

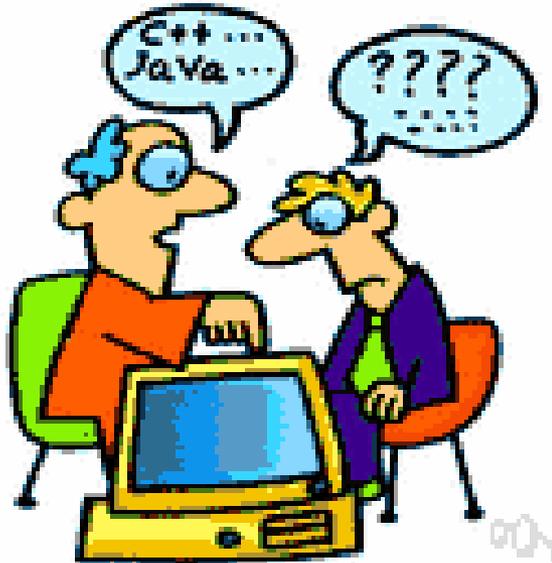
جامعة حماة

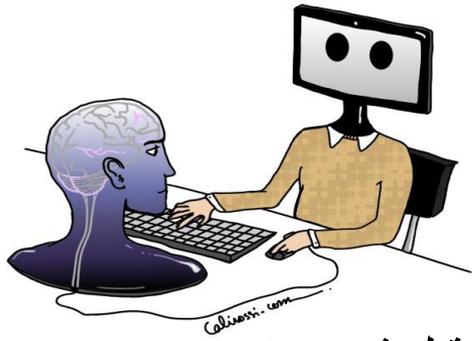
كلية العلوم في مصيف / السنة الأولى

المادة: لغات البرمجة – العام الدراسي ٢٠١٨ - ٢٠١٩

المحاضرة الثانية: الحاسب ولغة البرمجة باسكال

Computer & Pascal Programming Language





الحاسوب والبرمجة

- استُخدم الحاسب في تنفيذ العمليات الحسابية بشكل أساسي بعد ظهوره ومن قبل فنيين متخصصين في بعض العلوم فقط ، لكن سرعان ما اكتشف الإنسان إمكانية استخدام هذا الجهاز في كافة مجالات الحياة.
- فقد تبين للإنسان أن هذا الجهاز قادر على تخزين كميات هائلة من البيانات التي يمكن الوصول إليها بسرعة فائقة وبدون أخطاء، مما يسهل الوصول إلى المعلومة بشكل سريع وبأقل كلفة مادية.
- إن تحضير المعطيات والبيانات التي نريد إدخالها إلى الحاسب تعد عملية حساسة وهامة جداً، لأنها تعكس حالة واقعية من حياتنا اليومية، لذلك الخطوة الهامة الأولى الواجب علينا القيام بها هي التدقيق في اختيار شكل وآلية تمثيل تلك المعطيات في الحاسب بما يتناسب مع أرض الواقع وحيثيات وقواعد لغة البرمجة المستخدمة.

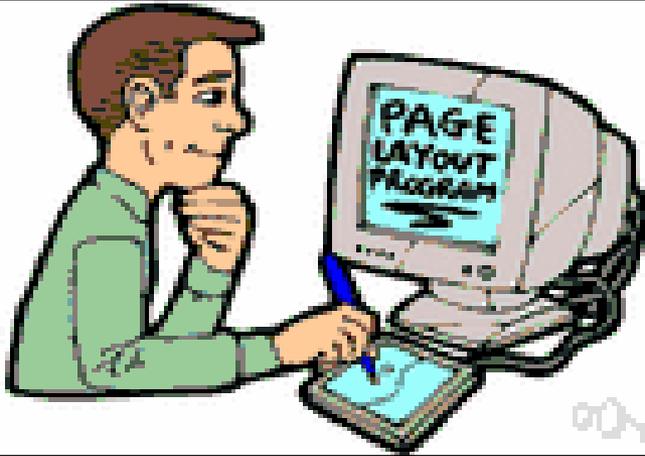


الحاسوب والبرمجة

• أهم مزايا الحاسوب التي يستفيد منها المبرمج في حل المشكلات:

١. **السرعة:** يستطيع الحاسوب إنجاز العمليات في وقت قصير جداً مقارنة بالوقت الذي يحتاجه الإنسان لإنجاز هذه العمليات بنفسه.
٢. **التكرار:** يتميز الحاسوب بالقدرة على تنفيذ العمليات أو الأوامر لعدة مرات يحددها المبرمج وتستفيد من الميزة الأولى لإتمامها بسرعة.
٣. **الدقة:** الحاسوب يستطيع تنفيذ العمليات الحسابية المعقدة بدقة متناهية.
٤. **التخزين:** قدرة الحاسوب على تخزين البيانات لاسترجاعها في وقت لاحق، ويتميز بحجم البيانات الكبير القادر على تخزينه والسرعة والدقة في استرجاعها.

