برمجة وتطبيقات الحاسوب

بلغة بيسك المرئي

إعداد

م. المهدي امحمد جبريل

بكالوريوس هندســـــــة إلكترونية

طالب ماجستير بالأكاديمية الليبية

Phone: **00218917154974**

E-Mail: Almahdi.jabriel@yahoo.com

كلمة المؤلف

بسم الله الرحمن الرحيم، والصلاة والسلام على خاتم الأنبياء والمرسلين سيدنا محمد وعلى آله وصحبه، أما بعد:

فإني أسأل الله عز وجل أن أكون وفقت في ترتيب هذه الورقات التي كانت ليست أكثر من تبيان بسيط لكل من هو مبتدئ في تعلم لغة فيجوال بيسك وراعيت ذلك بترتيب الفصول والأبواب، حتى تكون مسيرة الفهم وسهلة الاستيعاب.

وأسأل الله أن تكون فيها الفائدة والمعلومة الواضحة لكل مطلع

وأن تكون في ميزان حسناتنا...

اللهم اغفر لي ولوالدي وللمؤمنين يوم يقوم الحساب

اللهم اغفر لأمواتنا وشهدائنا وحينا أجمعين

والصلاة والسلام على رسول الله محمد خير الأنام

م. المهدي جبريل

مقدمـــــة:

ترجع بداية لغة بيسك إلى سنة (1960م) حيث تم تجهيزها على يد الأستاذين (Kurtz) و(Kenny) في كلية دارثموت.

سميت (BASIC) اشتقاقا من مجموعة الكلمات التالية:

Beginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode

أي لغة متعددة الأغراض للمبتدئين.

أنظمة (إصدارات) لغة بيسك (BASIC):

تعمل تحت نظام DOS:
- IBM BASIC
- QUICK BASIC
- GW BASIC
- TURBO BASIC

تعمل تحت Windows:
- VISUAL BASIC

جميع الأنظمة السابقة تعمل تحت نظام (DOS) ما عدا الإصدار (VISUAL BASIC) الذي يعمل تحت نظام نوافذ (Windows) بمختلف إصداراته.

لماذا فيجوال بيسك (VISUAL BASIC)؟:

نستخدم فيجوال بيسك للأسباب التالية:

1. إمكانية استخدام اللغة العربية في الإدخال والإخراج.
2. تعمل هذه اللغة تحت نظام نوافذ.
3. الإمكانات المتعددة لنظام نوافذ مع فيجوال بيسك والتي لا تتوفر في باقي الإصدارات التي تعمل تحت نظام (DOS).

الباب الأول: أساسيات البرمجة

ما هو البرنامج؟

هو عبارة عن مجموعة من الأوامر والتعليمات مكتوبة بإحدى لغات البرمجة والتي بتنفيذها يتحقق غرض معين.

ما هو المقصود بلغة البرمجة؟ وهل تختلف على البشر؟

تهدف لغة البرمجة إلى التخاطب مع الآلة أو الحاسوب، في حين لغات البشر تهدف على تخاطب البشر مع بعضهم البعض.

ما هي لغة الآلة؟

هي لغة برمجة لا يوجد بها إلا رمزان اثنان فقط هما الصفر والواحد (0، 1) وهي أول لغة برمجة في عالم الحاسوب.

كما أنها هي اللغة الأساسية التي يفهمها جهاز الحاسوب (لماذا؟؟؟) والسبب في ذلك هو إمكانية تمثيل هذين الرقمين تمثيلا فيزيائيا يمكن الآلة من التعرف عليه، ويمكن تمثيل ذلك بدائرة كهربائية مغلقة (تمثل 1)، أو دائرة كهربائية مفتوحة (تمثل 0).

ما هو المترجم؟

هو البرنامج المسئول عن تحويل برنامج مكتوب بلغة ما (مثل فيجوال بيسك) إلى لغة الآلة.

ويمكن تمثيله بالشكل التالي:

وظيفة المترجم

أنواع المترجمات:

يوجد هنالك نوعين من المترجمات هما:

1. المفسّر (Interpreter):

هو برنامج يقوم بترجمة البرنامج المكتوب بلغة ما إلى برنامج بلغة الآلة، وتتم عملية الترجمة فيه جملة جملة (ترجمة فورية).

2. المصرّف (Compiler):

هو برنامج يقوم بترجمة البرنامج المكتوب لغة ما إلى برنامج بلغة الآلة، وتتم عملية الترجمة فيه دفعة واحدة (ترجمة كلية).

ما هو نوع المترجم في لغة فيجوال بيسك؟

يصنف المترجم في لغة فيجوال بيسك من النوع المفسّر(Interpreter)، ويمكن ملاحظة ذلك عند حصول خطأ في كتابة برنامج في لغة فيجوال بيسك حيث وبشكل مباشر سيتغير لون الكتابة إلى اللون الأحمر مشيرا إلى وجود خطأ.

ما هي المهام التي يقوم بتنفيذها المترجم؟

يعمل المترجم على تنفيذ مهمتين أساسيتين هما:

1. ترجمة برنامج بلغة ما إلى لغة الآلة.
2. اكتشاف الأخطاء اللغوية في البرنامج وإعلام المستخدم بها.

أنواع الأخطاء:

تصنف الأخطاء في البرنامج إلى نوعين:

1. خطأ لغوي (Syntax error):

هو خطأ يحدث نتيجة لمخالفة قواعد اللغة، وعادة ما يتم اكتشاف من قبل البرنامج المترجم.

2. خطأ منطقي (Logical error):

هو خطأ يكتشفه البرنامج المترجم ولكن يكتشفه المستخدم (المبرمج) بتجربة البرنامج واختباره ببعض البيانات المتوقع نتائج معالجتها.

لغات البرمجة (Programming languages):

ما هي لغة البرمجة؟

هي لغة التواصل مع جهاز الحاسوب ووسيط للتخاطب بين البشر والآلة، وتتكون عادة من عدد محدود من المفردات والقواعد اللغوية.

أمثلة على لغات البرمجة:

- لغة بيسك (BASIC).
- لغة فورتران (FORTRAN).
- لغة باسكال (PASCAL).
- لغة سي (C).
- لغة (C++) والتي هي تطوير للغة سي.
- لغة دلفي (تطوير للغة باسكال).
- لغة فيجوال بيسك (تطوير للغة بيسك).
- لغة جـــــافا، وغيرها الكثير من اللغات الأخرى...

ما هي أوجه الاختلاف بين الحاسوب والإنسان؟

يمتاز الإنسان عن الحاسوب بقدرة التفكير فهو الذي يفكر للحاسوب ويرشده إلى طريقة حل أي مسألة.

بينما يمتاز الحاسوب عن الإنسان بالسرعة، الدقة، الحساب، وعدم النسيان (التخزين الكبير للبيانات).

ملاحظات هامة:

- تحتاج جميع لغات البرمجة إلى برنامج مترجم (مفسّر أو مصرف) لترجمة برامجها إلى لغة الآلة ما عدا لغة الآلة.
- في حالة وجود خطأ لغوي في البرنامج فإن تنفيذه يتوقف إلى أن يتم إصلاح الخطأ الموجود فيه عكس الخطأ المنطقي الذي إن وجد عادة ما يتم قبول تنفيذ البرنامج.

خطوات إعداد برنامج:

متى نحتاج إلى الحاسوب؟أو ما هي المسائل التي نحتاج في حلها إلى الحاسوب؟ أو متى نحتاج إلى كتابة برنامج؟

جهاز الحاسوب آلة مفيدة خاصة في المجالات التالية:

1. الحسابات المطوّلة والمتكررة والتي لا تحتمل الخطأ.
2. معالجة البيانات الضخمة مثل ترتيبها أو البحث عن معلومة معينة فيها...الخ.
3. الاتصالات ونقل المعلومـــــــات.
4. تخزين البيـــــانات.

الخطوات الواجب إتباعها عند إنجاز عمل برمجي ما:

1. تحديد المسألة: يقصد بها تحديد معطيات البرنامج والناتج المستهدف.
2. تصميم الحل: أي كتابة خوارزمية الحل وهي الخطوات والعمليات التي بإتباعها نصل إلى الحل المطلوب.
3. كتابة البرنامج بإحدى لغات البرمجة: هي عبارة عن ترجمة الخوارزمية المكتوبة بلغة المبرمج (العربية مثلا) إلى لغة يفهمها الحاسوب مثل (فيجوال بيسك).
4. تجربة البرنامج وتنفيذه على جهاز الحاسوب.
5. في حالة نجاح البرنامج نتوقف، وإلا (أي فشل تنفيذ البرنامج) فلا بد من مراجعة وتصحيح الخطأ في الخطوات السابقة من بدايتها.

الخوارزميـــــــــــات (Algorithms)

ما هي الخوارزمية؟

هي عبارة عن مجموعة من الخطوات التي تؤدي عند تنفيذها إلى الحل المطلوب.

لماذا سميت الخوارزمية بهذا الاسم؟

سميت الخوارزمية بهذا الاسم نسبة للعالم محمد بن موسى الخوارزمي الذي عاش في القرن التاسع ونبغ في علم الجبر.

ملاحظات هامة عن الخوارزميات:

عند إعداد خوارزمية يجب الانتباه إلى التالي:

1. لكل خوارزمية بداية (ابـــدأ) ونهاية (تـــوقف).
2. نستخدم في الخوارزميات الأمر (اقـــــرأ) لإدخال المعطيات.
3. نستخدم في الخوارزميات الأمر (أكتب) لإخراج المطلوب حسابه.
4. نستخدم في الخوارزميات الرمز (*) كعلامة للضرب، والرمز (/) كعلامة قسمة.

مثال ① : أكتب خوارزمية لقراءة عدد وطباعة تربيعه؟

الحل:

المعطيات: العدد (أ).

المطلوب: طباعة تربيعه (ب)، حيث: ب = أ × أ.

الخوارزمية المطلوبة:

1. ابدأ.
2. اقـرأ العدد (أ).
3. احسب: ب = أ * أ.
4. أكتب (ب).
5. توقف.

مثال ② : أكتب خوارزمية لقراءة عددين وإيجاد حاصل جمعهما؟

الحل:

المعطيات: العدد الأول (س)، العدد الثاني (ص).

المطلوب: حساب المجموع (م)، حيث: م = س + ص.

الخوارزمية المطلوبة:

1. ابـدأ.
2. اقرأ العدد الأول (س).
3. اقرأ العدد الثاني (ص).
4. احسب: م = س + ص.
5. أكتب (م).
6. توقف.

المخطط الانسيابي (Flow-chart)

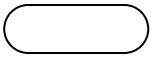
ما هو المخطط الانسيابي؟

هو عبارة عن مجموعة من الأشكال الهندسية تربط بينها أسهم لتوضيح عمل الخوارزمية.

ويمتاز هذا المخطط بوضوع أكثر من الخوارزمية وخاصة في توضيح عمل الحلقات والشروط المنطقية.

الأشكال الأساسية المستخدمة في المخطط الانسيابي:

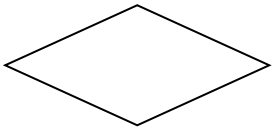
1. الشكل البيضاوي: يستخدم للبداية (ابدأ) والنهاية (توقف).



2. المستطيل: يستخدم للعمليات الحاسبية وتعيين قيم المتغيرات.

3. متوازي الأضلاع: للقراءة والكتابة (أو للإدخال والإخراج).

4. شكل المعين: يستخدم للمقارنات والعمليات المنطقية.



5. الدائرة: تستخدم لتوضيح نقاط توصيل المخطط الانسيابي، وغالبا في حالة المخططات ذات الأحجام الكبيرة.

6. السهم: يستخدم للتوصيل فيما بين الأشكال الهندسية السابقة وتوضيح مسار المخطط.

مثال ③ : أكتب خوارزمية الحل وارسم المخطط الانسيابي لقراءة طول وعرض مستطيل وحساب مساحته؟

الحل:

المعطيات:

طول المستطيل (ط)، عرض المستطيل (ع).

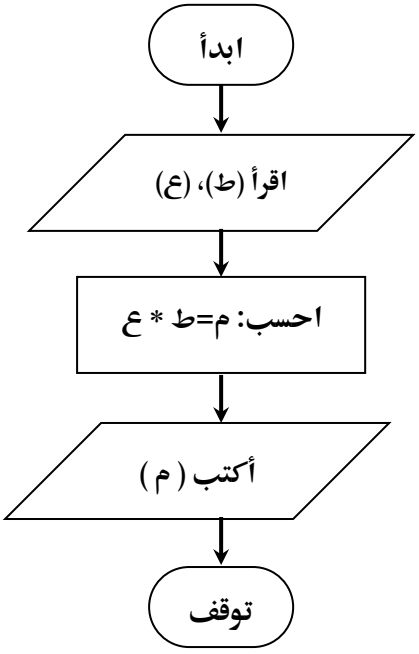
المطلوب:

حساب مساحة المستطيل (م)، حيث: م = ط × ع.

الخوارزمية المطلوبة:

1. ابدأ.
2. اقرأ (ط) و (ع).
3. احسب: م = ط * ع.
4. أكتب قيمة (م).
5. توقف.

المخطط الانسيابي:



ملاحظات هامة:

1. استخراج المخطط الانسيابي يتم من خوارزمية الحل، أي حتى لو كان المطلوب في المثال أرسم المخطط الانسيابي فقط فإنه يفضل لتلافي الوقوع في الخطأ أن يتم عمل الخوارزمية أولا.
2. يمكن الوصول إلى خوارزمية الحل إذا توفر المخطط الانسيابي.

مثال ④ : أكتب خوارزمية الحل وارسم المخطط الانسيابي لحساب متوسط أربعة أعداد؟

الحل:

المعطيات:

أربعة أعداد (ل)، (ك)، (ع)، (ط).

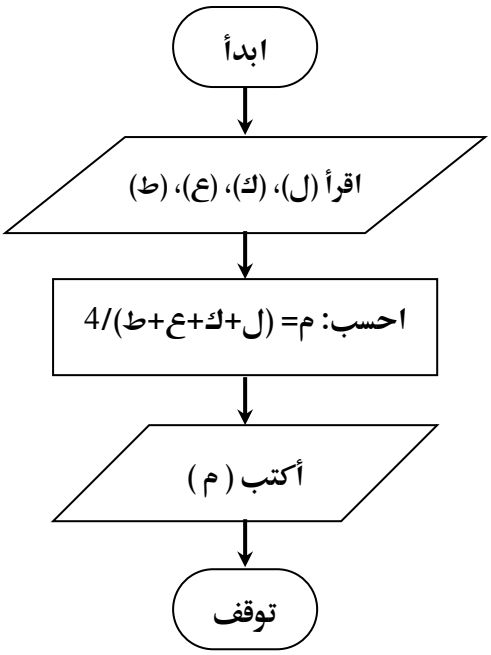
المطلوب:

حساب المتوسط (م)، حيث: م = (ل + ك + ع + ط)/4.

الخوارزمية المطلوبة:

1. ابدأ.
2. اقرأ (ل)، (ك)، (ع)، (ط).
3. احسب : م = (ل + ك + ع + ط)/4.
4. أكتب قيمة (م).
5. توقف.

المخطط الانسيابي:



مثال ⑤ : أكتب خوارزمية الحل وارسم المخطط الانسيابي لحساب الضريبة السنوية، علما بأن الضريبة (15٪) من الدخل السنوي إذا زاد هذا الدخل عن 4000 دينار، وإلا فإنه يعفى من الضريبة؟

الحل:

المعطيات: الدخل السنوي (د).

المطلوب: حساب الضريبة (ض)، حيث:

ض = 15٪ × د عندما (د >= 4000 دينار).

