

إطار عمل دوت نت (*.NET Framework*) :

* وهو بمثابة الجهاز العصبي المركزي لجميع تطبيقات فيجوال ستوديو دوت نت ، وهو إطار عمل يمكننا من توفير بيئة التصميم والتشغيل لتطبيقات الـ (*.NET*) : أي توفير

- ١ - إنتاج العديد من تطبيقات (*.NET*) مثل التطبيقات المكتبية *Desktop Applications* و تطبيقات الويب *Web Applications* و تطبيقات الموبايل *Mobile Applications*
- ٢ - توفير بيئة تشغيل لهذه التطبيقات

مكونات إطار عمل دوت نت (*.NET Framework*) :

- ١ - بيئة التشغيل (*Common Language Runtime (CLR)*)
- ٢ - مكتبات تصنيف النظام (*System Class Libraries*)
- ٣ - مترجمات (*Compiler*)
- ٤ - أدوات أخرى.

ملاحظة

• إطار عمل الدوت نت (*.NET Framework*) متوفر بشكل مجاني لتنصيبه علي أي نظام تشغيل .

شاشة *IDE* :

• هي شاشة كتابة البرامج وهي تحتوي على كل ما يحتاجه المطور *Developer* من أدوات ومميزات تساعد من إنشاء تطبيقات (الويندوز - الويب - الموبايل -)

• عند فتح برنامج *visual Basic . Net* سوف تظهر الشاشة التالية وهي صفحة البداية (*Start Page*) .

مكونات شاشة (*IDE*) بعد إنشاء مشروع جديد :

النموذج *Form*

وهو عبارة عن النافذة التي يصمم عليها واجهة البرنامج التي يتعامل معها المستخدم من خلال وضع أدوات التحكم (*Control*) المختلفة عليها .

- أدوات التحكم التي يمكن وضعها علي النموذج مثل (زر الأمر (Button) – صندوق النص (Textbox) إلخ) .

٢ صندوق الأدوات Tool Box

* يحتوي علي مجموعة من أدوات التحكم (Controls) التي يمكن وضعها علي النموذج والتي يستخدمها المبرمج في تصميم شاشات البرنامج .

- يتم تصنيف أدوات التحكم (Control) علي شكل فئات .
- توجد أمام كل فئة علامة (+) عند النقر عليها تظهر مجموعة من أدوات التحكم التابعة لها .
- يمكن عرض كل أدوات التحكم عن طريق الضغط علي علامة (+) أمام الفئة All Windows Form .
- بعض أدوات التحكم الشائعة (Common Controls) الإستخدام :
- زر الأمر (Button) .
- صندوق العنوان (Label) .
- صندوق الاختيار (Check box) .
- صندوق التحرير والسرد (Combobox) .
- صندوق النص (Textbox) .
- صندوق القائمة (List box) .
- صندوق زر راديو (Radio Button) .

٣ نافذة الخصائص Properties Windows

- تستخدم في تحديد خصائص الأدوات .
- لكل أداة من أدوات التحكم مجموعة من الخصائص (Properties) يمكن ضبطها من خلال نافذة الخصائص .
- تنقسم نافذة الخصائص إلي عمودين (عمود أيسر) و (عمود أيمن) (العمود الأيسر يوجد به إسم الخاصية (بينما العمود الأيمن يوجد به قيمة الخاصية) .
- تختلف الخصائص (Properties) المعروضة في نافذة الخصائص حسب العنصر النشط في شاشة (IDE) .

٤ مستعرض الحل Solution Explorer

- يعرض مستعرض الحل قائمة بملفات ومجلدات المشروع (Project) أو المشروعات (Projects) الموجودة ضمن الحل.

إنشاء مشروع جديد (New Project)

خطوات فتح برنامج Visual Basic . Net

- لفتح برنامج (Visual Basic . Net) نقوم باتباع الخطوات التالية :
- ١- إفتح قائمة (Start) ثم إختار (All Program) .
- ٢- إختار Microsoft Visual Studio حتي تظهر شاشة البداية (Start Page) .