العمليات التي يقوم بها الحاسب



١- إدخال البيانات

٢- تخزين البيانات في ذاكرة الحاسب

٣- إجراء العمليات الحسابية على البيانات

٤- إجراء العمليات المنطقية على البيانات

٥- تكرار تنفيذ أمر عدة مرات

٦- اخراج النتائج

لغات البرمجة Programming Languages

- لغة البرمجة: هي لغة تخاطب الإنسان مع الحاسب للتحكم به عن طريق أوامر وتعليمات يكتبها الإنسان، فيقوم الحاسب بترجمتها إلى لغة الآلة ثم ينفذها بدقة وبسرعة عالية جداً.
- لغات البرمجة عديدة جداً .. فهناك لغات برمجة لم تعد مستخدمة حالياً بسبب ظهور لغات أحدث منها قادرة على انجاز العمل بشكل أفضل وأسهل ...
- تطورت لغات البرمجة واستفادت من بعضها البعض لأنها متشابهة من الناحية الهجائية (أوامر اللغات والكلمات التي يفهمها الحاسب متشابهة في كل لغات البرمجة) مما يجعل الانتقال من لغة إلى لغة أمر في غاية السهولة. لذلك من يتقن لغة برمجة واحدة جيداً يمكنه تعلم لغات أخرى.
- كل لغات البرمجة سواء كانت قديمة أو حديثة لا بد أن تتحدث عن وتتطرق إلى المواضيع التالية: (المتغيرات / جمل الشرط والتحكم / جمل التكرار / الدوال / المصفوفات)

Types of Programming Languages أنواع لغات البرمجة

مستويات لغات البرمجة

لغات برمجة عالية المستوى
High Level Languages

لغات برمجة متدنية المستوى
Low Level Languages

١. لغات البرمجة الإجرائية Structural Programming
٢. لغات البرمجة المرئية Visual Programming

١. لغة الآلة Machine Language
٢. لغة التجميع Assembly Language



لغات البرمجة Programming Languages

• **لغات منخفضة المستوى / البسيطة Low Level Languages**: وتقسم إلى نوعين:

١- لغة الآلة **Machine Language**: وهي اللغة الوحيدة التي يفهمها الحاسب ؛ والسبب في ذلك أنها مكونة من رمزين اثنين هما (الصفر والواحد)

٢- لغة التجميع **Assembly Language**: هي لغة قريبة من لغة الآلة نوعاً ما، تستخدم بعض الرموز الخاصة، تعتبر صعبة بالنسبة لمستخدمي الحاسب والمبرمجين المبتدئين ، وذلك لأنها تتطلب فهماً دقيقاً لمحتويات الحاسب الداخلية (لغة الأسيمبلي).

• **لغات عالية المستوى High Level Languages**: لغات قريبة من لغات الإنسان ، وهذا هو سبب تسميتها باللغات العالية، وتتنقسم إلى نوعين:

١- لغات البرمجة الإجرائية **Structural Programming Languages**: تعتمد لغات البرمجة الإجرائية على قيام المستخدم بكتابة البرنامج مفصلاً إجراءً بعد إجراء، مثال عليها : لغة البيسك Basic.

٢- لغات البرمجة المرئية **Visual Programming Languages**: تعتبر هذه اللغات تطويرية تعتمد على قواعد اللغات التقليدية لتتمكن من العمل على أنظمة التشغيل ذات البيئة الرسومية Graphics مثال عليها : الفيچول بييسك Visual Basic والفيچوال سي C #، Visual C++ والديلفي Delphi

أمثلة على لغات البرمجة

• يجب الإشارة إلى أهمية اختيار لغة البرمجة القادرة على تمثيل البيانات بشكل أبسط وأنسب، حيث نلاحظ توافقية لغة برمجية معينة لنوع معين من العمليات التي نود تنفيذها على تلك البيانات أكثر من لغة أخرى، وبقدر ما يكون التمثيل أبسط ، يصبح البرنامج قادر على معالجة مجموعة أكبر من المسائل.

- لغة فوتران Fortran: تستخدم في التطبيقات العلمية والهندسية والرياضية التي تشتمل على عمليات حسابية معقدة.