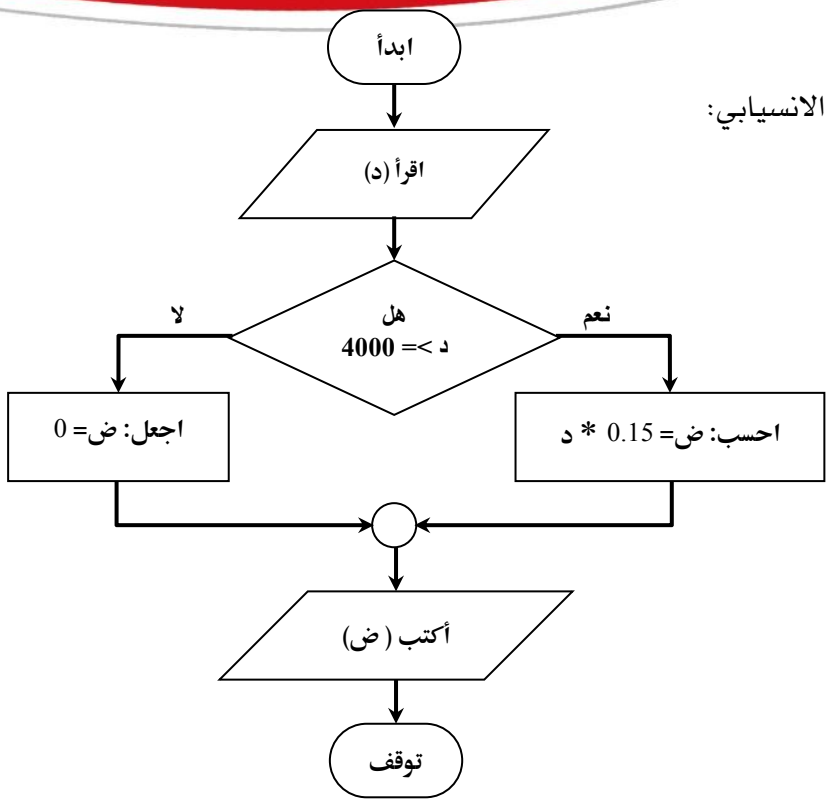
ض = 0 عندما (د < 4000 دينار).

نلاحظ هنا أن هنالك قيمة للضريبة (ض) وسيتم اختيار إحدى هاتين القيمتين بناءا على قيمة الدخل السنوي (د) إذا زاد عن 4000 دينار أو قل عنها.

الخوارزمية المطلوبة:

1. ابدأ.
2. اقرأ (د).
3. إذا كانت (د >= 4000) فاحسب: ض = 0.15 * د ، وإلا اجعل ض = 0.
4. أكتب قيمة (ض).
5. توقف.

المخطط الانسيابي:



مثال ⑥ : أكتب خوارزمية الحل وارسم المخطط الانسيابي لقراءة درجات طالب في عدد من المواد الدراسية وإيجاد مجموعة علما بأن عدد المواد تتم قراءته في الخوارزمية؟

الحل:

المعطيات: عدد المواد (ن)، درجات المواد (د).

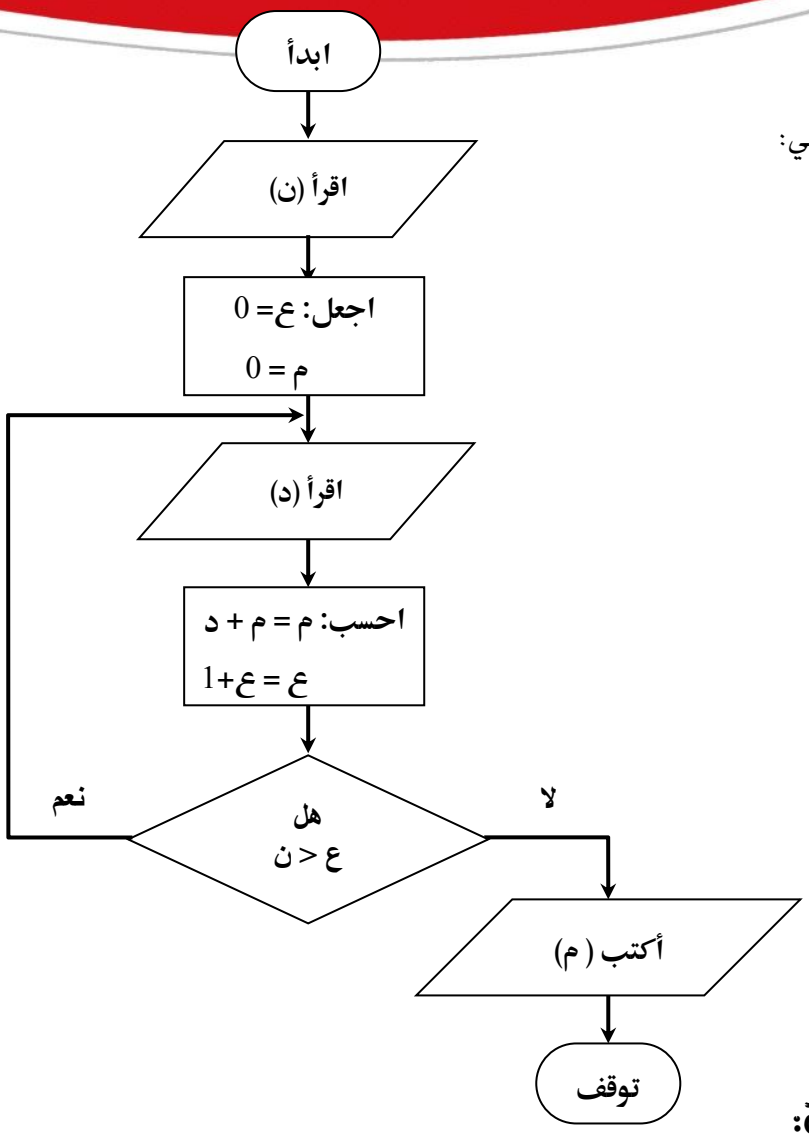
المطلوب: حساب المجموع (م).

ملاحظة: نلاحظ هنا أن عدد المواد غير معلوم، لذلك تم استخدام متغير (ن) ليحسب عدد تلك المواد ، وفي مثل هذه الحالات يجب استخدام متغير (عداد ع) يقوم بمتابعة عدد الدرجات التي تم قراءتها وتتوقف القراءة بعد أن تصبح (ع = ن).

الخوارزمية المطلوبة:

1. ابدأ.
2. اقرأ عدد المواد (ن).
3. اجعل (ع = 0) عداد المواد.
4. اجعل (م = 0) المجموع.
5. اقرأ الدرجة (د).
6. احسب: م = م + د.
7. احسب: ع = ع + 1.
8. إذا كانت (ع < ن) ارجع إلى الخطوة (5).
9. اكتب المجموع (م).
10. توقف.

المخطط الانسيابي:



ملاحظة هامة:

تم استخدام متغير وسميناه باسم عداد، وبالتالي فإن العداد هو متغير يتم تحديد قيمة بداية وقيمة زيادة وقيمة يتوقف عندها، والهدف الرئيسي منه هو تكرار عملية معينة عدد من المرات لتلافي الحجم الكبير والغير مقبول في الخوارزميات والمخططات والبرامج.

مثال ⑦ : ارسم المخطط الانسيابي لقراءة قيمة وطباعة كلمة "موجب" إذا كانت القيمة أكبر من أو تساوي الصفر، وكلمة سالب إذا كانت القيمة أصغر من الصفر؟

الحل:

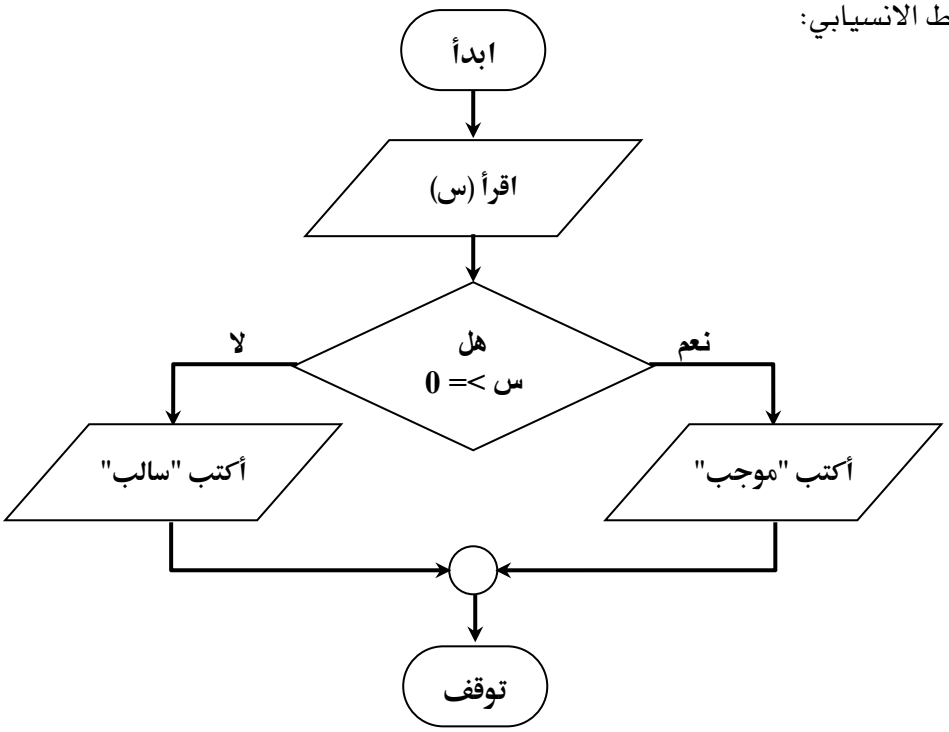
المعطيات: قيمة عددية (س).

المطلوب: طباعة كلمة (موجب) إذا كانت (س >=0) وكلمة (سالب) إذا كانت (س<0).

الخوارزمية المطلوبة:

1. ابدأ.
2. اقرأ (س).
3. إذا كانت (س >=0) اكتب "موجب"، وإلا اكتب "سالب".
4. توقف.

المخطط الانسيابي:



مثال ⑧ : أكتب خوارزمية الحل وارسم المخطط الانسيابي لطباعة الأعداد المحصورة بين (0 - 10)؟

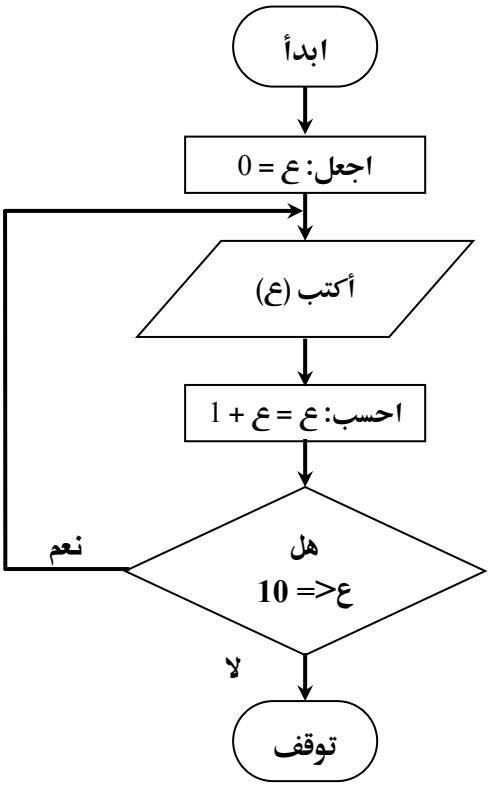
الحل:

المعطيات: الأعداد (0 -10).

المطلوب: طباعة هذه الأعداد تصاعديا.

الخوارزمية المطلوبة:

1. ابدأ.
2. اجعل (ع = 0).
3. أكتب (ع).
4. احسب: ع = ع + 1.
5. إذا كانت (ع <=10) ارجع إلى الخطوة (3).
6. توقف.

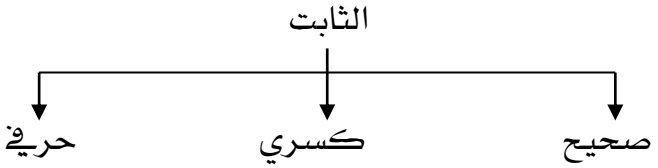


الباب الثاني: أساسيات لغة بيسك

يتناول هذا الباب بعض أساسيات لغة البرمجة فيجوال بيسك، والتي من أهمها أنواع الثوابت والمتغيرات بالإضافة إلى بعض الأوامر والإيعازات الأساسية في لغة بيسك.

ما هو الثابت (Constant)؟:

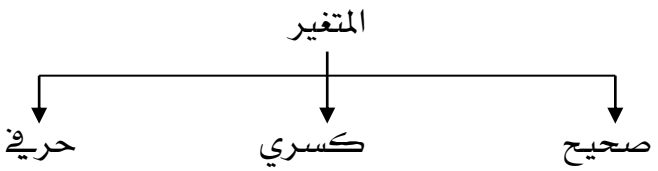
هو القيمة التي تبقى كما هي (لا تتغير) خلال عملية أو أثناء تنفيذ البرنامج، وهو إما أن يكون ثابت عددي مثل: 123 أو ثابت حرفي مثل: "علي".



- الثابت الصحيح: هو القيمة العددية التي لا تحتوى على علامة عشرية (أي لا تحتوي على جزء كسري) مثل: 23، 81 -، 45 ... الخ.
- الثابت الكسري: هو القيمة العددية التي تحتوي على علامة عشرية، مثل: 3.2، 4.0 -، 7.9 ... الخ.
- الثابت الحرفي: هو القيمة الحرفية التي تبقى كما هي خلال عملية ما، ويوضع بين علامتي التنصيص " "، مثل: "محمد"، "Ahmed"، "40"، "3.14" ... الخ.

ما هو المتغير (*Variable*)؟:

هو اسم موقع في ذاكرة الحاسوب لتخزين قيمة ما.



- المتغير الصحيح: هو المتغير الذي يتعين له قيمة ثابت عددي صحيح.
- المتغير الكسري: هو المتغير الذي يتعين له قيمة ثابت عددي كسري.
- المتغير الحرفي: هو المتغير الذي يتعين له قيمة ثابت حرفي.

طرق تحديد نوع المتغير في لغة فيجوال بيسك:

توجد هنالك طريقتان:

أ. الطريقة الأولى: بوضع علامة مميزة على يمين اسم المتغير حيث يستطيع المترجم تمييز نوع المتغير كما في الجدول التالي:

النوع	الرمز
الصحيح (Integer)	%
الصحيح الطويل (Long)	&
الكسري المفرد (Single)	!
الكسري المضاعف (Double)	#
العملة (Currency)	@
النضيد (String)	$

مثل:

Count% -متغير صحيح.

K% -متغير صحيح.

V# -متغير كسري مضاعف.

tax! – متغير كسري مفرد.

name$ - متغير حرفي.

address$ - متغير حرفي.

ب. الطريقة الثانية: بتحديد نوع المتغير في بداية البرنامج أو قبل استخدام المتغير، وكما يلي:

Dim *var* as *Type*

حيث:

Dim: كلمة محجوز اختصارا للكلمة (Dimension).

var: اسم المتغير، مثل: x, y, k ... الخ.

Type: نوع المتغير (صحيح (Integer)، كسري (Single)، حرفي (String) ... الخ).