خطوات إنشاء مشروع جديد (New Project)

- ١- من قائمة (File) إختار امر (New Project) أو إضغط علي (Create Project) من صفحة (Start Page) فيظهر لك مربع (New Project) .
- ٢- قم باختيار القالب (Windows Form Application) .
- ٣- أدخل إسم المشروع أمام خانة (Name) والبرنامج يعطينا إسمافترضياً هو WindowsApplication1 (والنزى يمكن تغييره) ثم إنقر علي (OK) .
- ٤- بعد النقر علي زر (OK) تظهر نافذة (IDE) حيث يتم إضافة نموذج جديد بإسم (Form1) تلقائياً بمجرد إنشاء مشروع .

إضافة نافذة نموذج (Form) جديد للمشروع (Project)

- ١- من قائمة (Project) إختار أمر (Add Windows Form) لتظهر شاشة (Add New Item) .

٢- اختر منها القالب المستخدم في إنشاء النموذج (Windows Form) ثم اضغط علي زر (Add) لتضاف نافذة النموذج إلي المشروع .

حفظ المشروع (Project) في أحد وسائط التخزين :

- عند إنشاء مشروع جديد يحفظ منه نسخة في ذاكرة الكمبيوتر فقط وبذلك يمكن فقده بسبب غلق الجهاز أو انقطاع التيار الكهربائي لذلك يجب حفظه علي إحدى وسائط التخزين .
- لحفظ المشروع (Project) يجب إتباع الخطوات التالية :
 - ١- إفتح قائمة (File) ثم اختر (Save All) لتظهر نافذة (Save Project) .
 - ٢- من نافذة (Save Project) اختر مكان الحفظ ثم اضغط علي زر (Save) .

ملحوظة مهمة

- عند حفظ المشروع لأول مرة يظهر إسم الحل مطابق لإسم المشروع .

إضافة مشروع (Project) جديد للحل (Solution) :

- لإضافة مشروع (Project) جديد للحل (Solution) نقوم بعمل الآتي :
 - ١- إختار قائمة (File) ثم القائمة الفرعية (Add) ثم الأمر (New Project) .

ملحوظة مهمة

- بعد إضافة المشروع (Project) سيظهر المشروع الجديد في نافذة الحل .

الفصل الثالث

أدوات التحكم Control

معلومات عامة

- هناك خصائص Properties مشتركة بين أدوات التحكم Controls المختلفة مثل خصائص (Name - Text - RightToLeft - BackColor - ForeColor - إلخ) .
- هناك خصائص Properties لا يظهر تأثيرها على أدوات التحكم Controls إلا بعد ضبط خصائص Properties أخرى مثل خاصيتي RightToLeft و RightToLeftLayout الخاصة بنافذة النموذج Form .
- هناك خصائص Properties خاصة بنافذة النموذج Form إذا تم ضبطها تطبق على نافذة النموذج Form وعلى أدوات التحكم Controls التي يتم وضعها على نافذة النموذج مثل خصائص (ForeColor - Font) .

نافذة الخصائص Properties Windows

- ١ - صندوق سرد يوضح خصائص الكائن النشط .
- ٢ - الجزء الأيسر من نافذة الخصائص ويعرض فيه إسم الخاصية المراد تغييرها للكائن المحدد .
- ٣ - الجزء الأيمن من نافذة الخصائص ويعرض فيه قيمة الخاصية المراد تغييرها للكائن .

ملحوظة مهمة

- لتغيير قيمة أحد الخصائص يجب تحديد (تنشيط) الأداة المراد تغيير خصائصها أولاً ثم تغيير قيمة الخاصية المراد تغييرها .

- للنموذج (Form) العديد من الخصائص (Properties) والجدول التالي يوضح بعض هذه الخصائص :

الوظيفة	الخاصية
إسم النموذج المستخدم داخل الكود .	Name
النص الظاهر علي شريط العنوان للنموذج Form	Text
شكل حدود (إطار) نافذة النموذج Form	Form Border Style
لون خلفية النموذج Form	Back Color
تحديد حجم النافذة علي الشاشة في وضع تكبير أو تصغير أو عادي .	Window State
تتحكم في ظهور أو إخفاء صندوق التحكم (True – False)	Control Box
تتحكم في ظهور أو إخفاء صندوق التصغير (True – False)	Minimize Box
تتحكم في ظهور أو إخفاء صندوق التكبير (True – False)	Maximize Box
التحكم في ظهور أو إخفاء أيقونة نافذة البرنامج علي شريط المهام (True – False)	Show In Task Bar
تحدد مكان نافذة النموذج (Form) علي الشاشة .	Start Position
تحديد تخطيط أدوات التحكم من اليمين إلي اليسار (True – False)	Right To Left Layout
تحديد إتجاه أدوات التحكم من اليمين إلي اليسار مثل إتجاه الكتابة لصندوق النص (True – False)	Right To Left
تحديد لون النص علي نافذة النموذج (Form) .	Fore Color