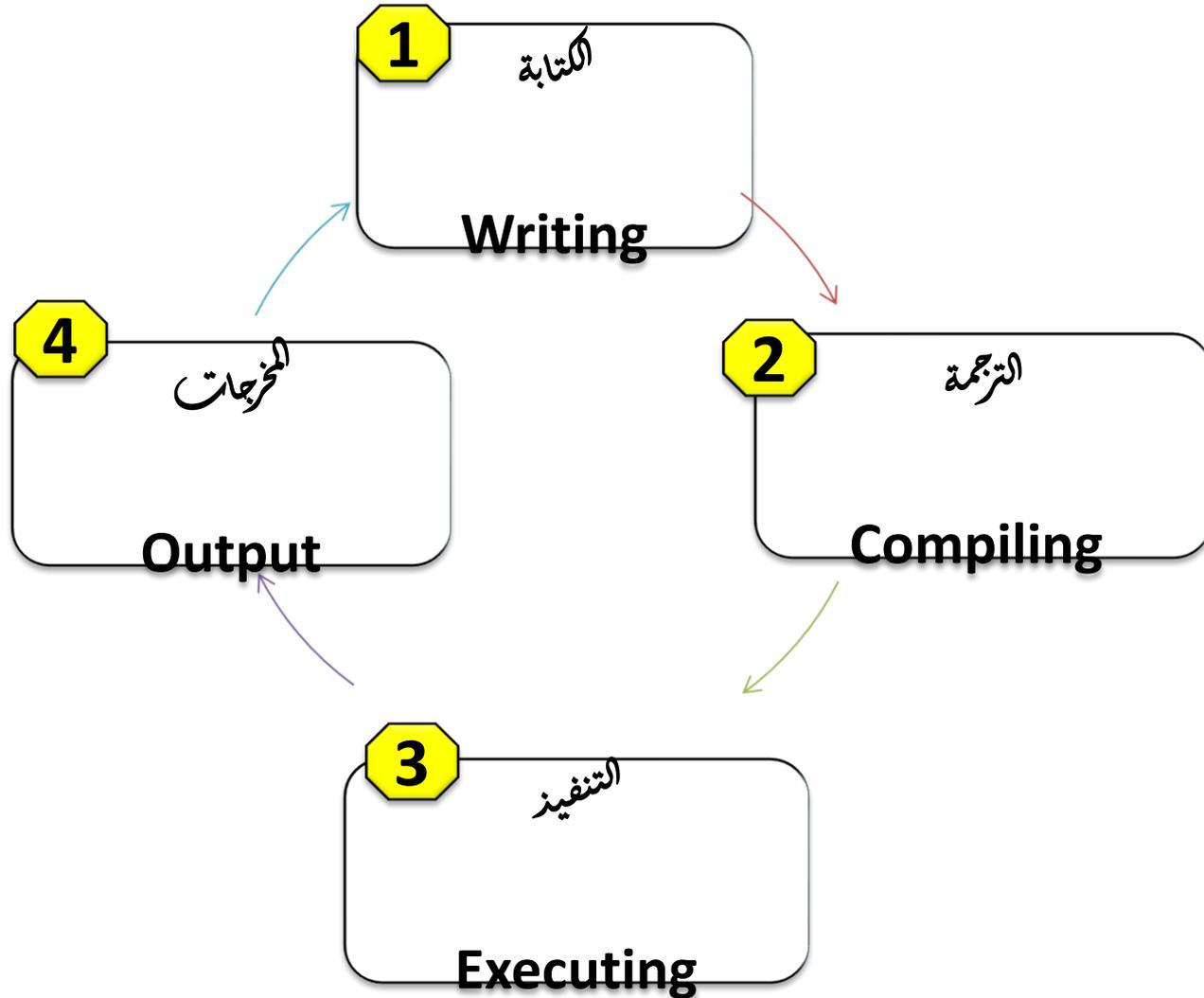
- لغة كوبول Cobol: تستخدم في التطبيقات التجارية.

- لغة باسكال Pascal: أفضل لغة تعليمية لطلاب الجامعات ومعاهد التعليم العالي.

