# مقدمة في البرمجة

مقدمة:

جهاز الحاسب ينقسم على قسمين:
مكونات مادية ( Hardware )، وبرامج ( Software )، وهذه المكونات المادية كالشاشة والذاكرة والطباعة لا تقوم بعملها إلا بوجود برامج تقوم بإعطاء الأوامر لهذه القطع، فالطابعة لا يمكن أن تقوم بالطباعة إلا بوجود برنامج يرسل لها الأوامر، وهكذا...

يقوم بعمل هذه البرامج أفراد متخصصون في البرمجة، يقومون بصنع البرامج اللازمة لأداء عمل معين، وتتنوع البرامج إلى عدة أنواع: كأنظمة التشغيل مثل ( Windows )، والتطبيقات الأخرى: كالتطبيقات المحاسبة والمالية، وبرامج طبية، وبرامج عسكرية، وبرامج خاصة بالإنترنت: كالتصفح، والمحادثة، وغيرها من الأنواع الأخرى.

البرمجة:
عندما نقوم بكتابة برنامج بواسطة الحاسب فإن معنى ذلك أننا نقوم بإعطائه التعليمات والأوامر اللازمة لتنفيذ عمليات معينة، ومن هذا فإننا نستطيع تعريف برنامج الحاسب بأنه مجموعة من التعليمات والأوامر التي تعطى للحاسب للقيام بمهام مرتبة ومحددة.

وكما أن الطالب لا يستطيع أن يفهم ما يلقيه المعلم إلا بلغة يفهمها كاللغة العربية أو اللغة الإنجليزية مثلاً، فإن الحاسب كذلك لا يستطيع تنفيذ تلك التعليمات والأوامر إلا بعد أن تكون مكتوبة بإحدى اللغات التي يستطيع فهمها والتعامل معها، وكل لغة من هذه اللغات لها أوامرها وتعليماتها الخاصة بها ، ولكن جميعها تتفق في أن يقوم الحاسب بعمل ما من جرّاء هذه التعليمات، ويسمّى من يقوم بعمل هذه البرامج بـ ( المبرمج ).

ومن هنا يمكن تعريف البرمجة بأنها مجموعة من التعليمات التي يكتبها المبرمج بإحدى لغات البرمجة لأداء عمل معين، ويقوم الحاسب بتنفيذها.

أقسام لغات البرمجة
يمكن تقسيم لغات البرمجة إلى قسمين:

أولاً – اللغات البسيطة ( Low – Level Language ) وتنقسم إلى نوعين:
- لغة الآلة ( Machine Language )

وهي أول لغة تم وضعها للحاسب ، وهي اللغة الوحيدة التي يفهمها الحاسب ، والسبب في ذلك أنها مكونة من رمزين اثنين هما : ( الصفر والواحد )، فجميع تعليمات البرنامج في هذه اللغة مكونة من الصفر والواحد، فمثلاً لكتابة الحرف ( L ) نقوم بكتابة الرموز ( ٠١٠١١٠٠٠ ) كشفرة للدلالة على الحرف ( L )، وهكذا...

ومن هنا فإن البرمجة بلغة الآلة تعتبر سهلة الفهم بالنسبة للحاسب ولكنها صعبة الفهم والقراءة بالنسبة للمبرمجين ومستخدمي الحاسب، ولذلك فإن البرمجة بها حالياً تقتصر على الشركات الصانعة لأجهزة الحاسب.

- اللغة الرمزية أو لغة التجميع ( Symbolic Language )

وهي لغة قريبة من لغة الآلة نوعاً ما ، إذ إنها تستخدم بعض الرموز الخاصة ، والتي يمثل كل رمز منها تعليمة أو أمراً له غرض بالبرنامج ،وهذه الرموز مكتوبة بحروف اللغة الإنجليزية مما يجعل المبرمج أكثر فهماً وأسهل قراءة لها من لغة الآلة ، فمثلاً نستعمل الرمز ( ADD ) لعملية الجمع، والرمز ( SUB ) لعملية الطرح. وهكذا...
والبرمجة بهذه اللغة تعتبر صعبة بالنسبة لمستخدمي الحاسب والمبرمجين المبتدئين، وذلك لأنها تتطلب فهماً دقيقاً لمحتويات الحاسب الداخلية، ولذا فإن تعلمها يقتصر على المبرمجين المحترفين الذين يرغبون في كتابة برامج سريعة.