الفرق بين الخاصية Name والخاصية Text :

Text	Name
النص المكتوب علي وجه الأداة .	إسم الأداة داخل كود البرنامج .
تكتب بحروف إنجليزية وعربية قيمتها بين علامتي تنصيص .	تكتب بحروف إنجليزية .
يمكن تكرار النص علي أي أداة أخرى	لا يمكن تكرار نفس الإسم للأداة .
• مهم جدا : القيمة الافتراضية للخاصيتين في بداية البرنامج تكون واحدة لأي أداة .	

سؤال مهم عند ضبط أي خاصية لأي أداة ماذا يحدث

- إما أن يظهر أثرها علي نافذة النموذج أو الأداة فورا أو لا يظهر أثرها علي نافذة النموذج أو الأداة إلا في وضع إختبار البرنامج (Start Debugging) .

سؤال مهم كيفية إختبار البرنامج أو عمل (Start Debugging)

- الطريقة الأولى : يتم النقر علي مفتاح F5 من لوحة المفاتيح .
- الطريقة الثانية : النقر علي أيقونة ▶ (Start Debugging) الموجودة في شريط الأدوات .
- الطريقة الثالثة : الضغط علي أمر (Start Debugging) من خلال قائمة (Debug) .
- لإيقاف إختبار البرنامج أو عمل (Stop Debugging) بالضغط علي أيقونة Stop ■ من شريط الأدوات القياسي . أو من خلال قائمة (Debug) .

- زر الأمر (Button) أحد أدوات التحكم (Control) التي يمكن رسمها علي النموذج (Form) .
- يستطيع مستخدم البرنامج الضغط علي زر الأمر (Button) لتنفيذ مهمة معينة .
- الوظيفة الأساسية لزر الأمر (Button) هي القيام بأداء عمل معين محدد مسبقا .
- لوضع زر الأمر Button علي نافذة النموذج يتم الضغط علي مرتين متتاليتين أو عن طريق السحب والإفلات ليظهر علي النموذج علي هيئة مستطيل .
- لزر الامر (Button) العديد من الخصائص (Properties) والجدول التالي يوضح بعض هذه الخصائص :

الخاصية	الوظيفة
Name	إسم الأداة المستخدم داخل الكود .
Text	النص الظاهر علي شريط العنوان للأداة Button
Back Color	لون خلفية أداة الزر Button
Fore Color	تحديد لون النص علي الأداة Button .
Font	تحديد (شكل وحجم وتأثير) خط النص الظاهر علي الزر Button .
Location	موقع زر الأمر Button علي النموذج Form .
Size	تحديد ارتفاع وعرض زر الأمر Button علي نافذة النموذج Form .
Enabled	تحديد إذا كانت زر الأمر Button سيستجيب لمشغل البرنامج أم لا (True – False)
Visible	تحديد ما إذا كان زر الامر Button مرئي أم لا (True – False) .

- الاداة (Label) أحد أدوات التحكم (Control) التي يمكن رسمها علي النموذج (Form) .
- تستخدم الاداة (Label) في إعطاء عنوان بسيط علي مستخدم البرنامج فهم محتوى النافذة .
- محتوى الاداة (Label) لا يمكن تغييره بالكتابة فيه من قبل المستخدم أثناء تشغيل البرنامج .
- للاداة (Label) العديد من الخصائص (Properties) والجدول التالي يوضح بعض هذه الخصائص :

الخاصية	الوظيفة
Name	إسم الأداة المستخدم داخل الكود .
Text	النص الظاهر علي شريط العنوان للأداة Label
Back Color	لون خلفية أداة الزر Label
Fore Color	تحديد لون النص علي الأداة Label .
Font	تحديد (شكل وحجم وتأثير) خط النص الظاهر علي الزر Label .
Size	تحديد ارتفاع وعرض زر الأداة Label علي نافذة النموذج Form .
Visible	تحديد ما إذا كان زر الاداة Label مرئي أم لا (True – False) .
Image	تحديد الصورة التي تظهر علي الأداة (Label) .
Auto Size	تحديد إذا كان حجم الأداة (Label) سيتغير تلقائيا حسب النص المكتوب عليها أم لا وتأخذ قيمتين (True – False) .
Border Style	تحديد شكل حدود (الإطار) أداة تحكم العنوان (Label) .