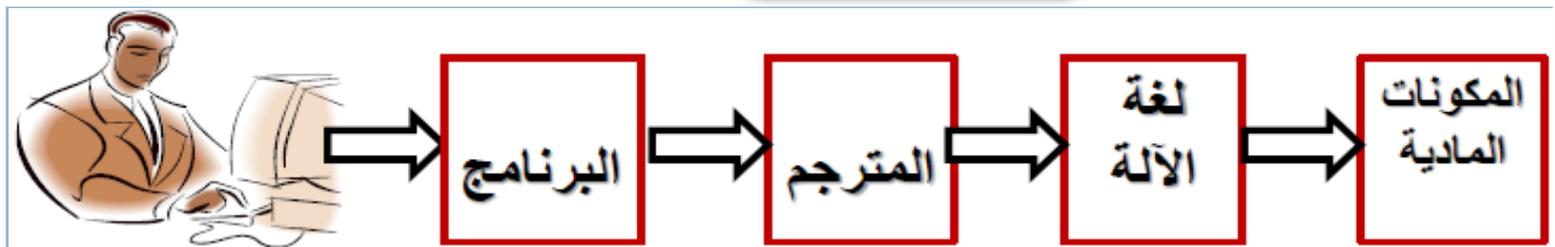
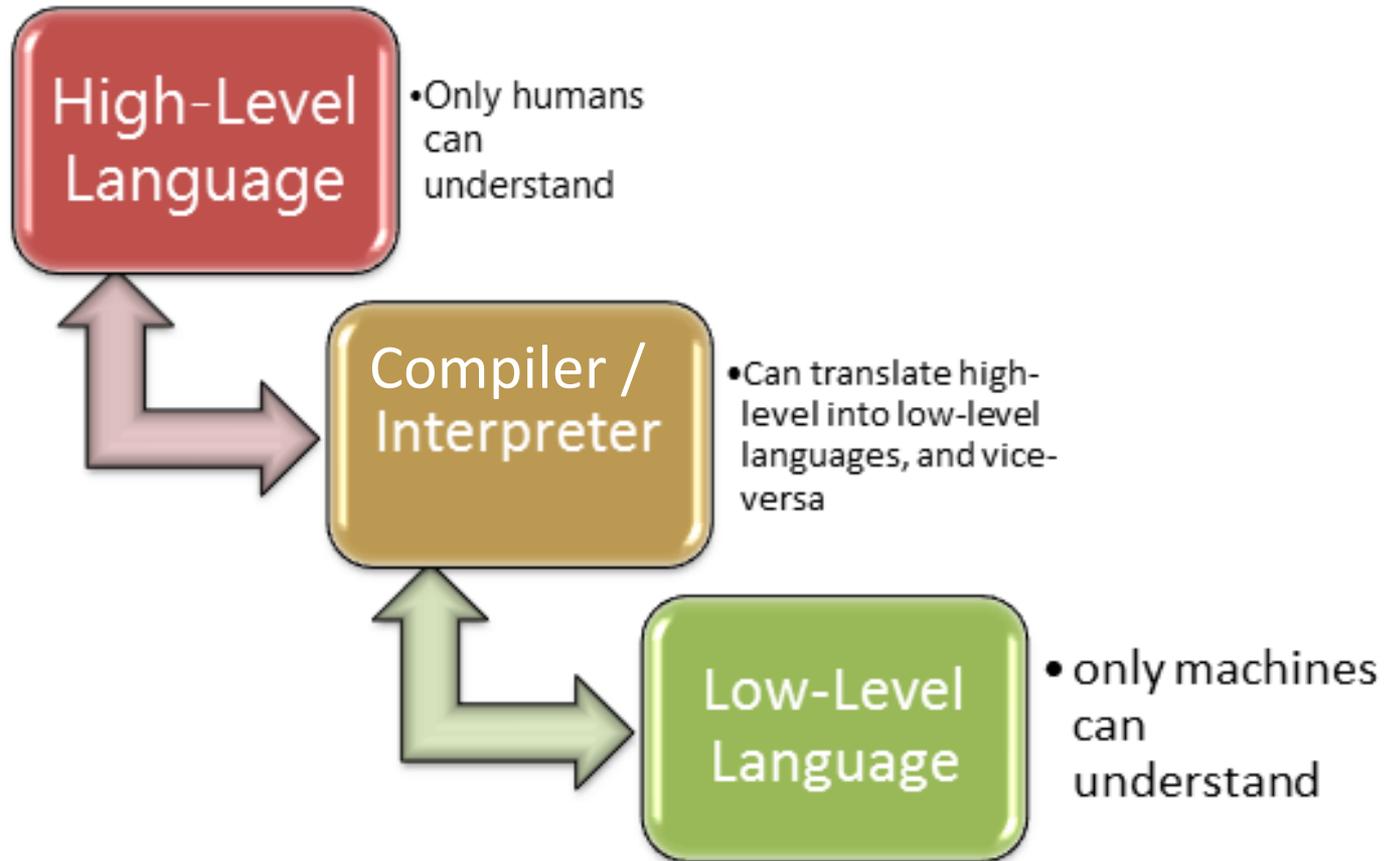
- لغة سي C: هي الأفضل لكتابة برمجيات النظم، حيث نجد أن معظم نظم التشغيل مكتوبة بهذه اللغة التي تمتاز بقدرتها على التحكم المباشر في المكونات المادية للحاسب، والاستفادة المثلى من إمكانيات الحاسب. ويمكن استخدام هذه اللغة في تطوير لغة برمجة أخرى.