ثانياً اللغة العالية ( High – Level Language ) :

وهي لغات قريبة من لغات الإنسان، وهذا هو سبب تسميتها باللغات العالية، وهذه اللغات العالية المستوى لا بد أن يتم تحويل برامجها بعد كتابتها إلى لغة الآلة حتى يتسنى للحاسب فهمها، وتتم عملية التحويل بواسطة مترجم خاص الحاسب ( Compiler ) يقوم بترجمة تلك اللغة إلى لغة الآلة، ولذا فإن لكل لغة من هذه اللغات مترجماً خاصاً، وتنقسم لغات البرمجة العالية إلى قسمين:

- لغة البرمجة الإجرائية ( Procedural Language )

وتسمى أحياً بالبرمجة التقليدية، وتعتمد لغة البرمجة الإجرائية على قيام المستخدم لها بكتابة البرنامج مفصلاً إجراءً بعد إجراء، وبالتالي يحدد المستخدم بدقة للحاسب التعليمات التي تؤدي إلى قيام الحاسب بالمهام المطلوبة للبرنامج خطوة بعد أخرى.
ويوضح الجدول التالي أهم هذه اللغات ومجال استخدام كل منها:

| اللغة | مجالات الاستخدام |
| --- | --- |
| بيسك ( Basic ) | المدارس والجامعات والمبتدئين في اللغة |
| باسكال ( Pascal ) | المجالات الإدارية والعلمية |
| فورتران ( Fortran ) | المجالات الهندسية والعلمية |
| كوبول ( Cobol ) | التطبيقات الإدارية والتجارية |
| سي ( C ) | لكتابة برامج أنظمة تشغيل الحاسب وبرامج التطبيقات |

منتديات الفارس التطويري
www.f55er.com

- لغة البرمجة بالعناصر ( Object Oriented Language )

تسمى أحياناً بالبرمجة المرئية أو البرمجة المسيّرة بالأحداث ( Event-driven Programming )، وتعتبر هذه اللغة لغة تطويرية تعتمد على قواعد اللغات التقليدية لتتمكن من العمل على أنظمة التشغيل ذات البيئة الرسوميّة، مثل ( Windows XP )، فمثلاً لغة بيسك طورت إلى فيجول بيسك ( Visual Basic أي بيسك المرئية ، لغة باسكال طورت إلى ديلفي، ولغة سي طورة إلى فيجول سي ( Visual C )، وهي لغات ذات بيئة تطوير أو تصميم متكاملة ( Integrated Design Environment )، أو اختصاراً ( IDE ).

ولا تتطلب البرمجة بالعناصر من المستخدم كتابة البرنامج على شكل خطوات إجرائية محددة أو كتابة أوامر وتعليمات متتابعة، وإنما يقوم المستخدم بوضع الكائنات أو العناصر على النموذج، والتي تمكن المستخدم من إدخال البيانات وإخراجها وتحديد الأحداث التي تتم على هذه العناصر، وتقوم لغة البرمجة بناءً على ذلك بإنشاء التعليمات والأوامر تلقائياً وتنفيذها.

وختاماً نذكر أن هذا الجيل من لغات البرمجة والذي يطلق عليه لغات البرمجة بالعناصر أو المرئية ليس خاتمة المطاف، حيث يسعى علماء الحاسب مستقبلاً لإيجاد الجيل الخامس من لغات البرمجة وهو ما يسمى بلغات الذكاء الاصطناعي، والتي ستكون لغات طبيعية كالتي يتخاطب بها البشر فيما بينهم، مثل اللغة العربية أو اللغة الإنجليزية، بحيث يمكنك مخاطبة الحاسب وإعطاؤه الأوامر والتعليمات للبرنامج من خلال الحديث مع جهاز الحاسب مباشرة، والذي سيكون قادراً على إدراك وتفهم عباراتك وأوامرك، كأن تقول له ( أكتب ) أو ( اعمل جدولاً )، أو ( احسب نتيجة الضرب لمعادلة جبرية )، أو تقوم بإملاء خطاب عليه، ونحو ذلك، وصدق الله القائل: ( وما أوتيتم من العلم إلا قليلاً )

التعامل مع البيانات:

قبل أن نتطرق إلى كتابة البرنامج بواسطة إحدى لغات البرمجة لابد لنا أن نتطرق إلى كيفية تعامل الحاسب مع البيانات، فالحاسب يتعامل مع أنواع مختلفة من البيانات

( أرقام – حروف – رموز – تاريخ ... ) ، ولأن هذه البيانات المدخلة إلى الحاسب تحتاج إلى معالجة لتخرج لنا في النهاية على شكل معومات يمكننا الاستفادة منها، ف'ننا نحتاج إلى استخدام أماكن معينة في ذاكرة الحاسب لحفظ وتخزين هذه البيانات ( الثوابت ) حتى تتم معالجتها، ويطلق على هذه الأماكن اسم متغيرات.
بعد الحصول على هذه البيانات وحفظها نحتاج إلى إجراء العمليات عليها، خاصة البيانات العددية والتي هي مجال الحصول على هذه البيانات وحفظها نحتاج إلى إجراء بعض العمليات عليها، خاصة البيانات العددية والتي هي مجال الحاسب الأساسي ، ومن هذه العمليات التي تجرى على البيانات العمليات الحسابية ( الجمع – الطرح – الضرب – القسمة والأسس – وغيرها ) وعمليات المقارنة ( > ، < ، = ، ..... ) وسوف نتعرف على طريقة إجراء هذه العمليات المختلفة باستخدام لغة فيجول بيسك.