- وهو أحد أدوات التحكم *Controls* التي يمكن رسمها على نافذة النموذج *Form* ويستخدم في استقبال مدخلات مستخدم البرنامج النصية .
- يمكن لمستخدم البرنامج تغيير محتوى أداة صندوق الكتابة *Textbox* أثناء تشغيل البرنامج .
- للأداة (*Textbox*) العديد من الخصائص (*Properties*) والجدول التالي يوضح بعض هذه الخصائص :

الخاصية	الوظيفة
<i>MaxLength</i>	تحديد الحد الأقصى لطول النص (عدد الحروف) لأداة صندوق الكتابة <i>Textbox</i> . حتى لا يتمكن المستخدم من إدخال نص يتعدى هذا الحد الأقصى وإن كان يستطيع إدخال أقل منه .
<i>PasswordChar</i>	تستخدم لمعرفة أو تحديد الحرف أو الرمز الذي سيظهر بدلاً من النص المكتوب في حالة إذا ما أردنا عمل كلمة السر أو المرور <i>Password</i> عند إدخالها مثل * أو \$ أو # أو @ إلخ لأداة صندوق الكتابة <i>Textbox</i> .
<i>MultLine</i>	تستخدم لمعرفة أو تحدد ما إذا كانت الأداة متعددة السطور أم لا (<i>True - False</i>) .

ملحوظة مهمة

- أداة صندوق الكتابة *TextBox* لها مجموعة أخرى من الخصائص مثل *Text - BackColor - Name - RightToLeft - Visible - Location - Size - Font - ForeColor* .
- (" ") هذه العلامة تعني أنها سلسلة حرفية فارغة (أى لا يتم وضع قيمة للخاصية *Text*) .

- وهو أحد أدوات التحكم *Controls* التي يمكن رسمها على نافذة النموذج *Form* وتستخدم في عرض قائمة من العناصر .
- للأداة (*Listbox*) العديد من الخصائص (*Properties*) والجدول التالي يوضح بعض هذه الخصائص :

الخاصية	الوظيفة
<i>Items</i>	عبارة عن مجموعة من العناصر التي تعرض علي صندوق القائمة (<i>Listbox</i>) .
<i>Sorted</i>	تحدد إذا كانت العناصر مرتبة أم لا (<i>True - False</i>) .
<i>SelectionMode</i>	تستخدم لمعرفة أو تحدد إذا ما كان من الممكن إختيار عنصر واحد أو أكثر من العناصر المعروضة في صندوق القائمة <i>Listbox</i> . وهي تأخذ قيم <i>None</i> لعدم إختيار أى عنصر (القائمة للعرض فقط) - <i>One</i> لإختيار عنصر واحد فقط - <i>MultiSimple</i> لإختيار عنصر واحد أو أكثر مباشرة باستخدام الفأرة - <i>MultiExtended</i> لإختيار عنصر واحد أو أكثر بمساعدة مفتاح <i>Shift</i> لتحديد عناصر متجاورة أو مفتاح <i>Ctrl</i> لتحديد عناصر غير متجاورة .

ملحوظة مهمة

- الأداة *Listbox* لها مجموعة أخرى من الخصائص مثل *Text - BackColor - ForeColor - Name - RightToLeft - Visible - Location - Size - Font* .

- وهو أحد أدوات التحكم *Controls* التي يمكن رسمها على نافذة النموذج *Form* وهي عبارة عن صندوق به قائمة من العناصر تسدل لإختيار إحداها .
- للأداة (*ComboBox*) العديد من الخصائص (*Properties*) والجدول التالي يوضح بعض هذه الخصائص :

الوظيفة	الخاصية
عبارة عن مجموعة من العناصر الموجودة بالقائمة .	Items
تحدد مصدر العناصر المقترحة لعملية الإكمال وتأخذ قيم عديدة منها (<i>FileSystem History – None – ListItems</i>)	Sorted
عبارة عن النص الذي علي أساسه تقترح جزء من عناصر القائمة وهي ت أخذ القيم (<i>None- Append – Suggest – SuggestAppend</i>)	SelectionMode

ملحوظة مهمة

- أداة التحكم *ComboBox* لها مجموعة أخرى من الخصائص مثل *ForeColor – BackColor – Text – Name – RightToLeft – Visible – Location – Size – Font* .