- لغة فيجوال بيسك Visual Basic: تستخدم في أغراض متعددة للمبتدئين، وأصلها لغة بيسك Basic تم تطويرها وتزويدها بواجهة مرئية كما يسميها البعض (لغة بيسك المرئية) وتتميز بسهولة تعلمها وسهولة استخدامها.

بناء البرامج



مراحل تنفيذ البرنامج



المفسر (Interpreter) والمترجم (Compiler)

- برنامجي المفسر والمترجم الخاصين بلغات برمجة الحاسب يقومان بترجمة اللغات العالية إلى لغة الآلة ليتمكن جهاز الحاسب من تنفيذ التعليمات والأوامر بسرعة عالية جداً.
- **برنامج المفسر Interpreter:** يقوم بترجمة أوامر وتعليمات البرنامج المكتوب بإحدى لغات البرمجة سطراً سطراً إلى لغة الآلة أثناء قرائنها ليتم تنفيذها سطراً سطراً أيضاً. ويمتلك المفسر القدرة على التفاعل مع البرنامج أثناء تنفيذه وإجراء أي تغيير في البرنامج ثم متابعة التنفيذ.
- **برنامج المترجم Compiler:** يقوم بترجمة أوامر وتعليمات البرنامج المكتوب بإحدى لغات البرمجة ككتلة واحدة إلى لغة الآلة أثناء قرائنها، ليتم تنفيذها دفعة واحدة، ويعتبر أسرع من برنامج المفسر Interpreter بعدة مرات في تنفيذ البرنامج، ولكن عند وجود أي خطأ في البرنامج لابد من إصلاح الخطأ وإعادة تنفيذ كل البرنامج من جديد.

مترجمات باسكال Pascal Compilers

- يوجد العديد من المترجمات التي تعمل على باسكال ، ومن أشهرها Free Pascal وهناك أيضا نسخه Borland Pascal وهناك نسخة التريو باسكال Turbo Pascal من شركة مايكروسوفت، و
- سنتعمد على إحدى تلك المترجمات في هذا المقرر.
- تختلف المترجمات عن بعضها البعض في طرق كتابة بعض الدوال وتنفيذها ، لأن كل مترجم له مكتباته ودواله الخاصة .
- الفكرة البرمجية هي نفسها في جميع المترجمات وليس في لغة باسكال فقط، بل إن جميع لغات البرمجة لها فكره برمجية واحدة وإن اختلفت طرق كتابه الكود من لغة لأخرى.



مقاطع الذاكرة Memory Segment's

يقوم مترجم باسكال Compiler بتقسيم ذاكرة الحاسب إلى المقاطع التالية:

- **مقطع الترميز Coding Segment**: لتخزين نص البرنامج بعد تحويله إلى لغة الآلة (0 , 1).
- **مقطع المعطيات Data Segment**: لتخزين قيم المتحولات المعرّف عنها بـ Var.
- **مقطع المكدس Stack Segment**: لتخزين قيم المتحولات المصرّح عنها في الدوال والإجراءات .
- **الكومة Heap**: القسم الحر من ذاكرة RAM، ويمكن أن يستخدم لتخزين قيم المتحولات الديناميكية فقط.
- تجدر الإشارة إلى وجود مجموعة من الدوال والإجراءات الخاصة بالتعامل مع ذاكرة الحاسب حيث تمكننا من التحكم ببعض العمليات ضمن الذاكرة بمختلف مقاطعها.

لغة البرمجة باسكال Pascal

- **لغة البرمجة "باسكال Pascal":** هي لغة تخاطب بين الإنسان والحاسب ولها قواعدها الخاصة، وتتكون من مجموعة الأوامر والتعليمات المكتوبة باللغة الإنكليزية حيث يقوم الحاسب بترجمتها إلى لغة الآلة ثم يقوم بتنفيذها.
- يقوم الإنسان باستخدام أوامر لغة باسكال للتعبير عن حل أية مشكلة تواجهه بمساعدة الحاسب.
- هناك عدة نسخ متتالية الإصدارات من لغة باسكال لكي تواكب التطورات المتسارعة في عالم البرمجيات والحاسب. وبرنامج الباسكال هو مجموعه من الأوامر بعضها قابل للتنفيذ Executable وبعضها غير قابل للتنفيذ Non-Executable .
- تعود قصة الرياضي والفيلسوف الفرنسي بليز باسكال إلى منتصف القرن السادس عشر ميلادي عام ١٦٤٢م حيث أن هذا الشاب والذي كان يبلغ الثامنة عشر ربيعا والذي كانت بواذر العبقرية والاختراع تبدو جلية على أعماله وحيث أنه كان شديد الإبداع في علم الرياضيات مولعا بالاكشافات العلمية وتطبيق النظريات الرياضية ففي نفس السنة اخترع أول حاسب نصف آلي وسمي (حاسب باسكال ١٦٤٢)، وكان ذلك لسبب أساسي وهو مساعده أبيه الذي كان يعمل في مؤسسة الضرائب محصلا للفواتير والذي كان يقضي معظم لياليه مستخدما العد اليدوي في إحصاء وتدقيق حسابات المبالغ التي حصلها وقد كان يشكل هذا النوع البطيء من الحساب إرباكا لعائلته، وبالتالي يأخذ منه الوقت الكبير.