المتغيرات والثوابت

تتعامل لغة فيجول بيسك مع مجموعة مختلفة من البيانات ( أرقام – حروف – رموز – تاريخ ... )، ولتخزين هذه البيانات فإن لغة فيجول بيسك تحتاج إلى أماكن تختزن هذه البيانات ليتم عليها إجراء العمليات المختلفة، وسنحتاج في دراستنا لهذا الجزء من البرمجة إلى التعامل مع هذه البيانات بأنواعها الثوابت والمتغيرات، ومن هذا فإنه يمكن تعريفها كالآتي:

| الثوابت | المتغيرات |
| --- | --- |
| وهي البيانات التي يتعامل معها البرنامج، وهي قيم ثابتة لا تتغير أثناء عمل البرنامج، وهذه الثوابت تنقسم إلى عدة أقسام، منها:<br><br>١. ثابت عددي.<br>٢. ثابت حرفي. | وهي أماكن يتم فيها تخزين قيم يمكن أن تتغير أثناء عمل البرنامج، وهذه المتغيرات تنقيم إلى عدة أقسام، منها:<br><br>١. متغير عددي.<br>٢. متغير حرفي.<br>٣. متغير منطقي. |
| منتديات الفارس التطويري<br>www.f55er.com | |

ولا بد من تسمية المتغيرات حتى يتمكن البرنامج من التعرف عليها.
ويشترط في تسمية المتغير في لغة فيجول بيسك شروط، هي:

- أن تبدأ التسمية بحرف من الحروف الأبجدية باللغةالإنجليزية.
- أن لاتتجاوز التسمية ٢٥٥ حرفاً
- أن لا تحتوي التسمية على بعض الرموز الخاصة ( + ،- ، / ، .... ) أو فراغ.

- أن لا تكون التسمية إحدى الكلمات أو الأوامر التي تستخدمها لغة البرمجة التي نتعامل معها ، مثل ( DIM ، CASE )

وبالتالي فإن الأسماء التالية صالحة للمتغيرات ( Mr2, SName, Hasub, A, B )
أما الأسماء التالية فهي غير صالحة لتسمية المتغير:

- ( 2Mr ): وذلك لأنها بدأت برقم ولم تبدأ بحرف.
- ( AM RE ): وذلك لأنها احتوت فراغ.
- ( DIM ): وذلك لأنها كلمة تستخدمها لغة فيجول بيسك.

تحديد نوع البيانات في المتغيرات
لتعريف المتغيرات نحتاج إلى تحديد نوع البيانات التي سوف تخزن فيها، ويوضح الجدول التالي بعض أنواع البيانات التي تتعامل معها المتغيرات، وطرق تعريف المتغيرات في البرنامج:

| نوع البيانات | الحجم | طرق التعريف في البرنامج | مثال القيمة المخزنة |
|---|---|---|---|
| عدد صحيح ( Integer ) | ٢ بايت | Dim Mr As Integer | Mr = 34 |
| عدد صحيح طويل ( Long ) | ٤ بايت | Dim Gb As Long | Gb = 15200000 |
| عدد عشري ( Single ) | ٤ بايت | Dim X2 As Single | X2 = 13.44 |
| عدد عشري مضاعف ( Double ) | ٨ بايت | Dim Qdm As Double | Qdm = 3.1415926 |
| العملة ( Currency ) | ٨ بايت | Dim Deb As Currency | Deb = 852.24 |
| سلسلة نصية ( String ) | بايت لكل حرف | Dim Name As String | Name = " Sara " |
| منطقي ( Boolean ) | ٢ بايت | Dim Bl As Boolean | Bl = true |
| تاريخ ( Date ) | ٤ بايت | Dim Bl As date | Dl = #02-05-08# |
| متنوع ( Variant ) | ١٦ بايت للأرقام<br>٢٢ بايت + بايت لكل | Dim Nk As Variant | Nk = 45.12 |

منتديات الفارس التطويري
www.f55er.com

ملاحظة

- التعريف من خلال التصريح المتغير : مثل ( Dim Nk as Variant )
- لتخزين البيانات داخل متغير سلسلة نصية لا بد أن تكون هذه البيانات محصورة بين علامتي اقتباس (" ") مثل: "أحمد" =Name
- لا بد من تحديد نوع البيانات بشك دقيق، فمثلاً لا يمكن أن نضع عدداً عشرياً ( ٤ ، ٥ ) في متغير من نوع عدد صحيح طويل ( Long ) ، إنما لا بد أن يكون من نوع ( Single ) أو ( Double ).