مثل:

Dim x **as** Integer (الإعلان عن متغير من نوع صحيح)

Dim Name **as** String (الإعلان عن متغير من نوع الحرفي)

Dim tax **as** Single (الإعلان عن متغير من نوع الكسري)

ملاحظة هامة:

في حالة لم يتم تحديد نوع المتغير بإحدى الطريقتين السابقتين سيتم استخدام النوع المرن لهذا المتغير (*variant*) والذي يصلح لتمثيل أي نوع من أنواع المتغيرات الستة السابقة بحيث يتم تحديد نوع هذا المتغير أثناء تنفيذ البرنامج بنوع أول قيمة يتم وضعها أو تخصيصها للمتغير.

شروط تسمية المتغير:

يشترط لقبول اسم المتغير المستخدم في البرنامج أن يكون كالتالي:

1. أن يبدأ اسم المتغير بحرف وليس برقم أو رمز.
2. أن لا يكون الاسم أحد الكلمات المحجوزة في لغة فيجوال بيسك مثل: (*If, print, end, sub* ... الخ).
3. أن لا يحتوي الاسم على رمز من رموز العمليات الرياضية أو بعض الرموز الخاصة مثل: (\ + ? -... الخ) باستثناء الرمز (_).

العبارات الحسابية:

يمكن أن تتكون العبارة الحسابية من ثابت واحد أو متغير واحد أو مجموعة من المتغيرات والثوابت والرموز الحسابية، مثل:

A + B
32/2 + 10/2
5/C + 3 + B

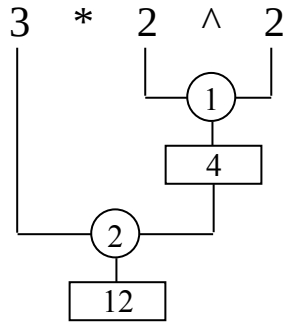
أسبقية (أولوية) تنفيذ العمليات الحسابية:-

جهاز الحاسوب يلتزم بالأولوية التالية في تنفيذه للعمليات الحسابية:

1. الأقواس.
2. الأس.
3. الضرب والقسمة (من اليسار إلى اليمين).
4. الجمع والطرح (من اليسار إلى اليمين).

مثال①: ما هو ناتج تنفيذ العبارة الحسابية التالية: 3 * 2^2 ؟

الحل:

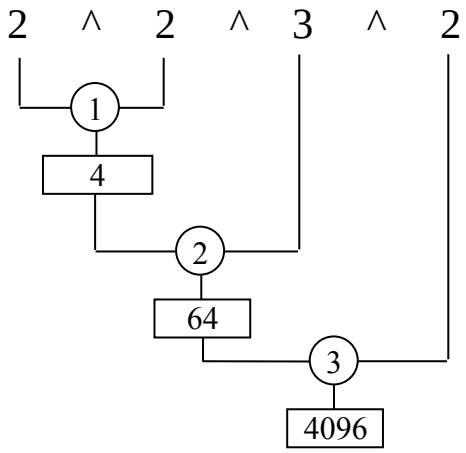


ملاحظة:

رمز الدائرة يشير إلى ترتيب تنفيذ العملية، والمستطيل يشير إلى قيمة تلك العملية.

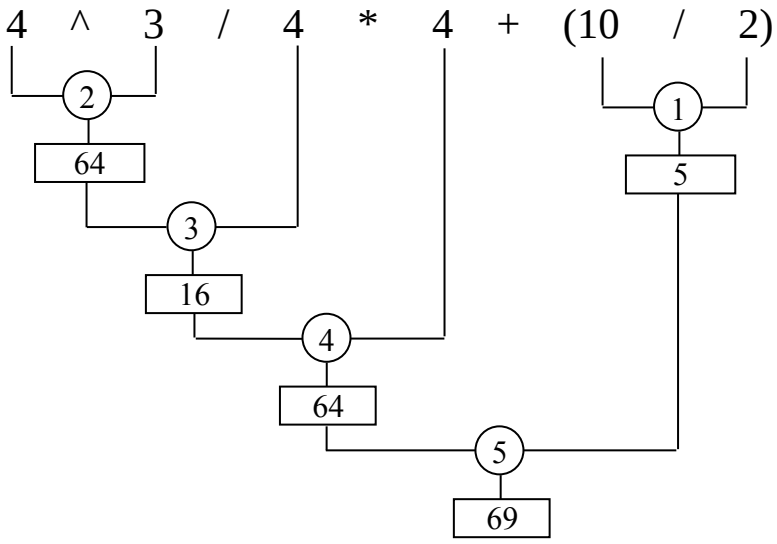
مثال②: تتبع تنفيذ العبارة الحسابية التالية: 2 ^ 3 ^ 2 ^ 2 ؟

الحل:



مثال③: تتبع تنفيذ العبارة الحسابية التالية: (10/2) + 4*4/4^3 ؟

الحل:



الأوامر الأساسية في لغة فيجوال بيسك

1. جملة التعيين (Let):

هي أمر من أوامر لغة بيسك بمعنى (اجعل)، وهو أمر اختياري وليس ضروري، ويعمل على تعيين قيمة أو ناتج عبارة إلى متغير ما، ويكون بالشكل العام التالي:

Let اسم المتغير = قيمة أو متغير

مثل:

```
Let x = 10 / 2
Let y = 15
Let B = (x – y) * 3
```

2. جملة النهاية (التوقف) (End):

تعمل هذه الجملة على إيقاف تنفيذ البرنامج بحيث لن يتم تنفيذ أي جملة بعدها، وهي تعمل عمل الأمر (توقف) في الخوارزمية.

3. جملة الطباعة (الكتابة) (Print):

تعمل هذه الجملة على طباعة أو إخراج نواتج تنفيذ البرنامج على الشاشة، وتكون بالشكل العام التالي:

Print قيمة أو متغير أو عبارة حسابية

مثل:

```
Print 10
Print x / 2 + 5
Print "الصف الثالث ثانوي"
```

كما أنه يمكن أن يطبع أكثر من قيمة أو متغير بأمر واحد فقط وذلك باستخدام فاصلة عادية (,) أو منقوطة (;)، وإذا انتهى هذا الأمر بفاصلة عادية (,) أو منقوطة (;) فإن ذلك يعني أن أمر الطباعة التالي سيكون في نفس السطر، بحيث إذا كانت الفاصلة عادية ستكون الطباعة متباعدة، وإذا كانت الفاصلة منقوطة فإن الطباعة ستكون متلاصقة، مثل:

في حالة فاصلة عادية

```
Print 1,2,3,4,5
```

سيكون ناتج الطباعة: 1 2 3 4 5

في حالة فاصلة منقوطة

```
Print 1;2;3;4;5
```

سيكون ناتج الطباعة: 1 2 3 4 5

مثال ④ : أكتب برنامج لحساب مساحة قطعة أرض على شكل مستطيل طوله (25.6 متر) وعرضه (34.5 متر)؟

الحل:

المعطيات: الطول (L) = 25.6 ، العرض (W) = 34.5 .

المطلوب: حساب مساحة قطعة الأرض (A)، حيث:

$$A = L * W$$

البرنامج المطلوب:

```
Let L = 25.6
Let W = 34.5
Let A = L * W
Print A , "مساحة قطعة الأرض"
End
```

4. توثيق البرنامج (Remarks):

يهدف توثيق البرنامج إلى كتابة ملاحظات عن:

1. الغرض من البرنامج.
2. كاتب البرنامج.
3. شرح متغيرات البرنامج.
4. شرح طريقة عمل البرنامج.
5. تاريخ كتابة البرنامج.
6. معلومات أخرى مختلفة.

يمكن استخدام التوثيق بإحدى طريقتين:

- باستخدام الأمر (Rem)، مثل:

```
Rem هذا البرنامج لحساب مساحة قطعة أرض مستطيلة الشكل
```

- باستخدام علامة التنصيص المفرد مثل:

```
' هذا البرنامج لحساب مساحة قطعة أرض مستطيلة الشكل
```

ملاحظة هامة:

كلا عبارتي التوثيق لا يتم تنفيذه من قبل الحاسوب، بل هي بمثابة الجمل الميتة، فهي تفيد فقط في معرفة شيء ما حول البرنامج أو عمله.

مثال ⑤ : أكتب برنامج لحساب عدد الأيام في 25 سنة علما بأن السنة 365.5 يوما؟

الحل:

المعطيات: عدد السنوات (y) = 25.

المطلوب: حساب عدد الأيام في السنوات، حيث:

$$d = y * 365.5$$

البرنامج المطلوب:

```
Rem هذا البرنامج لحساب عدد الأيام
Let y = 25
d = y * 365.5
Print d , "عدد الأيام"
End
```

مثال ⑥ : أكتب برنامج لحساب حجم كرة نصف قطرها (34سم) علما بأن:

حجم الكرة = (4 ط نق^3 ÷ 3)، وأن: ط = 3.14 تقريباً.

الحل:

المعطيات: نصف القطر (R) = 34 ، ط (P) = 3.14 .

المطلوب: حساب حجم الكرة (V)، حيث:

$$V = 4 * P * R^3/3$$

البرنامج المطلوب:

```
Rem هذا البرنامج لحساب حجم الكرة
R = 34
P = 3.14
V = 4 * P * R^3/3
Print V , "حجم الكرة"
End
```

الباب الثالث: تنفيذ برنامج بيسك المرئي

سيتم التعرض في هذا الباب إلى بيئة بيسك والمكونات الأساسية في نوافذه والوظائف الأساسية المتضمنة لها.

تشغيل بيئة بيسك المرئي:

يتم تشغيل برنامج بيسك المرئي كالتالي:

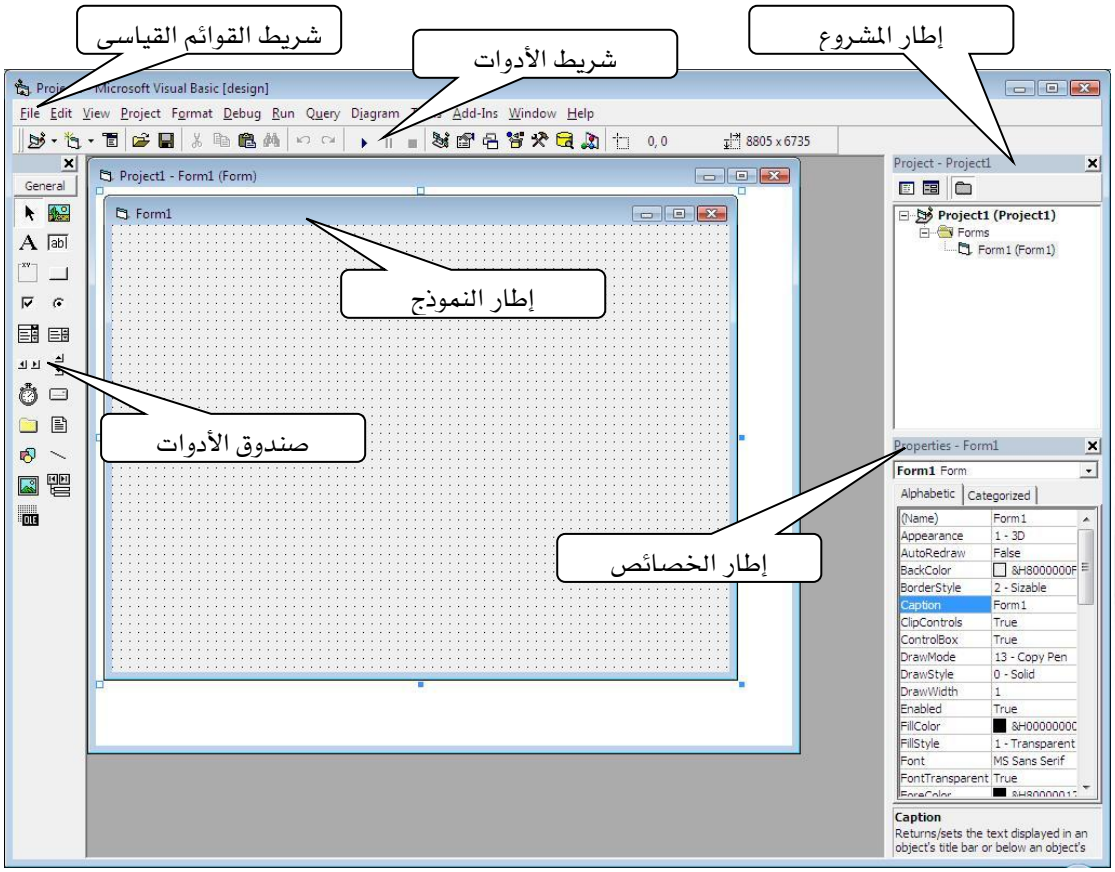
1. من قائمة ابدأ :

قائمة ابدأ ☜ البرامج ☜ Microsoft visual Basic ☜ Microsoft visual Basic 5.0

2. من سطح المكتب:

بالنقر المزدوج على أيقونة الاختصار الخاصة بفيجوال بيسك إن وجدت على سطح المكتب.

عندها تظهر النافذة المبينة.



شكل يبين النافذة الرئيسية لبرنامج فيجوال بيسك

مكونات نافذة فيجوال بيسك:

تتكون هذه النافذة من الأجزاء التالية:

- صندوق الأدوات (Tool box).
- إطار النموذج (Form window).
- إطار الخصائص (Properties window).
- شريط القوائم القياسي (Main menu).
- شريط الأدوات (Tools bar).
- إطار المشروع (Project window).

1. صندوق الأدوات (Tool box):

يحتوي هذا الصندوق على مجموعة من أزرار التحكم التي تساعد في تصميم واجهة المستخدم أو واجهة البرنامج.

من أهم هذه الأزرار:

- الزر (Label) A :

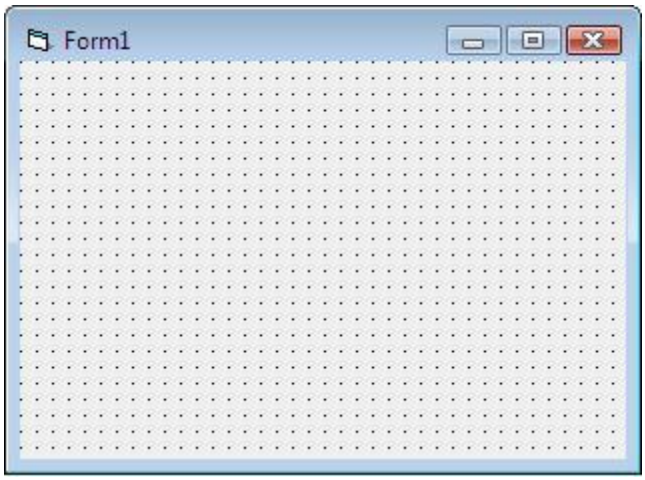
يستخدم في كتابة توضيح المخرجات على نافذة البرنامج.

- الزر (Text Box) abl :

يتم بواسطته رسم إطار مستطيل الشكل يوضع فيه نص معين في الإدخال أو في الإخراج.

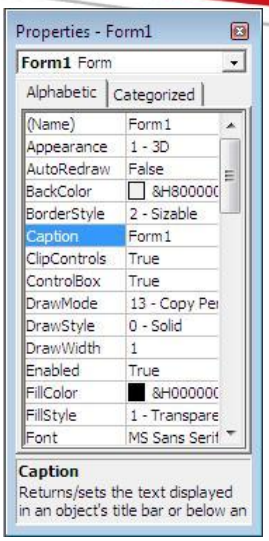
- الزر (Command Button) :

يستخدم لاستقبال الأوامر من المستخدم.



2. إطار النموذج (Form window):

وهو يمثل الواجهة التي يتم فيها التفاعل بين البرنامج والمستخدم، حيث تتم عملية إدخال البيانات وإخراجها في هذه النافذة ولذلك تسمى بواجهة المستخدم (User interface).