الشكل العام (القالب) لبرنامج باسكال Pascal

تتألف الأوامر من كلمات دلالية أو تعريفية يجب على المبرمج تعريفها في بداية كل برنامج ولتوضيح ذلك نرى مخطط لبرنامج باسكال:

Program The Total (input, output);

{This program find sum of two integer number }

تعريف باسم البرنامج الذي
يلي كلمة Program و أدوات
الإدخال والإخراج

Var

A, B, Total: integer;

منطقة التعريف للمتغيرات والثوابت والدوال

Begin

Write ('Enter two integer numbers to be add: ');

Readln (a, b);

Total: = a+b;

Writeln ('The sum of', A, 'and', B, 'is', Total);

End.

جسم البرنامج المحصور بين
Begin و End الذي يحوي
على الأوامر التنفيذية

أقسام برنامج باسكال Pascal

- نلاحظ برنامج باسكال يتكون من ثلاثة أجزاء رئيسية هي:

١. الجزء الأول: التعريف باسم البرنامج ووسيلة الإدخال والإخراج ونشاهد كلمة أساسية وهي Program يتبعها اسم البرنامج Name ويجب أن يكون دال على عمل البرنامج وبعدها أدوات الإدخال والإخراج; (input, output) متبوعة بفاصلة منقوطة.

٢. الجزء الثاني: قسم التعاريف Var وهو خاص بتعريف الثوابت Constants والمتغيرات Variables والدوال Functions التي سنستخدمها داخل جسم البرنامج.

٣. الجزء الثالث: جسم البرنامج مخصص لكتابة الأوامر الخاصة بلغة باسكال ومحصورة بين كلمتي Begin و End. متبوعة بـ (.) تدل على نهاية البرنامج.

- ملاحظة: يجب الانتباه لوضع العلامة (;) الفارزة المنقوطة بعد كل سطر برمجي مكتوب بلغة باسكال ما عدا End الموجودة في نهاية البرنامج تكون متبوعة بالنقطة كما أسلفنا والكلمات Var و Begin تكون غير متبوعة بشئ.

التعريف باسم البرنامج (تعليمة Program)

• كل برنامج مكتوب بلغة باسكال يجب أن يبدأ بتعليمة Program ويلى التعليمة اسم البرنامج (التسمية اختيارية)، ويجب أن يكون الاسم ذو دلالة على محتوى البرنامج مثل (الجمع sum، الطرح minus، الضرب multiple، القسمة division، العدد الأكبر max، العدد الأصغر min، المتوسط avg..... إلخ)، وبعدها وسيلة الإدخال والإخراج ضمن قوسين (input , output) وفي نهاية السطر البرمجي فاصلة منقوطة (;).

• Program sum (input , output) ;

• يمكن كتابة وسيلة الإدخال والإخراج (i , o) وبعدها فاصلة منقوطة (;) :

• Program sum (i , o) ;

• يمكن إهمال كتابة وسيلة الإدخال والإخراج وكتابة فاصلة منقوطة (;) بعد اسم البرنامج مباشرة :

• Program sum ;

تعريف المتغيرات / المتحولات Variables (تعليمة Var)

- ❖ المتغيرات Var هي أسماء أماكن يتم حجزها مسبقاً في ذاكرة الحاسب (أحجام محددة) ويحدد أنواع أنواع البيانات التي ستخزن فيها ليتم معالجتها والاستفادة منها في إنجاز عمل معين، ويتم تسمية هذه المواضيع باسم تعريفي.
- ❖ تسمية المتغير تسهل علينا التعامل مع البيانات المخزنة ضمن ذاكرة الحاسب ومعالجتها أثناء تنفيذ البرنامج .
- ❖ المتغيرات لها أنواع وتختلف حسب نوع البيانات التي نحتاجها مثل أرقام صحيحة ، أرقام عشرية ، نصوص ، حروف ، ... وغير ذلك.
- ❖ تتطلب لغة باسكال أن تكون كل المتغيرات معرفّة قبل استخدامها. وحتى في الوقت الذي تعرّف فيه المتغير، يجب أن تحدّد نوع البيانات.
- ❖ نستخدم المتغيرات ونقوم بتعريفها لعدة أسباب:

- ربط اسم المتغير بعنوان محدد في ذاكرة الحاسب.
- تحديد نوع البيانات التي سيتم تخزينها في هذا المتغير، وبالتالي تحديد حجم الذاكرة اللازمة لتخزين هذه البيانات لكي يتعامل الحاسب مع القيم بالصورة الصحيحة عند تحويلها للنظام الثنائي.