- لكل متغير سعة تخزينية ( حجم ) ، وهذه السعة قد تؤثر عندما يكون لديك عدد كبير من البيانات المخزنة ، مما قد يبطئ عمل البرنامج ويستهلك مساحات التخزين.
- نستخدم نوع البيانات ( Variant ) إذا لم يحدد نوع البيانات بشكل دقيق، حيث يمكن أن يحتوي على أي نوع من البيانات ما عدا نوع السلسلة النصية ( String ).
- عند تعريف متغير يمكن أن نستخدم رموزاً بدل اسم النوع ، فمثلاً بدل Integer نستخدم % ويكون التعريف كالآتي ( Dim Mr% )، حيث إن هذه الرموز كانت تستخدم في لغة بيسك الأساسية ولكن من الأفضل استخدام التعريف المباشر ( Dim Mr As Integer ).

العمليات الحسابية

ويقصد بها عمليات: الجمع، والطرح، والضرب، والقسمة، ويوضح الجدول التالي هذه العمليات الحسابية والعاملات الخاصة بها:

| العملية الحسابية | العامل | الشرح | المثال |
|---|---|---|---|
| الجمع | + | لجمع عددين من أي نوع | 50 + 5 = 55 |
| الطرح | - | لطرح عددين من أي نوع | 40 - 8 = 32 |
| الضرب | * | لضرب عددين من أي نوع | 5 * 6 = 30 |
| القسمة | / | لقسمة عددين والناتج ذو فاصلة | 10 / 4 = 2.5 |
| الأس | ^ | الأس لعدد | 3^3 = 23 |

منتديات الفارس التطويري
www.f55er.com

أولوية تنفيذ العمليات في الحاسب

- الأقواس: حيث يقوم الحاسب بتنفيذ العملية المحصورة بين الأقواس مهما كانت العملية.
- عمليات الأس
- عمليات الضرب والقسمة: وهما بنفس الدرجة، ويتم تنفيذ العمليات من اليسار إلى اليمين، فمثلاً:

في المثالين السابقين يتبين أن الضرب والقسمة متساويا الأولوية، والتنفيذ يكون من اليسار لليمين.

- عمليات الجمع والطرح: وهما بنفس الدرجة ويتم تنفيذ العمليات من اليسار إلى اليمين فمثلاً:

أولوية الأمثلة على أولوية تنفيذ العمليات في الحساب

عوامل المقارنة:

ويقصد بها عوامل المقارنة بين عددين أو سلسلتين من الحروف لنعرف : إن كانا متساويين، أو غير متساويين، أو إن كان أحدهما أكبر من الأخر، وهكذا، ويوضح الجدول عوامل المقارنة المستخدمة في لغة فيجول بيسك.

| العامل | معناه |
| --- | --- |
| = | يساوي |
| < > | لا يساوي |
| > | أكبر من |
| < | أصغر من |
| > = | أكبر من أو يساوي |
| < = | أصغر من أو يساوي |

منتديات الفارس التطويري
www.f55er.com

ملاحظة

وكما لا حظت فإن ناتج العمليات الحسابية التي درسناها سابقاً عبارة عن عدد يمثل حاصل العملية. أما الناتج في عمليات المقارنة فهو إما صحيح ( True ) أو خطأ ( False ).

مثال ١

ما نتيجة عملية المقارنة التالية : 8 > 5

الحل

هي نتيجة صحيحة وذلك لأن العدد ( 8 ) فعلاً هو أكبر من العدد ( 5 ).

منتديات
الفارس التطويري
www.f55er.com

والفائدة من استخدام المقارنات التحقق من شرط معين داخل البرنامج، كما سندرس في الباب الثالث ( البرمجة بلغة فيجول بيسك ) _ إن شاء الله

تحويل التعابير الجبرية:
عندما يكون لدينا مسألة أو تعبير جبري ونريد حلها بواسطة الحاسب لابد أن تكتب بطريقة يفهمها الحاسب ، باستخدام العمليات الحسابية وعمليات المقارنة التي مرت معنا.

بالتوفيق والسعادة في الدارين الدنيا والآخرة
مع تحيات
منتديات الفارس التطويري
www.f55er.com