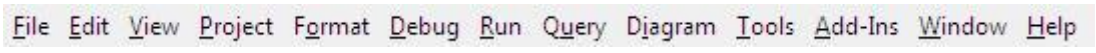


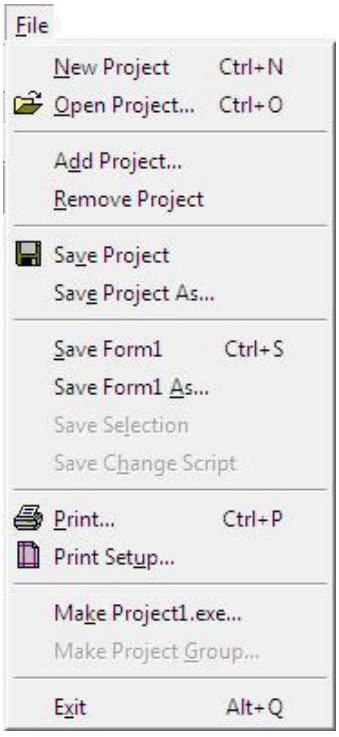
3. إطار الخصائص (Properties window):

يتم عن طريق هذا الإطار تحديد خصائص أي أداة يتم رسمها على إطار النموذج، مثل تغيير اسم الأداة، اللون، الحجم ... الخ، ويوجد عادة يمين أسفل الشاشة.

4. شريط القوائم القياسية (Main menus):

نجد في هذا الشريط الأوامر التي نحتاج لها في إعداد وتنفيذ برنامج بيسك المرئي، ومنها:





أ. قائمة ملف (File):

وتحتوي على الأوامر الهامة التالية:

❖ الأمر (New project) للبدء في مشروع أو برنامج أو تطبيق جديد.

- الأمر (Open project) لفتح مشروع مخزن.
- الأمر (Save Form) لتخزين النافذة فقط.
- الأمر (Save Form as) لتخزين النافذة فقط باسم آخر أو في موقع آخر.
- الأمر (Save project) لتخزين المشروع بالكامل (النافذة والبرنامج).
- الأمر (Save project as) لتخزين المشروع بالكامل باسم آخر أو في موقع آخر.

ب. قائمة تحرير (Edit):

ومن أهم أوامرها:

- أمر القطع (القص) (Cut).
- أمر النسخ (Copy).
- أمر اللصق (Paste).
- أمر الإلغاء (الحذف) (Delete).
- أمر البحث (Find).

ج. قائمة تنفيذ (Run):

نحتاج لهذه القائمة عند إجراء تنفيذ للبرنامج، وتشمل:

- الأمر (Start) لبدأ تنفيذ البرنامج.
- الأمر (End) لإنهاء تنفيذ البرنامج.
- الأمر (Restart) لإعادة تنفيذ البرنامج.

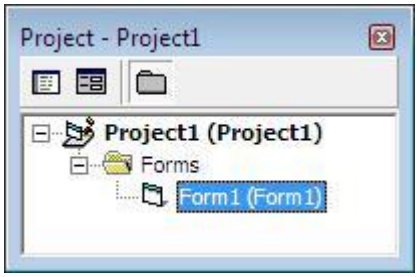
بالإضافة إلى مجموعة أخرى من القوائم تحتوي وظائف مختلفة.

5. شريط الأدوات (Tool bar):

يحتوي هذا الشريط على مجموعة من الأزرار المختصرة للوصول إلى الأوامر كثيرة الاستخدام، مثل:

- زر التخزين (الحفظ).
- زر بداية تنفيذ البرنامج.
- زر إنهاء تنفيذ البرنامج.
- زر النسخ.
- زر القص.
- زر اللصق.

6. إطار المشروع (Project window):



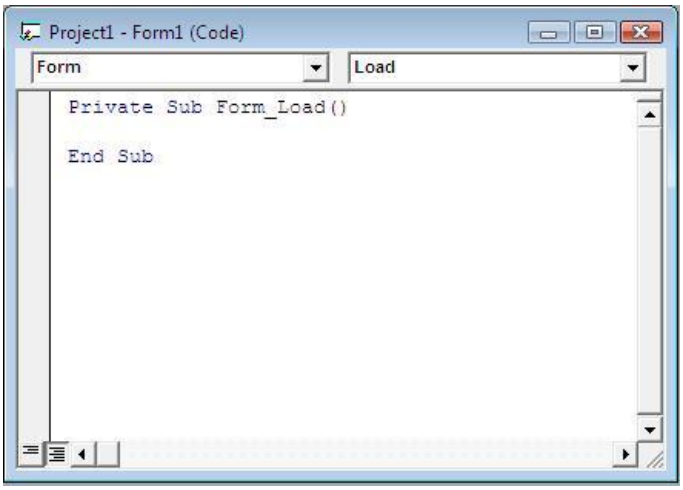
توفر هذه النافذة زرين هما:

- زر عرض البرنامج (View code): لعرض شفرة البرنامج.
- زر عرض النافذة (View object): لعرض نافذة أو نموذج البرنامج.

كيف يتم تنفيذ البرنامج على جهاز الحاسوب؟:

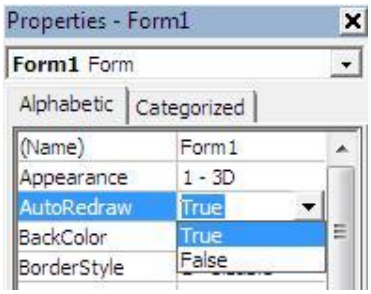
قبل تنفيذ البرنامج يجب عمل التالي:

1. كتابة البرنامج بالكامل على ورق.
2. تشغيل جهاز الحاسوب ثم نشغل برنامج فيجوال بيسك ونفتح ملف جديد.
3. ننقر نقرا مزدوجا على نافذة النموذج، عندها تظهر نافذة تحرير البرنامج الخاصة بكتابة البرنامج أو التعديل فيه، كما في الصورة ص (87).



4. نقل البرنامج من الورقة إلى النافذة المذكورة.

ملاحظات هامة:



1. توجد خاصية في مربع الخصائص باسم (Auto Redraw) نقوم بتغييرها من (False) إلى (True) حتى يتمكن البرنامج من الكتابة على النافذة مباشرة وذلك فقط في حالة وجود أمر الطباعة (Print) في البرنامج.
2. لا حاجة لكتابة الأمر (End) لأنه موجود أصلا بصيغة (End sub).
3. لتنفيذ البرنامج ننقر الزر (Run) من قائمة التنفيذ أو شريط الأدوات، أو نضغط المفتاح (F5) من لوحة المفاتيح.
4. في حالة ما يراد تعديل أو تطوير البرنامج فإنه يجب إيقاف تنفيذه أولا.
5. في حالة وجود أخطاء إملائية أثناء كتابة البرنامج يفضل تصحيحها أولا قبل الشروع في التنفيذ أو الانتقال إلى جملة أخرى.

تخزين البرنامج:

تأتي هذه المرحلة في العادة بعد كتابة البرنامج، وتتم كما يلي:

1. النقر على القائمة (File) ونختار منها (Save project).
2. نكتب اسم النافذة المناسب ثم ننقر الزر (save).
3. نكتب اسم البرنامج المناسب.
4. النقر على الزر (save).

ملاحظات:

- يجب أن يتم إيقاف تنفيذ البرنامج قبل الشروع في التخزين.
- يفضل تسمية البرامج بأسماء ذات علاقة بالهدف منها ليسهل تذكرها.

امتدادات (ملحق) ملفات البرنامج:

بعد تخزين المشروع يتكون لدينا ملفين بالامتدادين التاليين:

- الامتداد (frm) يضاف إلى اسم ملف النموذج (النافذة).
- الامتداد (vbp) يضاف إلى اسم ملف البرنامج (الشفرة).

وإضافة هذين الامتدادين إلى ملفات المشروع تتم بشكل تلقائي من قبل برنامج فيجوال بيسك.

الباب الرابع: إدخال وإخراج البيانات

يهدف هذا الباب إلى تقديم بعض الأدوات التي تمكن المستخدم من إدخال وإخراج البيانات بشكل عام، وبالتالي يكون البرنامج ذا صفة عملية أكبر.

أولا: دالة الإدخال (Inputbox):

تعمل هذه الدالة على إدخال قيمة لأي متغير في البرنامج أثناء وقت تنفيذ البرنامج، وتمتلك الشكل العام التالي:

```
V = InputBox("رسالة مناسبة")
```

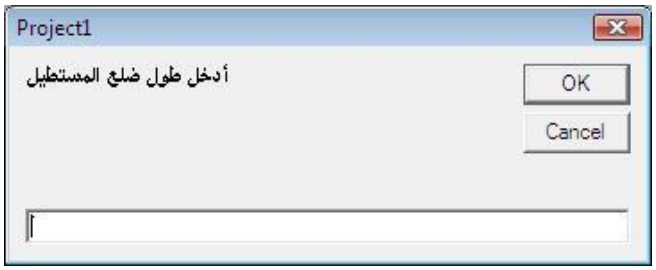
حيث:

V : متغير في البرنامج.

مثل:

```
L = InputBox("أدخل طول ضلع المستطيل")
```

ينتج عنها ظهور النافذة التالية:



ملاحظة هامة:

يجب الانتباه إلى تحديد نوع المتغير في هذه الدالة وإلا فإنه سيعامل كمتغير من النوع الحرفي (string) وليس المرن.

مثال ① : أكتب برنامج لقراءة عدد وطباعة تربيعه؟

الحل:

المعطيات: عدد (n).

المطلوب: طباعة تربيعه (a).

البرنامج المطلوب:

```
Private Sub Form_Load()
n% = InputBox("أدخل العدد")
a% = n% * n%
Print a% , "تربيع العدد هو"
End Sub
```

مثال ② : أكتب برنامج لقراءة طول وعرض مستطيل وحساب وطباعة مساحته ومحيطه؟

الحل:

المعطيات: طول المستطيل (x)، عرض المستطيل (y).

المطلوب: حساب المساحة (a)، والمحيط (c)، حيث:

$$a = x * y$$
$$c = (x + y) * 2$$

البرنامج المطلوب:

```
Private Sub Form_Load()
x! = InputBox("أدخل الطول")
y! = InputBox("أدخل العرض")
a! = x! * y!
c! = (x! + y!) * 2
Print a! , "مساحة المستطيل"
Print c! , "محيط المستطيل"
End Sub
```

مثال ③ : أكتب برنامج لحساب قيمة الزكاة على مبلغ مالي علما بأن نسبة الزكاة هي (2.5٪)؟

الحل:

المعطيات: مبلغ مالي (x).

المطلوب: حساب قيمة الزكاة (y)، حيث:

$$y = x * 2.5 / 100$$

البرنامج المطلوب:

```
Private Sub Form_Load()
x! = InputBox("أدخل قيمة المبلغ المالي")
y! =  x! * 2.5 / 100
Print x! , "قيمة المبلغ المالي"
Print y! , "قيمة الزكاة على هذا المبلغ"
End Sub
```

ثانيا: مربع النص (Text Box):

يستخدم مربع النص كأداة إدخال أو إخراج لقيم متغيرات البرنامج، وينوب عن أمر الطباعة (Print) في البرنامج.

ملاحظات:

- يفضل دائما رسم مربع النص قبل كتابة اسمه في البرنامج.
- الصيغة العامة لاستخدام مربع النص في الإخراج هي كالتالي:

اسم المتغير = Text1.text

حيث:

Text1.text : الخاصية التي عن طريقها نستطيع الكتابة في مربع النص.

مثال ④ : أكتب برنامج لتحويل مسافة مقاسة بالأميال إلى الكيلومترات، علما بأن:

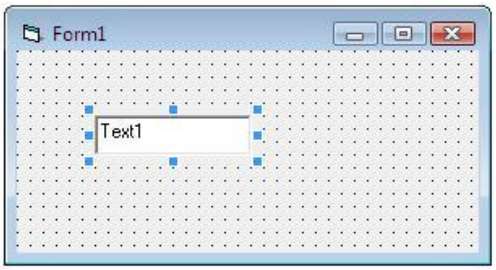
1 ميل = 1.6093 كيلو متر، وذلك باستخدام دالة الإدخال ومربع النص في الإخراج؟

الحل:

المعطيات: مسافة مقاسة بالأميال (Miles).

المطلوب: تحويل هذه المسافة إلى كيلومترات (Km).

أولا: تصميم نافذة البرنامج:



ثانيا: البرنامج المطلوب:

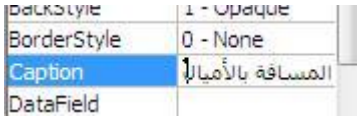
```
Private Sub Form_Load()
Miles! = InputBox("أدخل المسافة بالأميال")
km! =  Miles! * 1.6093
Text1.text = km!
End Sub
```

ثالثا: أداة الوصف (Label):

تستخدم هذه الأداة لكتابة وصف نصي لمخرجات أو مدخلات نافذة البرنامج، وهي أداة تكون مرافقة لمربع النص في الغالب.

والوصف في هذه الأداة يمكن كتابته بطريقتين:

1. عن طريق النموذج (من مربع الخصائص):



حيث يتم تحديد أداة الوصف بالنقر عليها أولا ثم الذهاب إلى مربع الخصائص أمام الخيار (caption) ونكتب التوضيح المناسب.

2. عن طريق البرنامج:

شرط أن يتم رسم الأداة أولا ثم نكتب السطر التالي في البرنامج:

```
Label1.caption  = "التوضيح المناسب"
```

ملاحظة:

الترقيم في أسماء الأدوات المدرجة على نافذة البرنامج يتحكم به برنامج فيجوال بيسك بشكل تلقائي، ويمكن تغيير أسماء هذه الأدوات بحسب ما يناسب المبرمج من الخاصية (Name) في مربع الخصائص.

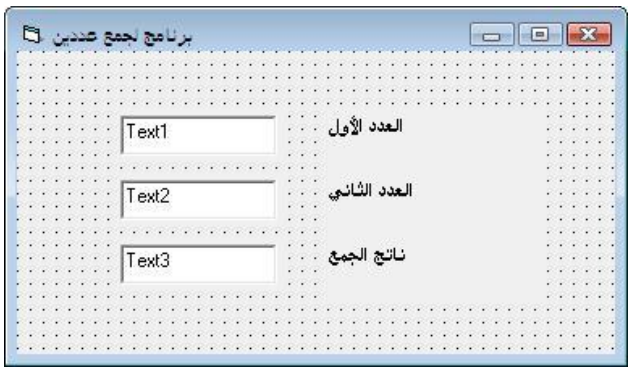
مثال ⑤ : أكتب برنامج لقراءة عددين وطباعة حاصل جمعها باستخدام مربع النص وكتابة توضيح المخرجات المناسب؟

الحل:

المعطيات: عددين (A)، (B).

المطلوب: إيجاد حاصل جمعهما (C).

أولا: تصميم نافذة البرنامج:



نلاحظ أنه تم كتابة التوضيح في أداة الوصف واسم النافذة أيضا عن طريقة الخاصية (Caption) من مربع الخصائص.