مفهوم أنواع البيانات The Concept of Data Types

- تصنّف المتغيرات وفقاً لبعض خواصها الهامة المشتركة وبعض الشروط القياسية التي تميّزها ويمكن تقسيمها إلى:
 ١. متغيرات تأخذ مجموعة محدودة من القيم يتم تعريفها من قبل المبرمج، ومجموعة جماعات هذه القيم
 ٢. متغيرات عددية (صحيحة، حقيقية، عقدية، بوليانية)
 ٣. دوال ومؤشرات على مجموعة دوال وغيرها
- يتم تعريف بنى المعطيات المحددة من خلال عبارات تعريف (تعلّمة Type مثلاً) التي تقوم بدورها بحجز أماكن في ذاكرة الحاسب وتكون غالباً على شكل صفوف لها بنية محددة على شكل سجلات ذات أنواع متماثلة ومرتبّة.

أنواع البيانات في لغة باسكال Pascal

- تعتبر البيانات العنصر الأساسي الذي يتعامل معه البرنامج والذي تقوم عليه المعالجة لذا سنقوم بتعريف أنواع البيانات وطريقة التصريح عنها في لغة باسكال . تمكننا هذه اللغة من تمييز نوعين من البيانات:
- الأول: وهو ما تشترك فيه كل لغات البرمجة لذلك سنطلق عليه اسم البيانات القياسية وهي :
 ١. الأعداد الصحيحة Integer وهي الأعداد الصحيحة الكاملة التي لا تحوي على كسور مثل 11 ، 456 ، 23333.. الخ
 ٢. أعداد الحقيقية Real وهي الأعداد الحقيقية التي تحوي كسور فيها مثل ٢.٣٤٣٤
 ٣. الرموز Char وهي الحروف والأشكال المميزة مثل A, @, H, ^, &
 ٤. المنطقية Boolean ويمكن أن تأخذ قيمتين (إما صحيح True أو خطأ False)
 ٥. الخيوط الرمزية String وهي عبارة عن خيط من الرموز Char م
- الثاني: وهو البيانات الخاصة التي يصرح عنها المبرمج وذلك بتعريف أنواع جديدة من البيانات ليست من الأنواع الخمسة التي تكلمنا عنها وسيتم تعريفها لاحقاً.

أنواع البيانات في لغة باسكال Data Types of Pascal

البيانات المركبة
structured

المجموعة
Set

المصفوفات
array

الدوال
function

السجلات
record

الملفات
file

المتحولات المطلقة
Absolute

المؤشرات
pointer

البيانات البدائية البسيطة
Primitive Data

البسيط
type

المجال
Subrang

البيانات الثابتة
Const

البيانات البدائية القياسية
Standard Primitive

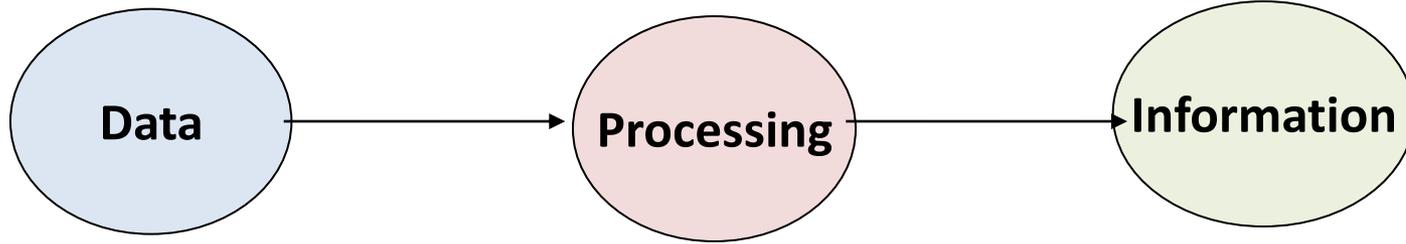
الصحيح
Integer

الحقيقي
Real

المحرفي
Char

البولياني
Boolean

كيف يتعامل الحاسب مع البيانات ؟



يتم تخزين البيانات في أماكن بالذاكرة تسمى (المتغيرات أو الثوابت)

- **الثوابت Constant** : البيانات التي يتعامل معها البرنامج، وهي "أماكن يتم حجزها في ذاكرة الحاسوب تقوم بتخزين قيم بيانات ثابتة لا تتغير أثناء تنفيذ البرنامج"، وهذه الثوابت تنقسم إلى عدة أقسام، منها:
 - أ- ثابت عددي .
 - ب- ثابت حرفي.