خصائص نافذة زر الاداة *GroupBox*

- وهو أحد أدوات التحكم *Controls* التي يمكن رسمها على نافذة النموذج *Form* ويستخدم في إحتواء وتنظيم أدوات التحكم ذات الوظيفة الواحدة على نافذة النموذج . مثل أدوات *CheckBox* و *RadioButton* .
- للأداة (*GroupBox*) العديد من الخصائص (*Properties*) والجدول التالي يوضح بعض هذه الخصائص :

الوظيفة	الخاصية
إسم الأداة المستخدم داخل الكود .	Name
النص الظاهر على الأداة <i>GroupBox</i> .	Text
موقع زر الأداة <i>GroupBox</i> علي النموذج <i>Form</i> .	Location
تحديد لون النص على الأداة <i>GroupBox</i> .	Fore Color
تحديد (شكل وحجم وتأثير) خط النص الظاهر على الأداة <i>GroupBox</i> .	Font
تحديد إرتفاع وعرض زر الأداة <i>GroupBox</i> علي نافذة النموذج <i>Form</i> .	Size
تحدد ما إذا كان زر الأداة <i>GroupBox</i> مرئي أم لا (<i>True – False</i>) .	Visible
تحديد الصورة التي تظهر في خلفية الأداة (<i>GroupBox</i>) .	BackgroundImage
تحدد إتجاه كتابة النص من اليمين إلى اليسار (<i>True – False</i>) .	RightToLeft

خصائص نافذة زر الاداة *RadioButton*

- وهو أحد أدوات التحكم *Controls* التي يمكن رسمها على نافذة النموذج *Form* وهي تتيح للمستخدم إختيار بديل واحد فقط من بين عدة بدائل .
- لا يستطيع المستخدم أن يترك البدائل دون إختيار واحدة منها .
- للأداة (*RadioButton*) العديد من الخصائص (*Properties*) والجدول التالي يوضح بعض هذه الخصائص :

الوظيفة	الخاصية
إسم الأداة المستخدم داخل الكود .	Name
موقع زر الأمر <i>RadioButton</i> علي النموذج <i>Form</i> .	Location
تحديد لون النص الأداة <i>RadioButton</i> .	Fore Color
تحديد (شكل وحجم وتأثير) خط النص الظاهر على الزر <i>RadioButton</i> .	Font
تحديد إرتفاع وعرض زر الاداة <i>RadioButton</i> علي نافذة النموذج <i>Form</i> .	Size
تحدد ما إذا كان زر الأداة <i>RadioButton</i> مرئي أم لا (<i>True – False</i>) .	Visible
تحديد الصورة التي تظهر في خلفية الأداة (<i>RadioButton</i>) .	BackgroundImage
تحدد إتجاه كتابة النص من اليمين إلى اليسار (<i>True – False</i>) .	RightToLeft

- وهو أحد أدوات التحكم Controls التي يمكن رسمها على نافذة النموذج Form وهي تتيح للمستخدم إختيار بديل واحد أو أكثر من بديل .
- يستطيع المستخدم أن يترك البدائل دون إختيار واحدة منها ، أو أن يختار كل البدائل .
- للأداة (CheckBox) العديد من الخصائص (Properties) والجدول التالي يوضح بعض هذه الخصائص :

الخاصية	الوظيفة
Name	إسم الأداة المستخدم داخل الكود .
Location	موقع زر الأداة CheckBox علي النموذج Form .
Fore Color	تحديد لون النص الأداة CheckBox .
Font	تحديد (شكل وحجم وتأثير) خط النص الظاهر علي الزر CheckBox .
Size	تحديد إرتفاع وعرض زر الأداة CheckBox علي نافذة النموذج Form .
Visible	تحدد ما إذا كان زر الأداة CheckBox مرني أم لا (True – False) .
BackgroundImage	تحديد الصورة التي تظهر في خلفية الأداة (CheckBox) .
RightToLeft	تحدد إتجاه كتابة النص من اليمين إلي اليسار (True – False) .