ثانيا : البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لإيجاد حاصل جمع عددين
A! = InputBox("أدخل العدد الأول")
B! = InputBox("أدخل العدد الثاني")
C! =  A! + B!
Text1.text = A!
Text2.text = B!
Text3.text = C!
End Sub
```

ونلاحظ ناتج تنفيذ البرنامج كما في النافذة التالية:

مثال ⑥ : تفاهم شريكان في مشروع اقتصادي بتقسيم العائد بينهما بنسبة 1 : 3، أكتب برنامج لقراءة العائد وتقسيمه بينهما ، استخدم دالة الإدخال ومربع النص وأداة الوصف؟

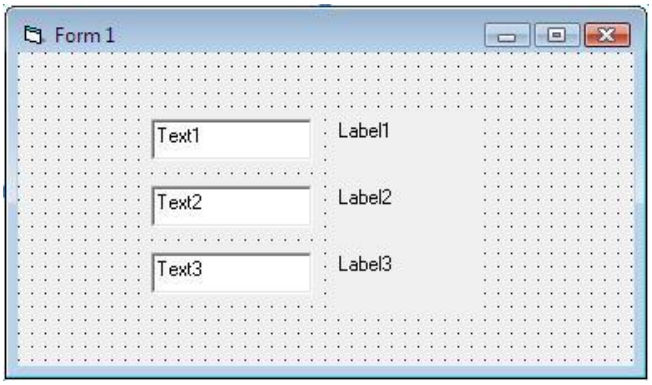
الحل:

المعطيات: عائد مشروع اقتصادي (A).

المطلوب: حساب نصيب الأول(x1) والثاني (x2) بحيث:

$$x1 = 1/4 * A$$
$$x2 = 3/4 * A$$

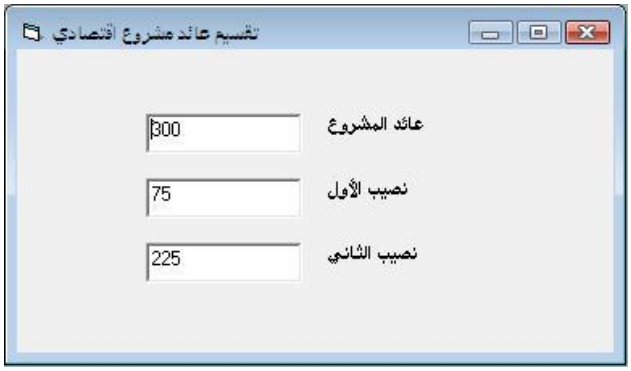
أولا : تصميم نافذة البرنامج:



ثانيا : البرنامج المطلوب:

```
Private Sub Form_Load()
Rem قام بكتابة هذا البرنامج المهدي جبريل
A! = InputBox("أدخل عائد المشروع الاقتصادي")
x1! = 1/4 * A!
x2! = 3/4 * A!
Form1.caption = "تقسيم عائد مشروع اقتصادي"
Label1.caption = "عائد المشروع"
Label2.caption = " نصيب الأول"
Label3.caption = "نصيب الثاني"
Text1.text = A!
Text2.text = x1!
Text3.text = x2!
End Sub
```

نلاحظ هنا أن التعليق في أداة الوصف تم كتابته من خلال البرنامج وكذلك عنوان نافذة البرنامج، والنافذة التالية تبين ناتج تنفيذ البرنامج:



الباب الخامس: جمل التحكم

تهدف جمل التحكم بشكل عام إلى إنجاز العبارات المنطقية، ويوجد شكلين أساسين لهذه الجمل في الفيجوال بيسك، هما:

- جملة (إذا كان – وإلا) (If – then - else).
- هيكل الاختيار المتعدد (Select case).

أولا: جملة (إذا كان – وإلا) (If – then - else):

تأخذ هذه الجملة الشكل العام التالي:

```
If (شرط منطقي) then
        جملة أو أكثر
Else
        جملة أو أكثر
End if
```

في حالة تحقق الشرط المنطقي (True) سيتم تنفيذ مجموعة الجمل التي تأتي بعد الكلمة (Then)، وفي حالة عدم تحققه (False) سيتم تنفيذ مجموعة الجمل التي تأتي بعد الكلمة (else).

ملاحظة هامة:

ليس من الضروري أن تحتوي جملة (If) على (else)، وبالتالي ستكون جملة (If) بالشكل العام التالي:

```
If (شرط منطقي) then
        جملة أو أكثر
End if
```

مثال①: أكتب برنامج يقوم بقراءة قيمتين ويطبع أكبرهما؟

الحل:

المعطيات: قيمتين (A) ، (B).

المطلوب: إيجاد أكبر قيمة من بينهما (Max)، بحيث:

إذا كانت (A>B) إذاً

Max = A

وإلا

Max = B

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لإيجاد العدد الأكبر من بين عددين
A! = InputBox("أدخل العدد الأول")
B! = InputBox("أدخل العدد الثاني")
If (A! > B!) then
        Max! = A!
Else
        Max! = B!
End if
Print Max!, "العدد الأكبر"
End Sub
```

وعند اختبار البرنامج بالقيمتين (10، 3) نتجت النافذة التالية:

رموز العمليات المنطقية:

يمكن حصر العمليات المنطقية بالجدول التالي:

الرمز	المعنى
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوي
<=	أصغر من أو يساوي
=	يساوي
<>	لا يساوي

ملاحظة: نلاحظ أن معنى العمليات المنطقية أخذ بأن هذه العمليات مكتوبة باللغة الإنجليزية وهي اللغة التي تكتب بها البرامج في بيئة البرمجة فيجوال بيسك.

مثال ② : أكتب برنامج لحساب قيمة الضريبة علما بأن: الضريبة = 0 إذا كان الدخل السنوي أقل من 6000 دينار، والضريبة = 5٪ من قيمة الدخل السنوي إذا كان هذا الدخل أكبر من أو يساوي 6000 دينار؟ استخدم مربع النص وأداة الوصف؟

الحل:

المعطيات: دخل سنوي (income).

المطلوب: حساب قيمة الضريبة (tax)، بحيث:

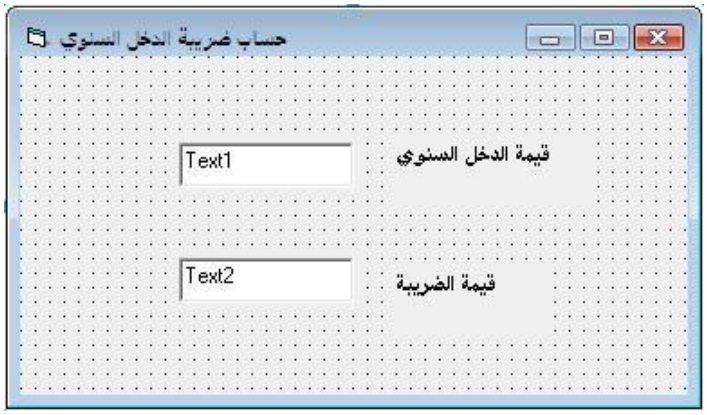
إذا كانت (income >= 6000) إذاً

tax = income * 5 / 100

وإلا

tax = 0

أولا: تصميم نافذة البرنامج:



ثانيا: البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لحساب ضريبة الدخل السنوي
income! = InputBox("أدخل الدخل السنوي")
If (income! >= 6000) then
        tax! = income! * 5 /100
Else
        tax! = 0
End if
Text1.text = income!
Text2.text = tax!
End Sub
```

وعند تنفيذ البرنامج وافتراض الدخل السنوي بقيمة (6000 دينار) نتجت النافذة التالية:

مثال ③ : أكتب برنامج لحساب تكاليف فرش أرضية مبنى بحيث يتم إدخال مساحة الفرش بالمتر المربع مع العلم أن تكلفة المتر المربع الواحد 10 دينار، وهناك خصم (12٪) إذا كانت المساحة أكثر من 150 متر مربع؟

الحل:

المعطيات: مساحة فرش أرضية (A).

المطلوب: حساب تكلفة الفرش (T)، بحيث:

إذا كانت (A>150) إذاً

$$T = (A * 10)\ (1 - 0.12)$$

وإلا

$$T = (A * 10)$$

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem  هذا البرنامج لحساب تكاليف فرش أرضية
A! = InputBox("أدخل مساحة الفرش")
If (A! >150) then
        T! = (A! * 10) * (1 – 0.12)
Else
        T! = A! * 10
End if
Label1.caption = "مساحة الفرش"
Label2.caption = "تكلفة الفرش"
Text1.text = A!
Text2.text = T!
End Sub
```

الروابط المنطقية:

تهدف الروابط المنطقية لربط شرطين أو أكثر لغرض تقليص عدد المقارنات في البرنامج، ويتواجد منها نوعين رئيسين في لغة فيجوال بيسك هما كالتالي:

1. الرابط المنطقي (AND) (بمعنى "و"):-

يربط بين عبارتين منطقتين (E1) و (E2) لتنتج عبارة منطقية جديدة (E3) كالتالي:

E3 = E1 AND E2

إذا كانت كلا العبارتين (E1، E2) صحيحتين (True) فإن ناتج العبارة (E3) سيكون صحيحا ، وإلا فإن ناتج العبارة سيكون غير صحيح (False).

جدول عبارة (AND):

E1	E2	E3 = E1 AND E2
T	T	T
T	F	F
F	T	F
F	F	F

مثل:

1. (4 > 3) AND (3 > 2) ☞ T

2. (6 < 5) AND (10 > 9) ☞ F

2. الرابط المنطقي (OR) (بمعنى "أو"):-

يربط بين عبارتين منطقتين (E1) و (E2) لتنتج عبارة منطقية جديدة (E3) كالتالي:

E3 = E1 OR E2

إذا كانت كلا العبارتين (E1، E2) غير صحيحتين (False) فإن ناتج العبارة (E3) سيكون غير صحيح، وإلا فإن ناتج العبارة سيكون صحيح (True) دائما.

جدول عبارة (OR):

E1	E2	E3 = E1 OR E2
T	T	T
T	F	T
F	T	T
F	F	F

مثل:

1. (-5 > 0) OR (3 > 8) ☞ F

2. (6 < 5) OR (10 > 9) ☞ T

مثال ④ : أكتب برنامج لطباعة أكبر قيمة من بين ثلاثة قيم؟ استخدم مربع النص وأداة الوصف في نافذة البرنامج؟

الحل:

المعطيات: ثلاثة قيمة (A, B, C).

المطلوب: إيجاد أكبر قيمة من بين الثلاثة، بحيث:

إذا كانت (A>B) و (A>C) إذا

Max = A

وإلا إذا كانت (B>A) و (B>C) إذا

Max = B

وإلا

Max = C

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لإيجاد أكبر قيمة من بين ثلاثة
A! = InputBox("أدخل العدد الأول")
B! = InputBox("أدخل العدد الثاني")
C! = InputBox("أدخل العدد الثالث")
If (A! > B!) AND (A! > C!) then
        Max! = A!
Else
If (B! > A!) AND (B! > C!) then
        Max! = B!
Else
        Max! = C!
End if
End if
Label1.caption = "العدد الأكبر"
Text1.text = Max!
End Sub
```

ثانيا: جملة الاختيار المتعددة (هيكل الاختيار) (Select case):

تختلف هذه الجملة الشرطية عن جملة (If) من حيث عدد التفرعات، فجملة (If) الشرطية تتكون من تفرعين فقط، في حين الهيكل الاختيار يحتوي عدة تفرعات.

ملاحظة:

يمكن إضافة تفرع باسم (Case else) في جملة الاختيار المتعدد بحيث تكون آخر تفرع، فإذا لم تتوافق جميع مقارنات الفروع السابقة يتم اللجوء لهذا التفرع، والذي عادة ما يحتوي على تنبيه لخطأ في الإدخال.

مثال ⑤ : أكتب برنامج يقرأ حرف ويطبع التالي:

- كلمة (Red) إذا كان الحرف (R).
- كلمة (Blue) إذا كان الحرف (B).
- كلمة (Green) إذا كان الحرف (G).

استخدم جملة الاختيار مرة وجملة (If) مرة أخرى؟.

الحل:

أولا: باستخدام جملة (If) الشرطية:

```
Private Sub Form_Load()
Rem هذا البرنامج باستخدام جملة إذا كان – وإلا
C = InputBox("أدخل الحرف R أو B أو G")
If (C = "R") OR (C = "r") then
        Print "Red"
else
If (C = "B") OR (C = "b") then
        Print "Blue"
else
If (C = "G") OR (C = "g") then
        Print "Green"
else
        Print "اختيار خاطئ"
End if
End if
End if
End Sub
```

ثانيا: باستخدام جملة الاختيار المتعدد:

```
Private Sub Form_Load()
Rem  هذا البرنامج باستخدام جملة الاختيار المتعدد
C = InputBox("أدخل الحرف R أو B أو G")
Select case C
Case "R", "r"
            Print "Red"
Case "B", "b"
            Print "Blue"
Case "G", "g"
            Print "Green"
Case else
            Print "اختيار خاطئ"
End select
End Sub
```

يعتبر هذا البرنامج مختصر وأيسر في التتبع مقارنة بالبرنامج المكتوب بجملة (If) الشرطية.

مثال ⑥ : أكتب برنامج لطباعة التقدير بناءا على المعدل الذي تحصل عليه الطالب بعد أن تتم قراءته كالتالي:

- ضعيف: المعدّل أكبر من صفر وأصغر من 50.
- مقبول: المعدّل أكبر من أو يساوي 50 وأصغر من 65.
- جيد: المعدّل أكبر من أو يساوي 65 وأصغر من 75.
- جيد جدا: المعدّل أكبر من أو يساوي 75 وأصغر من 85.
- ممتاز: المعدّل أكبر من أو يساوي 85 وأصغر أو يساوي 100.

باستخدام جملة الاختيار المتعدد؟

الحل:

المعطيات: معدل طالب (G).

المطلوب: طباعة التقدير بناءا على المبين في المثال.

لأن المعدل هو قيمة عددية كسرية سنستخدم أداة (Is) لتشمل جميع الأجزاء الكسرية للقيم.

البرنامج المطلوب باستخدام جملة الاختيار المتعدد:

```
Private Sub Form_Load()
Rem هذا البرنامج لطباعة تقدير الطالب بناءا على المعدل
G! = InputBox("أدخل معدل الطالب")
Select case G!
Case Is < 50
        Print "ضعيف"
Case Is < 65
        Print "مقبول"
Case Is < 75
        Print "جيد"
Case Is < 85
        Print "جيد جدا"
Case Is < 100
        Print "ممتاز"
Case else
        Print "معدل خاطئ"
End select
End Sub
```

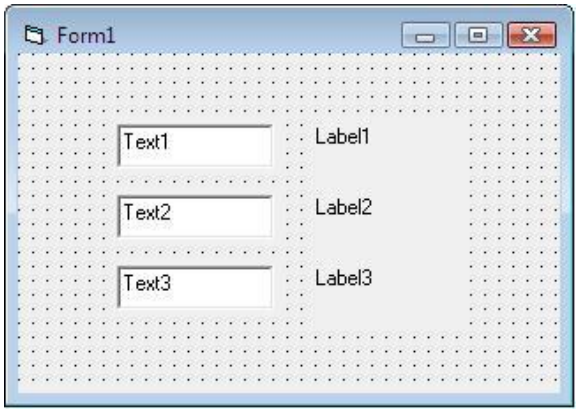
مثال ⑦ : أكتب برنامج يقرأ قيمتين، ويجري إحدى العمليات الحسابية عليها بناءا على اختيار المستخدم، استخدم مربع النص وأداة الوصف في تصميم نافذة البرنامج؟

الحل:

المعطيات: قيمتين (A,B).

المطلوب: إجراء إحدى العمليات الحسابية الأربعة على القيمتين بناءا على طلب المستخدم.

أولا: تصميم نافذة البرنامج:



ثانيا : البرنامج المطلوب:

```
Private Sub Form_Load()
Rem  2008 هذا البرنامج كتب بتاريخ
A! = InputBox("أدخل القيمة الأولى")
B! = InputBox("أدخل القيمة الثانية")
C = InputBox("/  *  -  +  :إختر العملية المراد عملها")
Select case C
Case "+"
            T! = A! + B!
            M$ = "ناتج الجمع"
Case "-"
            T! = A! - B!
            M$ = "ناتج الطرح"
Case "*"
            T! = A! * B!
            M$ = "ناتج الضرب"
Case "/"
            T! = A! / B!
            M$ = "ناتج القسمة"
Case else
            T! = 0
            M$ = "اخترت عملية خاطئة"
End select
Label1.caption = "القيمة الأولى"
Text1.text = A!
Label2.caption = "القيمة الثانية"
Text2.text = B!
Label3.caption = M$
Text3.text = T!
End Sub
```

الباب السادس: الحلقات

تهدف الحلقات بشكل عام إلى تكرار عمل معين عدد من المرات أو وفق شرط معين يتحقق، هذا التكرار يتم بسرعة عالية جدا.

ومن أهم أنواع هذه الحلقات:

1. حلقة (من –إلى) (For - Next).
2. حلقة (نفذ – طالما) (Do - While).
3. حلقة (نفذ – حتى) (Do - Until).

أولا: حلقة (من – إلى) (For - Next):

تمتلك هذه الحلقة الشكل العام التالي:

```
For i = n to m step x
عمل معين يراد تكراره (جملة أو أكثر)
Next i
```

حيث:

i: عداد الحلقة (دليل الدورة).

n: قيمة بداية العدّ.

m: قيمة نهاية العدّ أو التوقف.

x: قيمة خطوة العدّاد.

مثال① : أكتب برنامج لطباعة الأعداد الزوجية المحصورة بين (0 -50)؟

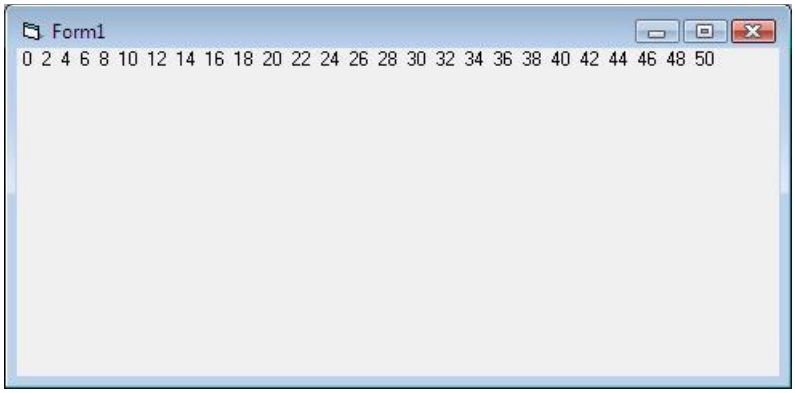
الحل:

نحتاج هنا إلى عداد يبدأ من الصفر ويزداد بخطوة قدرها (2) ويتوقف عند القيمة (50).

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج يطبع الأعداد الزوجية من 0 إلى 50
For k = 0 to 50 step 2
    Print k;
Next k
End sub
```

ناتج تنفيذ البرنامج:



ملاحظة:

نلاحظ هنا أن الطباعة متلاصقة وفي سطر واحد وهذا يرجع للفاصلة المنقوطة الموضوعة في نهاية أمر الطباعة بداخل الحلقة.

مثال② : أكتب برنامج لحساب مبيعات محل لمدة أسبوع واحد من مبيعات يومية؟

الحل:

المعطيات: مبيعات يومية لمدة أسبوع واحد (d).

المطلوب: حساب إجمالي مبيعات الأسبوع (Sum).

نحتاج هنا إلى عداد يبدأ من واحد ويزداد بخطوة قدرها (1) ويتوقف عند القيمة (7).

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لحساب مبيعات محل لمدة أسبوع واحد
Sum! = 0
For i = 1 to 7
    d! = InputBox("أدخل مبيعات اليوم")
    Sum! = Sum! + d!
Next i
print sum!, "إجمالي مبيعات الأسبوع"
End sub
```

ملاحظة:

عندما يكون قيمة خطوة العد بالقيمة (1) فلا داعي لكتابة هذه القيمة في الحلقة، وبالتالي فإن ذلك يعني أنه إذا لم يتم تحديد قيمة الخطوة في الحلقة (For) سيعتبرها فيجوال بيسك بقيمة (1).

مثال③ : أكتب برنامج يقرأ درجات (n) من الطلبة ويطبع أكبر درجة؟

الحل:

المعطيات: درجات (n) من الطلبة.

المطلوب: إيجاد أكبر درجة من بين تلك الدرجات (Max).

نلاحظ هنا أننا نحتاج إلى قيمة (n)، لذلك سنستخدم دالة الإدخال لقراءتها.

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لإيجاد أكبر درجة طالب
Max! = 0
n! = InputBox("أدخل عدد الطلبة")
For i = 1 to n
    d! = InputBox("أدخل درجة الطالب")
    if (d! > Max!) then
            Max! = d!
    End if
Next i
Label1.caption = "أكبر درجة"
Text1.text = Max!
End sub
```

مثال④ : أكتب برنامج يقرأ درجات وأسماء 10 طلاب ويطبع فقط درجات وأسماء ذوي التقدير الممتاز؟

الحل:

المعطيات: أسماء ودرجات 10 طلا ب:

المطلوب: طباعة أسماء ودرجات ذوي التقدير الممتاز.

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لإيجاد أكبر درجة طالب
For i = 1 to 10
    Name$ = InputBox("أدخل اسم الطالب")
    d! = InputBox("أدخل درجة الطالب")
    if (d! >= 85) then
            Print Name$, d!
    End if
Next i
End sub
```

مثال⑤ : أكتب برنامج يطبع الشكل التالي:

```
1  2  3  4  5  6
1  2  3  4  5
1  2  3  4
1  2  3
1  2
1
```

الحل:

نحتاج في مثل هذه الحالات إلى حلقتين متداخلتين، الخارجية لعد السطور والداخلية لعد الأعمدة.

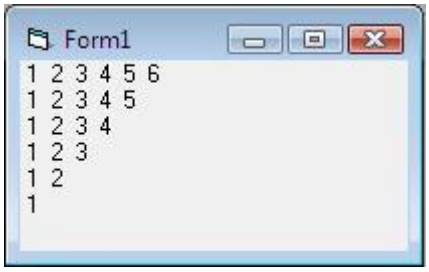
عدد الأسطر (6) لذلك فإن حلقة السطور ستبدأ بالقيمة (1) وتنتهي بالقيمة (6).

عدد الأعمدة أو الرموز في كل سطر يتناقص في كل سطر برمز واحد، لذلك سيبدأ بالقيمة (6) وهي عدد الرموز في السطر الأول، وينتهي بالقيمة (1) وهي عدد رموز آخر سطر.

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem مثال على تداخل الحلقات
n = 6
For i = 1 to 6
    For j = 1 to n
            Print j;
    Next j
    n = n – 1
    Print
Next i
End sub
```

الشكل الناتج بعد تنفيذ البرنامج:



إذا كان الشكل المطلوب طباعته كما يلي:

```
1
1  2
1  2  3
1  2  3  4
1  2  3  4  5
1  2  3  4  5  6
```

هنا عدد الأعمدة أو الرموز في كل سطر يزداد في كل سطر برمز واحد، لذلك سيبدأ بالقيمة (1) وهي عدد الرموز في السطر الأول، وينتهي بالقيمة (6) وهي عدد رموز آخر سطر.

وفي حالة كان الشكل المطلوب طباعته كالتالي:

```
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

فإنه فقط يتم تغيير أمر الطباعة (Print j;) إلى الأمر (Print "*";).

مثال⑥: أكتب برنامج لطباعة الشكل التالي:

```
A   B   B
B   A   B
B   B   A
```

الحل:

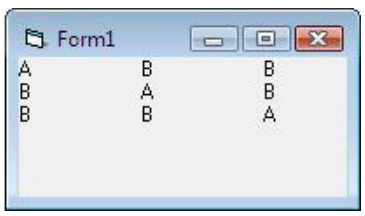
هنا أيضا نحتاج إلى حلقتين حلقة للسطور وأخرى للأعمدة، ولكن يجب البحث عن علاقة منطقية في الشكل بالإجابة على السؤال: متى يطبع (A) ومتى يطبع (B)؟

في هذا الشكل نلاحظ أنه كلما تساوى رقم السطر ورقم العمود طبع الرمز (A)، وإلا فإنه يطبع الرمز (B).

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem مثال على تداخل الحلقات
For i = 1 to 3
    For j = 1 to 3
        if (i = j) then
            print "A",
        else
            print "B",
        end if
    Next j
    Print
Next i
End sub
```

ناتج تنفيذ البرنامج:



ثانيا: حلقة (نفذ – طالما) (Do - While):

تعمل هذه الحلقة على تكرار تنفيذ عمل معين طالما يتحقق شرط معين، وتأخذ الشكل العام التالي:

```
Do  While (شرط منطقي)
عمل معين يراد تكراره (جملة أو أكثر)
loop
```

حيث:

(شرط منطقي): هو شرط استمرار الحلقة.

loop: نهاية الحلقة.

ملاحظة:

يمكن تحوير هذه الحلقة لتعمل عمل حلقة (For)، حيث يتم تعيين متغير بقيمة مبدئية للعد وشرط استمرار وقيمة زيادة للعداد.

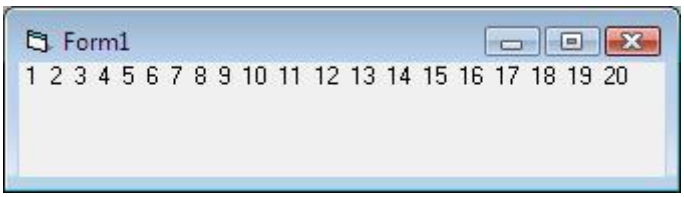
مثال⑦: أكتب برنامج لطباعة الأعداد من (1 -20) باستخدام حلقة (نفذ – طالما)؟

الحل:

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem مثال على حلق نفذ – حتى
C = 1
Do While (C<=20)
    Print C;
    C = C + 1
Loop
End sub
```

وعند تنفيذ البرنامج تظهر النافذة التالية:



مثال⑧ : أكتب برنامج لحساب متوسط درجات مجموعة طلاب في مادة الحاسوب، علما بأن عدد الطلبة غير معلوم، ويتوقف البرنامج عند إدخال درجة سالبة؟

الحل:

المعطيات: درجات مجموعة من الطلاب غير معلومي العدد.

المطلوب: المتوسط الحسابي لدرجاتهم (ave).

نلاحظ هنا أن عدد الطلبة غير معلوم لذلك لا يمكن استخدام دالة القراءة، وحتى يتم إيجاد المتوسط الحسابي لدرجات الطلاب يجب أولا إيجاد حاصل جمع درجاتهم ثم عدد الطلاب الذين تم جمع درجاتهم، لذلك نحتاج إلى متغيرين هما:

(n) عداد لعد المواد التي يتم جمعها.

(sum) متغير لجمع قيمة جميع الدرجات المدخلة.

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج يتوقف عند إدخال درجة سالبة
n% = 0
Sum! = 0
g! = InputBox("أدخل درجة الطالب")
Do While (g!>=0)
    Sum! = Sum! + g!
    n% = n% + 1
    g! = InputBox("أدخل درجة سالبة للتوقف")
Loop
Ave! = Sum! / n%
Print Ave!,"المتوسط الحسابي لدرجات الطلاب"
End sub
```

ملاحظة هامة:

- علامة القسمة (/) تعيد ناتج كسري، مثل:

  3 / 2 = 1.5

- علامة القسمة (\) تعيد الجزء الصحيح لناتج القسمة، مثل:

  3 \ 2 = 1

ثالثا: حلقة (نفذ – حتى) (Do - Until):

تعمل هذه الحلقة على تكرار تنفيذ عمل معين إلى أن يتحقق شرط معين، لذلك هذا الشرط يسمى شرط توقف، وتأخذ هذه الحلقة الشكل العام التالي:

```
Do  Until (شرط منطقي)
عمل معين يراد تكراره (جملة أو أكثر)
loop
```

حيث:

(شرط منطقي): هو شرط توقف الحلقة.

loop: نهاية الحلقة.

مثال⑨ : أكتب البرنامج يقوم بجمع أعداد موجبة ويتوقف عند إدخال قيمة سالبة؟

الحل:

المعطيات: مجموعة أعداد موجبة يتوقف إدخالها عند إدخال قيمة سالبة.

المطلوب: إيجاد حاصل جمع الأعداد الموجبة (sum).

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem مثال على حلق نفذ – طالما
Sum! = 0
y! = 0
Do Until (y! < 0)
    Sum! = Sum! + y!
    y! = InputBox("أدخل قيمة سالبة للتوقف")
Loop
Label1.caption = "مجموع القيمة الموجبة"
Text1.text = Sum!
End sub
```

ملاحظة حول الحلقات بشكل عام:

1. الشرط في حلقة (نفذ – طالما) شرط استمرار، بينما في حلقة (نفذ – حتى) شرط توقف.
2. يمكن إعادة كتابة البرنامج المكتوب بحلقة (نفذ -طالما) باستخدام حلقة (نفذ -حتى)، ويتطلب ذلك كتابة اسم الحلقة (Until - While) وتغيير الشرط من شرط توقف إلى شرط استمرار أو العكس.
3. يمكن إعادة كتابة البرنامج من حلقة (for) إلى أي حلقة شرطية، والعكس غير صحيح، حيث يمكن استخدام حلقة (for) فقط في حالة علمنا عدد مرات التكرار وبداية ونهاية العد وقيمة الزيادة.

الباب السابع: المصفوفات

ما هي المصفوفة؟

هي مجموعة من البيانات أو العناصر ذات نوع واحد وتحت اسم واحد.

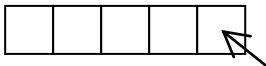
كل عنصر من عناصر المصفوفة يتطلب موقع (خلية) في ذاكرة الجهاز.

مثل: مصفوفة أيام الأسبوع – مصفوفة الجدول الدراسي ... الخ جميعها مصفوفات يمكن تخزينها في ذاكرة الجهاز تخزينا مؤقتا يتلاشى بعد إيقاف تنفيذ البرنامج.

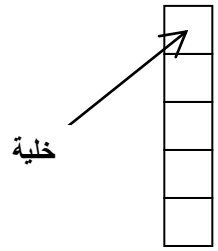
وفي هذا الباب سنتعرض لنوعين هما: مصفوفة ذات البعد الواحد، ومصفوفة ذات البعدين.

أولا: مصفوفة البعد الواحد (One dimensional):

هي مصفوفة عناصرها إما في صف واحد أو في عمود واحد.



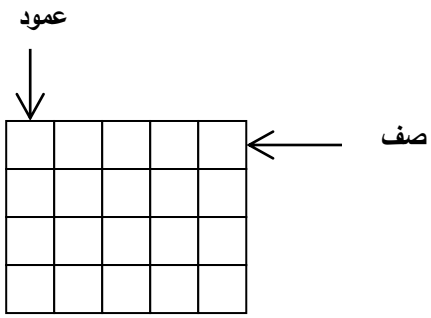
مصفوفة ذات بعد واحد في صف واحد وعناصرها (5)



مصفوفة ذات بعد واحد في عمود واحد وعناصرها (5)

ثانيا: مصفوفة البعدين (Tow dimensional):

هي مصفوفة مرتبة في أكثر من عمود وأكثر من صف.



مصفوفة ذات بعدين بحجم (4*5) – أي 4 صفوف و 5 أعمدة – تحتوي على 20 خلية أو عنصر

حجز مواقع المصفوفة في الذاكرة:

وهي أو خطوة يجب القيام بها عند استعمال المصفوفة في البرنامج، ويتم ذلك كما يلي:

أ. حجز مواقع المصفوفة الأحادية البعد:

```
Dim a (n) as Type
```

حيث:

Dim: كلمة محجوزة في فيجوال بيسك اختصار لكلمة (Dimension).

a : اسم المصفوفة.

n : عدد عناصر المصفوفة.

Type : نوع عناصر المصفوفة.

ب. حجز مواقع المصفوفة ثنائية البعد:

```
Dim a (n, m) as Type
```

حيث:

Dim: كلمة محجوزة في فيجوال بيسك اختصار لكلمة (Dimension).

a : اسم المصفوفة.

n : عدد الصفوف في المصفوفة.

m : عدد الأعمدة في المصفوفة.

Type : نوع عناصر المصفوفة.

أمثلة:

1. Dim number(10) as integer
   - تحجز (10) مواقع للمصفوفة (number) من النوع الصحيح (integer).

2. Dim cost(25) as single
   - تحجز (25) موقع للمصفوفة (cost) من النوع الكسري (single).

3. Dim table(5,7) as string
   - تحجز (35) موقع للمصفوفة (table) من النوع الحرفي (string).

4. Dim x(50)
   - تحجز (50) موقع للمصفوفة (x) من النوع المرن (variant).

5. Dim V as single
   - الإعلان عن متغير باسم (V) من النوع الكسري (single).

قراءة وطباعة المصفوفة:

مثال① : أكتب برنامج يقوم بقراءة (5) قيم ويحسب متوسطها الحسابي ويطبع الفرق بين كل قيمة والمتوسط باستخدام المصفوفات؟

الحل:

نحتاج في هذا المثال إلى:

- مصفوفة (V) عدد عناصرها (5).
- متغير (sum) حاصل جمع القيم.
- متغير (ave) المتوسط الحسابي للقيم.

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem  مثال على قراءة مصفوفة
Dim V(5) As Single
Sum! = 0
For i = 1 to 5
    V(i) = InputBox("أدخل القيمة")
    sum! =  sum! + V(i)
Next i
ave! = sum! / 5
for i = 1 to 5
    d = V(i) – ave
    print V(i),d
Next i
End sub
```

مثال② : أكتب برنامج يقوم بقراءة أيام الأسبوع وطباعتها بترتيب معكوس؟

الحل:

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem طباعة أيام الأسبوع بترتيب معكوس
Dim days(7) As Single
days(1) = "السبت"
days(2) = "الأحد"
days(3) = "الاثنين"
days(4) = "الثلاثاء"
days(5) = "الأربعاء"
days(6) = "الخميس"
```

```
days(7) = "الجمعة"
for i = 7 to 1 step -1
    print days(i)
Next i
End sub
```

مثال③ : أكتب برنامج لقراءة أسماء وأرقام هواتف (10) أشخاص وطباعتها بترتيب معكوس؟

الحل:

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem  مثال على قراءة وطباعة مصفوفة
Dim n(5), p(5) As String
for i = 1 to 5
    n(i) = InputBox("أدخل الاسم")
    p(i) = InputBox("أدخل الرقم")
Next i
for i = 1 to 5
    print n(i), p(i)
Next i
End sub
```

مثال④ : أكتب برنامج يقوم بقراءة عدد الطلبة بمعهد دراسي حسب السنة الدراسية والتخصص، ثم يطبع هذه البيانات في جدول مناسب ؟ ويحسب عدد الطلبة في كل سنة دراسية ؟ والعدد الإجمالي للطلبة بالمعهد ، علما بأن نظام المعهد 3 سنوات دراسية و4 تخصصات؟

الحل:

بداية لنا أن نتصور شكل الجدول المراد طباعته كالتالي:

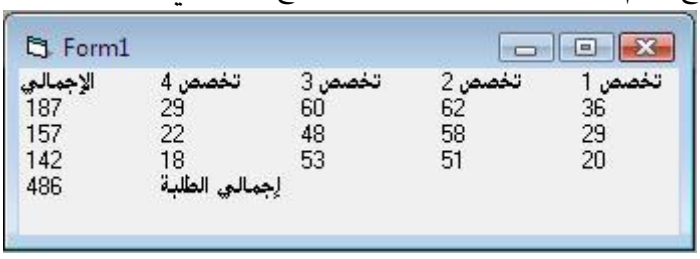
السنة الدراسية	تخصص 1	تخصص 2	تخصص 3	تخصص 4	الإجمالي
سنة أولى	St(1,4)	St(1,3)	St(1,2)	St(1,1)	Sum (1)
سنة ثانية	St(2,4)	St(2,3)	St(2,2)	St(2,1)	Sum (2)
سنة ثالثة	St(3,4)	St(3,3)	St(3,2)	St(3,1)	Sum (3)
إجمالي عدد الطلبة					Total

- نحتاج إلى مصفوفة رئيسية (st) لتخزين عدد الطلبة حسب السنة الدراسية والتخصص.
- نحتاج إلى مصفوفة (sum) لتخزين عدد الطلاب حسب السنة الدراسية.
- نحتاج إلى متغير (Total) لحساب إجمالي عدد الطلبة.

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem  تمت كتابة هذه البرنامج في العام 2008
Dim st(3,4), sum(3), Total As Integer
Rem في هذه الحلقة سيتم قراءة مصفوفة الطلبة وحساب إجمالي الطلبة في كل سنة
for i = 1 to 3
    sum(i) = 0
    for j = 1 to 4
                st(i,j) = InputBox("أدخل الطلاب حسب السنة والتخصص")
                sum(i) = sum(i) + st(i,j)
    Next j
Next i
Rem في هذا الجزء من البرنامج سيتم حساب إجمالي الطلبة بالمعهد
Total = 0
for i = 1 to 3
        Total = Total + sum(i)
Next i
Rem في هذا الجزء من البرنامج سيتم طباعة البيانات في شكل جدول
Print "تخصص 1","تخصص 2","تخصص 3","تخصص 4","الإجمالي"
for i = 1 to 3
    Print sum(i),
    for j = 1 to 4
                Print st(i,j),
    Next j
     Print
Next i
     Print Total, "إجمالي الطلبة"
End sub
```

عند اختبار البرنامج بقيم افتراضية كانت نافذة النواتج كالتالي:



مثال⑤ : أكتب برنامج لتخزين جدول الامتحانات وطباعته لأسبوع واحد على شكل مصفوفة بحيث يتم إدخال البيانات كالتالي:

اليوم	الامتحان الأول	الامتحان الثاني

ثم طباعة هذا الجدول بشكل مناسب؟

الحل:

يمكن تمثيل جدول الامتحانات وتوزيع رقم العمود والصف عليه كالتالي:

اليوم	الامتحان الأول	الامتحان الثاني
(1,3)	(1,2)	(1,1)
(2,3)	(2,2)	(2,1)
(3,3)	(3,2)	(3,1)
(4,3)	(4,2)	(4,1)
(5,3)	(5,2)	(5,1)
(6,3)	(6,2)	(6,1)

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج لتخزين جدول الامتحانات
Dim Table(6,3) As String
Rem في هذه الحلقة سيتم قراءة جدول الامتحانات
for i = 1 to 6
        Table(i,3) = InputBox("أدخل يوم الامتحان")
        Table(i,2) = InputBox("أدخل الامتحان الأول")
        Table(i,1) = InputBox("أدخل الامتحان الثاني")
Next i
Rem في هذه الحلقة سيتم طباعة جدول الامتحانات
Print "جدول الامتحانات"
Print "اليوم","الأول","الثاني"
For i = 1 to 6
        For j = 1 to 3
                Print Table(i,j),
        Next j
        Print
Next i
End sub
```

عند تنفيذ البرنامج واختباره ببعض القيم تظهر النافذة التالية:

مثال⑥ : أكتب برنامج لقراءة أسماء ودرجات عشرة طلاب في مادة دراسية واحد ، ويطبع اسم ودرجة الطالب المتحصل على أعلى درجة باستخدام المصفوفات؟

الحل:

البرنامج المطلوب:

```
Private Sub Form_Load()
Dim St(10) As String
Dim G(10) As Single
Rem هذا الجزء من البرنامج لقراءة أسماء ودرجات الطلاب
for i = 1 to 10
        St(i) = InputBox("أدخل اسم الطالب")
        G(i) = InputBox("أدخل الدرجة")
Next i
Rem هذا الجزء من البرنامج لفحص أعلى درجة طالب
Max! = 0
For i = 1 to 10
        If (G(i) > Max!) then
                Max! = G(i)
                N$ = St(i)
        End if
Next i
Rem  هذا الجزء من البرنامج لطباعة أسماء الطلاب والدرجات وصاحب أعلى درجة
Print "الاسم","الدرجة"
For i = 1 to 10
        Print G(i), St(i)
Next i
Print
Print Max, N$, "الطالب الأعلى درجة"
End sub
```

عند تنفيذ البرنامج واختباره ببعض القيم تظهر النافذة التالية:

مثال⑦: إذا كانت المصفوفة Q(k) تمثل عدد القطع من الصنف رقم k، والمصفوفة P(k) تبين سعر القطعة الواحدة من الصنف k، أكتب برنامج يقرأ عدد القطع التي تم بيعها من كل صنف وسعر القطعة، ثم يطبع البيانات المدخلة ومجموع المبيعات بافتراض أن عدد الأصناف التي تم بيعها هو (12)؟

الحل:

هذا البرنامج جزء من فاتورة تجارية كالتالي:

رقم متسلسل (k)	عدد القطع (الكمية) Q(k)	سعر القطعة P(k)	سعر الكمية PQ(k)
:	:	:	:
		الإجمالي	Total

سعر الكمية = عدد القطع × سعر القطعة

البرنامج المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج يمثل فاتورة تجارية
Dim P(12), Q(12), PQ(12), Total As Single
for k = 1 to 12
        Q(k) = InputBox("أدخل كمية الصنف")
        P(k) = InputBox("أدخل سعر القطعة")
        PQ(k) = Q(k) * P(k)
Next k
Rem هذا الجزء من البرنامج لحساب إجمالي الفاتورة
Total = 0
For k = 1 to 12
        Total = Total + PQ(k)
Next k
Print "سعر الكمية", "الكمية", "سعر القطعة"
For k = 1 to 12
        Print PQ(k), Q(k), P(k)
Next k
Print
Print  Total, "إجمالي الفاتورة"
End sub
```

الباب الثامن: البرامج الفرعية

تهدف البرامج الفرعية إلى تقسيم البرنامج الرئيسي إلى أجزاء عدة، هذا التقسيم لعدة أسباب أهمها تبسيط العمل من حيث سهولة البرمجة والاختبار والتعديل وإمكانية الاستفادة من البرنامج الفرعي في أكثر من مشروع مستقبلا.

يمكن تقسيم البرامج الفرعية في لغة فيجوال بيسك إلى ثلاثة أنواع:

1. الإجراء الفرعي (Sub procedure).
2. إجراء الدالة (Function procedure).
3. إجراء الخاصية (Property procedure).

أولا: الأجراء الفرعي (*Sub procedure*):-

وهو أحد أهم أنواع البرامج الفرعية، ويمكنه أن يستقبل قيمة أو أكثر من برنامج الاستدعاء وإعادة قيمة أو أكثر إلى برنامج الاستدعاء، ويمتلك الشكل العام التالي:

```
Sub name (V1,V2,…,Vn)
جملة أو أكثر
End sub
```

حيث:

Sub : كلمة محجوزة تشير إلى بداية البرنامج الفرعي.

name : اسم الإجراء الفرعي.

(V1,V2,…,Vn) : متغيرات الإدخال والإخراج من وإلى البرنامج الفرعي، وتسمى أيضا بالبارامترات (*Parameters*).

مثال①: أكتب برنامج فرعي لحساب مساحة مستطيل؟ واكتب برنامج الاستدعاء (البرنامج الرئيسي)؟

الحل:

مدخلات هذا البرنامج الفرعي هي الطول (L) والعرض (W)، ومخرجاته هي المساحة (A).

البرنامج الفرعي المطلوب:

```
Sub Area (L!,W!,A!)
    A! = W! * L!
End sub
```

يمكن تمثيل هذا البرنامج الفرعي كما في الشكل التالي:

برنامج الاستدعاء المطلوب:

```
Private Sub Form_Load()
Rem هذا البرنامج يمثل البرنامج الرئيسي
L! = InputBox("أدخل طول المستطيل")
W! = InputBox("أدخل عرض المستطيل")
Call Area (L!,W!,A!)
Label1.caption = "مساحة المستطيل"
Text1.text = A!
End sub
```

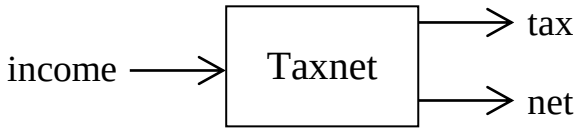
ملاحظات هامة:

- سمي برنامج الاستدعاء بهذا الاسم لأنه يحتوي على العبارة () والتي تعمل على تنفيذ الإجراء الفرعي المبين اسمه أمامها، كما يسمى هذا البرنامج أيضا بالبرنامج الرئيسي.
- يكتب البرنامج الفرعي في نافذة تحرير البرنامج فوق أو تحت البرنامج الرئيسي ولا يؤثر ذلك في تنفيذ الإجراء ولا البرنامج الرئيسي.

مثال②: أكتب إجراء فرعي لحساب الضريبة (tax) وصافي الدخل (net) بافتراض أن الضريبة تعادل (23٪) من إجمالي الدخل (income)؟ واكتب البرنامج الرئيسي؟

الحل:

نفترض أن اسم الإجراء (Taxnet)، ويمكن تمثيل هذا البرنامج الفرعي كما في الشكل التالي:



البرنامج الفرعي المطلوب:

```
Sub Taxnet (income!, tax!, net!)
    tax! = income! * 0.23
    net! = income! – tax!
End sub
```

البرنامج الرئيسي:

```
Private Sub Form_Load()
Rem هذا البرنامج يمثل البرنامج الرئيسي
income! = InputBox("أدخل قيمة الدخل")
Call Taxnet (income!, tax!, net!)
Text1.text = income!
Text2.text = tax!
Text3.text = net!
End sub
```

عند تنفيذ البرنامج تظهر النافذة التالية:

مثال③: أكتب البرنامج الفرعي (count) الذي يستقبل قيمة (k) ثم يطبع الأعداد من (1 إلى k)؟ واكتب برنامج الاستدعاء؟

الحل:

البرنامج الفرعي المطلوب:

```
Sub Count(k%)
    For i = 1 to k%
        Print i;
    Next i
End sub
```

البرنامج الرئيسي:

```
Private Sub Form_Load()
Rem هذا البرنامج يمثل البرنامج الرئيسي
k% = InputBox("أدخل عدد صحيح")
Call Count (k%)
End sub
```

ثانيا: الدالة (*Function*) :-

برنامج الدالة يستلم قيمة أو أكثر من البرنامج الرئيسي، ولكنه يعيد قيمة واحدة فقط في اسمها، وتمتلك الدالة الشكل العام التالي:

```
Function name (V1,V2,…,Vn) as Type
جملة أو أكثر
End Function
```

حيث:

Function : كلمة محجوزة تشير إلى بداية الدالة.

name : اسم الدالة.

(V1,V2,…,Vn) : متغيرات مدخلة للدالة (بارامترات الدالة).

Type: نوع الدالة (Integer, Single, …) ويتحدد بنوع القيمة الوحيدة التي تعيدها الدالة.

ملاحظات:

- في حالة لم يتم تحديد نوع للدالة فإن برنامج فيجوال بيسك سيعتبرها من النوع المرن.
- الإجراء الفرعي ليس له نوع لأنه يعيد أكثر من قيمة أحيانا.

مثال④ : أكتب الدالة التي تقوم بحساب عدد الدقائق في عدد من الأيام؟ مع العلم أن عدد الأيام يدخل في برنامج رئيسي؟

الحل:

نفرض اسم للدالة وليكن (MDay).

برنامج الدالة (MDay):

```
Function MDay(x!) as single
    MDay = x! * 24 * 60
End Function
```

البرنامج الرئيسي:

```
Private Sub Form_Load()
Rem هذا البرنامج يمثل البرنامج الرئيسي
M! = InputBox("أدخل عدد الأيام")
Print MDay (M!), "عدد الدقائق"
End sub
```

ملاحظة:

نلاحظ أنه تم استدعاء الدالة بالمتغير (M)، والذي يقابله المتغير (x) في الدالة، ويعتبر المتغير (x) متغير داخلي معرف في داخل الدالة (MDay) فقط، حيث تنسخ قيمة المتغير (M) إلى المتغير (x) وتنفذ الدالة وتعيد قيمتها في إسمها.

مثال⑤ : أكتب الدالة (Max2) التي توجد أكبر عدد من بين عددين، واستعملها في إيجاد أكبر عدد من بين 5 أعداد؟

الحل:

برنامج الدالة (Max2):

```
Function Max2(x!, y!)
        If (x! > y!) then
                Max2 = x!
        Else
                Max2 = y!
        End if
End Function
```

برنامج الرئيسي:

```
Private Sub Form_Load()
Rem هذا البرنامج يمثل البرنامج الرئيسي
m! = 0
for k = 1 to 5
      a! = InputBox("أدخل قيمة عددية")
      m! = Max2(m!, a!)
Next k
Text1.text = m!
End sub
```

مثال⑥ : هل يمكن كتابة المثال ① باستخدام الدالة؟

الحل:

نعم، لأن الإجراء الفرعي المستخدم يعيد قيمة واحدة فقط، وهذا لا يتعارض مع الدالة التي تعيد قيمة واحدة فقط هي الأخرى.

ويمكن كتابة برنامج الدالة التالي:

```
Function Area(L!, W!)
      Area = L! * W!
End Function
```

مثال ⑦ : هل يمكن كتابة المثال ② باستخدام الدالة ولماذا؟

الحل:

لا ، لأن البرنامج الفرعي المستخدم من نوع إجراء يعيد قيمتين وبالتالي فإنه لا يمكن كتابة دالة تقوم بذلك.

الدوال الجاهزة (الدوال الداخلية):-

تحتوي لغة فيجوال بيسك العديد من الدوال الجاهزة التي يمكن استدعاؤها بدون أن يكتب برنامجها الفرعي، لأن هذا البرنامج جاهز ضمن لغة بيسك، ويمكن تقسيم هذه الدوال إلى:

أولا: الدوال الرياضية:

ومن هذه الدوال:

- دالة الجذر التربيعي (Sqr)، تعيد الجذر التربيعي لعدد، مثل:

Sqr (4) = 2

- دالة اللوغاريتم الطبيعي (Log)، تعيد اللوغاريتم الطبيعي لعدد، مثل:

Log (10) = 2.302585

- دالة القيمة المطلقة (abs)، تحذف الإشارة السالبة من العدد، مثل:

Abs (-6) = 6
Abs (9) = 9

- دالة جيب الزاوية (sin)، تعيد جيب الزاوية، ويجب أن تكون الزاوية المراد حساب جيبها بالتقدير الدائري (*Radians*)، مثل:

Sin (30*3.14159/180) = 0.5

- دالة الإشارة (sgn)، تعيد القيمة (1) إذا كانت القيمة موجبة، والقيمة (-1) إذا كان القيمة سالبة، مثل:

Sgn (-10) = -1
Sgn (3) = 1

- دالة حذف الجزء الكسري من العدد (int) أو (fix)، تعمل هذه الدالة على إلغاء الجزء الكسري بحذفه من القيمة، مثل:

Int (4.87) = 4
Fix (5.98) = 5

ثانيا: دوال السلاسل الحرفية:

ومن هذه الدوال:

- دالة عدد الرموز في سلسلة حرفية (Len)، تعيد طول سلسلة حرفية، مثل:

Len ("محمد") = 4
Len ("Ali Ali") = 7

- دالة التحويل من رقم إلى نضيد (Str) تعمل على تحويل قيمة عددية إلى قيمة حرفية، مثل:

Str (45) = "45"

- دالة التحويل من نضيد إلى رقم (Val)، تعمل على تحويل نضيد إلى قيمة عددية، مثل:

Val ("3.5") = 3.5
Val ("-86") = -86

ثالثا: دوال إحصائية:

من هذه الدوال:

- دالة المتوسط الحسابي (average) تعيد المتوسط الحسابي لمجموعة قيم مثل:

Average (2,4,6,8) = 5

- دالة الانحراف المعياري (StdDev) تعيد قيمة الانحراف المعياري لمجموعة من قيم عددية، مثل:

StdDev (2,4,6,8,10) = 3.12

- دالة القيمة العشوائية (Rnd) تعطي قيمة عشوائية غير متوقعة، مثل:

Rnd (100) = 0.7055475

رابعا: دوال الوقت والتاريخ:

من هذه الدوال:

- الدالة (Time) تعيد الوقت الحالي لجهاز الحاسوب.
- الدالة (Timer) تعيد عدد الثواني منذ منتصف الليل (12:00 ص) وحتى هذه اللحظة.
- الدالة (Date) تعيد التاريخ الحالي لجهاز الحاسوب.

الباب التاسع: أدوات بيسك المرئي

يحتوي صندوق الأدوات في لغة فيجوال بيسك على مجموعة كبيرة من الأدوات التي تفيد في تصميم واجهات متميزة سهلة الاستخدام والتنفيذ خاصة باستخدام الفأرة، وهي من الطرق الحديثة في برمجة الحاسوب، وتسمى بالبرمجة الموجهة بالحدث (*Event driven programming*).

ومن أهم هذه الأدوات:

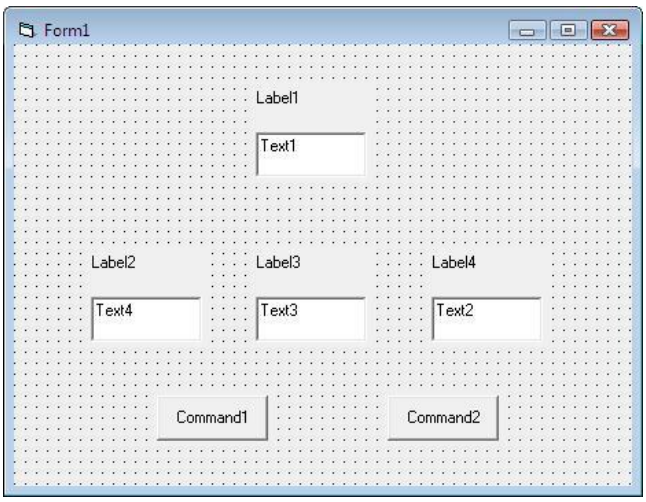
أ. زر الأمر (*Command button*):

يستخدم هذا الزر لتمكين المستخدم من إصدار أوامر أثناء تنفيذ البرنامج.

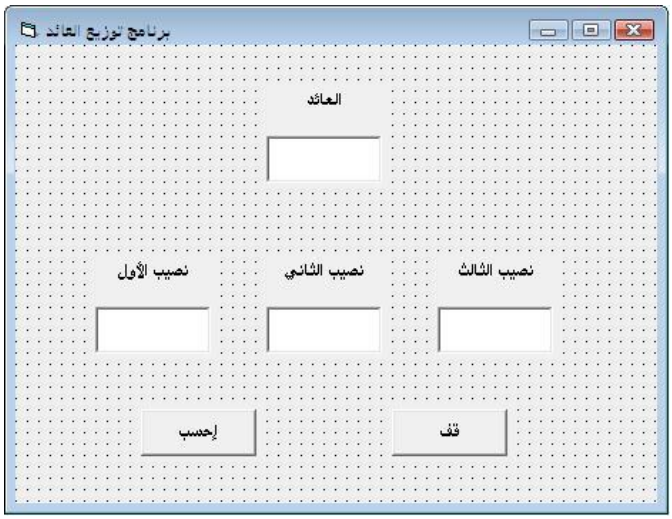
مثال①: أكتب برنامج لتوزيع عائد من مشروع اقتصادي على ثلاثة شركاء بنسبة 3: 2: 1؟

الحل:

أولا: تصميم نافذة البرنامج:



ثم ومن خلال مربع الخصائص نقوم بضبط كائنات النافذة كما في النافذة التالية:



ننقر على الزر المسمى (احسب) نقرا مزدوجا ونكتب البرنامج التالي:

```
Private Sub Command1_Click()
Rem هذا البرنامج مكتوب في الزر احسب
income! = Text1.text
Text2.text = income! / 2
Text3.text = income! / 3
Text4.text = income! / 6
End sub
```

ثم ننقر على الزر (توقف) نقرا مزدوجا ونكتب العبارة:

```
End
```

عند تنفيذ البرنامج واختبار وإدخال القيمة 600 كعائد للمشروع تظهر النافذة التالية:

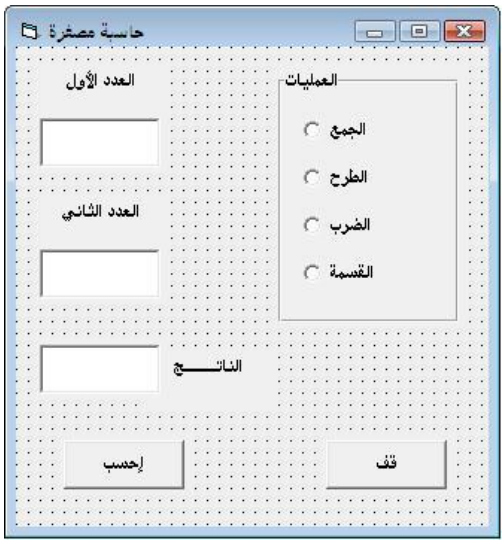
زر الاختيار (*Option Button*): Option1

يستخدم هذا الزر لعرض أكثر من خيار أثناء التنفيذ، فتظهر نقطة سواء على هذا الزر وذلك يعني أنه تم تحديد الخيار المقابل لهذا الزر.

مثال② : أكتب برنامج يعطي ناتج العمليات الحسابية الأربعة (+ - × /) لأي عددين يتم إدخالهما؟

الحل:

أولا : تصميم النافذة:



نكتب البرنامج التالي في الزر (احسب):

```
Private Sub Command1_Click()
Rem هذا البرنامج مكتوب في الزر احسب
Dim V1, V2, V3 as Single
V1 = Val (Text1.text)
V2 = Val (Text2.text)
If (Option1.value) then
      V3 = V1 + V2
Else
If (Option2.value) then
      V3 = V1 - V2
Else
If (Option3.value) then
      V3 = V1 * V2
Else
If (Option4.value) then
      V3 = V1 / V2
End if
End if
End if
End if
Text3.text = str(V3)
End sub
```

عند تنفيذ البرنامج واختباره ببعض القيم تظهر النافذة التالية:

ثالثا: مربع القائمة (List Box):

تستخدم هذه الأداة عند التعامل مع القوائم وخاصة إذا طلب من المستخدم اختيار بند واحد من مجموعة خيارات.

مثال③ : أكتب برنامج يعرض قائمة ببعض البلدان، بحيث تظهر مساحة البلد بالميل المربع بمجرد النقر على اسم البلد؟

الحل:

أولا تصميم النافذة:

ثانيا: لتعبئة القائمة بأسماء البلدان، ننقر على نافذة البرنامج (في جزء خالي) نقرا مزدوجا ونكتب البرنامج التالي:

```
Private Sub Form_Load()
Rem هذا البرنامج لتعبئة القائمة بالبلدان
List1.additem ("أفغانستان")
List1.additem ("تـــونـس")
List1.additem ("كنـــدا")
List1.additem ("مصـــــر")
End sub
```

بالنقر المزدوج على القائمة المرسومة على النافذة ونكتب البرنامج التالي:

```
Private Sub List1_Click()
Select case List1.listindex
Case 0
      Text1.text = 360000
Case 1
      Text1.text = 63378
Case 2
      Text1.text = 3831033
Case 3
      Text1.text = 363250
End select
End Sub
```

وعند تنفيذ البرنامج تظهر النافذة التالية:

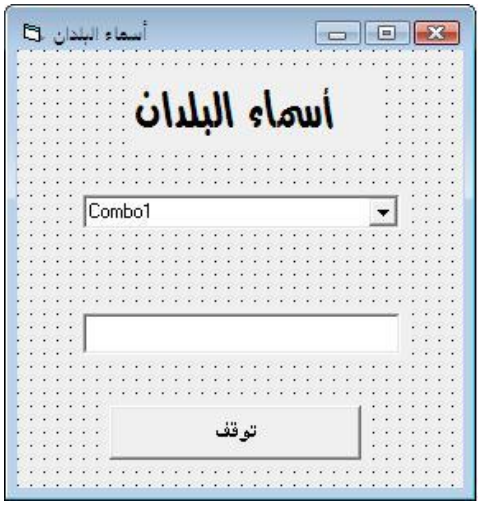
رابعا: مربع السرد (Combo Box):

وهو شبيه جدا بمربع القائمة إلا أنه يحتاج سطر واحد فقط على نافذة البرنامج ويستخدم في حالة عدم وجود مساحة كافية على النافذة.

مثال ④ : أكتب برنامج باستخدام مربع السرد يحتوي على اسم البلد باللغة العربية وعند اختيار البلد يعرض اسمها باللغة الإنجليزية في مربع نص موجود على النافذة؟

الحل:

أولا نقوم بتصميم نافذة البرنامج:



ثانيا: لتعبئة مربع السرد بأسماء البلدان، ننقر على نافذة البرنامج (في جزء خالي) نقرا مزدوجا ونكتب البرنامج التالي:

```
Private Sub Form_Load()
Rem هذا البرنامج لتعبئة القائمة بالبلدان
Combo1.additem ("لـــيبيـــا")
Combo1.additem ("تـــونـــس")
Combo1.additem ("مصـــــر")
Combo1.additem ("سوريـــا")
Combo1.additem ("الأردن")
End sub
```

بالنقر المزدوج على مربع السرد المرسوم على النافذة ونكتب البرنامج التالي:

```
Private Sub Combo1_Click()
Select case Combo1.listindex
Case 0
      Text1.text = "Libya"
Case 1
      Text1.text = "Tunisia"
Case 2
      Text1.text = "Egypt"
Case 3
      Text1.text = "Syria"
Case 4
      Text1.text = "Jordan"
End select
End Sub
```

وعند تنفيذ البرنامج تظهر النافذة التالية:

خلاصة الباب التاسع:

تم التعرض في هذا الباب للأدوات التالية:

ر.م	اسم القطعة	الوظيفة
1	Command Button	إصدار أمر من المستخدم إلى البرنامج.
2	Option Button	منح فرصة اختيار الخيار المناسب من بين مجموعة خيارات.
3	List Box	عرض قائمة دفعة واحدة تحتوي مجموع عناصر.
4	Combo Box	عرض قائمة عنصر تلو الآخر.
5	Text Box	عرض نص أو نواتج برنامج في مربع.
6	Label	عرض وصف ما على نافذة البرنامج.