الفصل الرابع نافذة الكود (Code Window)

تعريف نافذة الكود :

- هي التي من خلالها يمكن كتابة الأوامر والتعليمات (الكود) بلغة الفيجوال بيزك دوت نت Visual Basic.Net

طرق فتح نافذة الكود :

- يمكنك فتح نافذة الكود (Code Window) بأكر من طريقة وهذه الطرق هي :
 - ١- من قائمة (View) إختار أمر (Code) .
 - ٢- إضغطعلي مفتاح (F7) من لوحة المفاتيح .
 - ٣- الضغط علي الأداة مرتين متتاليتين بزر الفأرة الأيسر .
 - ٤- في نافذة الحل (Solution Explorer) إضغط بالز الأيمن للفأرة علي الملف (Form1.vb) ومن القائمة المختصرة للملف إختار الأمر (Code Window) .

```

Form1.vb* Form1.vb [Design]* Form1.vb [Design] Form2.vb [Design] Form1.vb Start Page
(frmproduct Events) Load
Public Class frmproduct

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Label1.BackColor = Color.Red
    End Sub

    Private Sub frmproduct_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    End Sub
End Class
    
```

* محتويات نافذة الكود Code Window

- ١- اسم الملف الذي يخزن به الكود (Form1.vb) .

- ٢- اسم الملف الذي يخزن به واجهة نافذة النموذج *Form1* (واجهة التصميم) .
- ٣- الإعلان عن تصنيف (*Class*) باسم (*Form1*) .
- ٤- ما بين السطرين تكتب الأكواد الخاصة بالتصنيف (*Form1*) .
- ٥- سطر نهاية التصنيف (*Class*) .

* معالج الحدث Event Handler

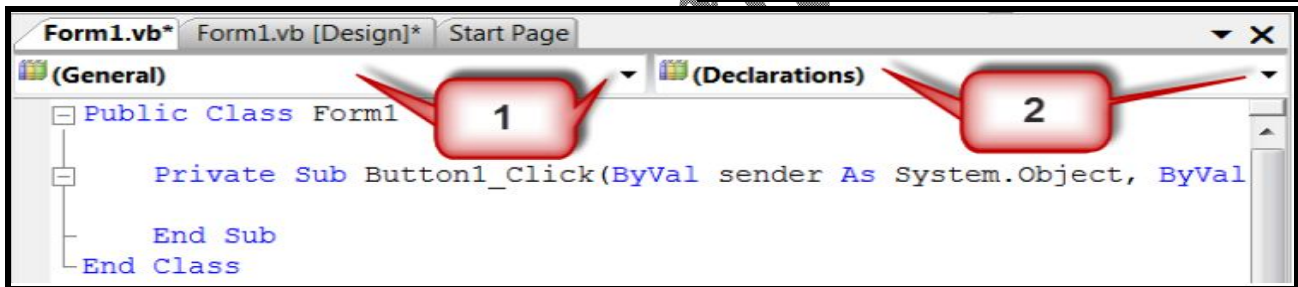
- هو الإجراء الذي يتم استدعائه عند وقوع حدث معين .



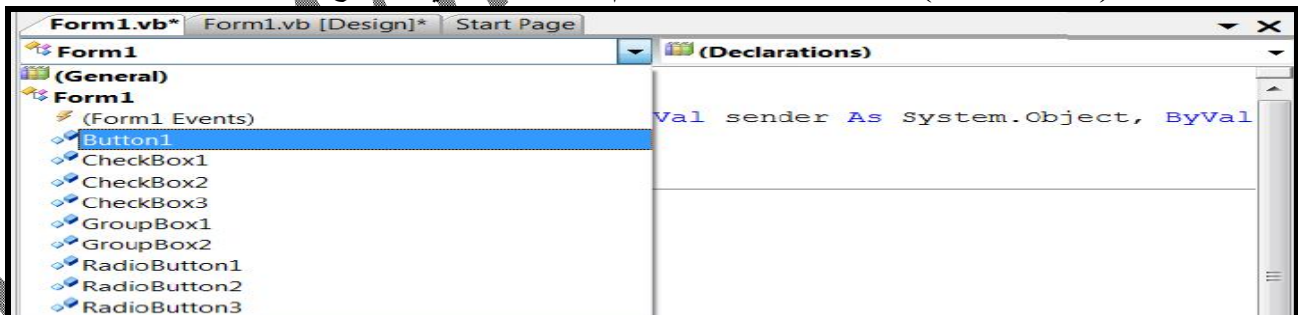
تشير الأرقام في نافذة معالج الحدث *Event Handler* إلى :

- ١- اسم الإجراء مكون من (اسم الكائن واسم الحدث) .
- ٢- سطر نهاية الإجراء .
- ٣- المسبب في استدعاء الإجراء .
- ٤- يكتب الكود بين السطرين الذي ينفذ عند استدعاء الإجراء بعد وقوع الحدث .
- ٥- سطر الإعلان عن التصنيف (*Form1*) .
- ٦- سطر نهاية التصنيف (*Class*) .

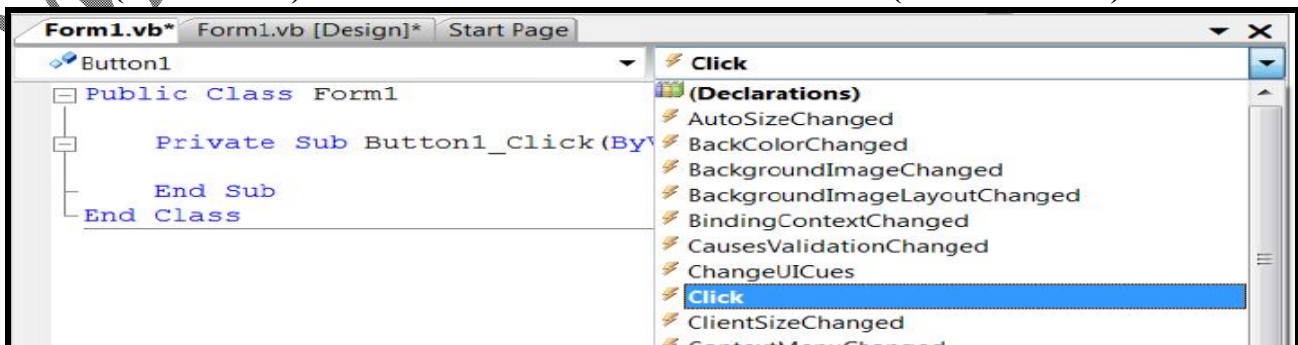
* مكونات نافذة معالج الحدث Code Window



- ١- القائمة (*Class Name*) تعرض أسماء أدوات التحكم الموضوعه علي النموذج *Form* .



- ٢- القائمة (*Method Name*) تعرض الأحداث الخاصة بالعنصر المختار من القائمة (*Class Name*) .



معلومات عامة

الخاصية

- هي التي تحدد شكل وسمه الكائن مثل الخاصية (*Text - Name*) .
- كل كائن له مجموعة من الخصائص يمكن تغييرها (ضبطها) سواء من نافذة الخصائص أو تغييرها برمجيا في نافذة الكود (*Code Window*) .
- يمكن ضبط الخصائص برمجيا في لغة *VB.Net* بالصيغة التالية :

Control Name . Properties = Value

القيمة = الخاصية . إسم أداة التحكم (الكائن)

- برنامج لتغيير النص المكتوب علي الزر *Button1* إلي كلمة جمهورية مصر العربية .

Button1.Text = "جمهورية مصر العربيت"

- برنامج لتغيير لون النص المكتوب علي الزر *Button1* إلي اللون الأحمر .

Button1.ForeColor = Color.Red

- برنامج لتغيير لون خلفية الزر *Button1* إلي اللون الأزرق .

Button1.BackColor = Color.Blue

- برنامج لجعل الزر *Button1* غير متاح .

Button1.Enabled = False

- برنامج لإخفاء الزر *Label1* .

Button1.Visible = False

- كود لتغيير نوع الخط إلي *Arial* وحجم الخط إلي ٣٠ لأداة *Label1*

Label1.Font = New Font ("Arial", 30)

- كود لجعل أداة *Textbox1* متعددة الأسطر .

Textbox1.Multiline = True

إنتهت المراجعة النظرية لمنهج الـ *visual Basic . Net*

تمنياتي لكم بالنجاح والتفوق

Mr.Mohammad Zyan

للتواصل :

E_Mail : Mrmohammadzyan@yahoo.com