- **المتغيرات Variables** : البيانات التي يتعامل معها البرنامج، وهي "أماكن يتم حجزها في ذاكرة الحاسوب تقوم بتخزين قيم بيانات يمكن أن تتغير أثناء تنفيذ البرنامج"، وهذه المتغيرات تنقسم إلى عدة أقسام، منها:
 - أ- متغير عددي.
 - ب- متغير حرفي.
 - ج- متغير منطقي.

التصريح عن البيانات المتغيرة بلغة باسكال (تعليمة Var)

- يتم في هذا القسم من البرنامج التصريح عن نوع البيانات بالبرنامج قبل استخدامها ويتم التصريح حسب نوع البيانات Integer , Real , Boolean , Char أو نوع قام المستخدم بتعريفه ويخضع اسم المتغير إلى شروط وهي:

١. يجب أن يبدأ بحرف (ولا تقبل الأسماء التي تبدأ بأرقام)
٢. يجب أن يحوي الاسم حروفا وأرقاما فقط
٣. يجب ألا يكون الاسم من ضمن الكلمات المحجوزة بالذاكرة أي من كلمات لغة البرمجة.
٤. يفضل دائما أن يكون الاسم له معنى للتسهيل.

أمثلة:

- ✓ A: integer تم تعريف مكان بالذاكرة اسمه A وهو يقبل البيانات فقط من نوع الأعداد الصحيحة
- ✓ B: Real تم تعريف مكان بالذاكرة اسمه B وهو يقبل البيانات من النوع الصحيح وأيضا النوع الحقيقي الذي يحوي كسور
- ✓ Grade: Char تم تعريف مكان بالذاكرة اسمه Grade وهو يقبل البيانات من النوع Char
- ✓ F: Boolean تم تعريف مكان بالذاكرة اسمه F يقبل بيانات من النوع Boolean أي يقبل قيم منطقية (True / False)

تصريح البيانات المتغيرة Var

١- البيانات البدائية القياسية Standard Primitive Type

١- ١- النوع الصحيح Integer Type: يمكن تعريف المتحول الصحيح وفقاً للقيمة التي ستسند إليه، ويجب تحديد نوع المتحول بشكل دقيق بما يتوافق مع القيم التي سيتم تخزينها فيه، وإلا فإن الحاسب سيعطي خطأ أثناء تنفيذ البرنامج، ونميّز الأنواع الصحيحة التالية:

- shortint [-128 ... 127] (1byte) بما فيها إشارة العدد
 - byte [0 ... 255] (1byte) بدون إشارة
 - integer [-32768 ... 32767] (2byte) بما فيها إشارة العدد
 - word [0 ... 65535] (2byte) بدون إشارة
 - longint [- 2147483648 ... 2147483643] (4byte) بما فيها إشارة العدد
- تعرّف المتحوّلات الصحيحة على الشكل التالي ووفقاً للقيم التي ستحفظ فيها:
Var a: shortint; b: byte; c: integer; d: word; e: longint;
 - يتم تنفيذ العمليات الحسابية القياسية التالية على البيانات الصحيحة:
 - ✓ الجمع (+) والطرح (-) والضرب (*)
 - ✓ القسمة الصحيحة (div): تعطي ناتج القسمة الصحيح بدون باقٍ $c := a \text{ div } b$
 - ✓ باقي القسمة (mod): تعطي باقي القسمة الصحيحة $h := a \text{ mod } b$

تصريح البيانات المتغيرة Var

١ - البيانات البدائية القياسية Standard Primitive Type

٢ - ١ - النوع الحقيقي **Real Type**: يمكن تعريف المتحول الحقيقي وفقاً للقيمة التي ستسند إليه، ويجب تحديد نوع المتحول بشكل دقيق بما يتوافق مع القيم التي سيتم تخزينها فيه، وإلا فإن الحاسب سيعطي خطأ أثناء تنفيذ البرنامج، ونميز الأنواع الحقيقية التالية:

-single [$1.5 \times 10^{-45} \dots 3.4 \times 10^{+38}$] (4byte)

- real [$2.9 \times 10^{-39} \dots 1.7 \times 10^{+38}$] (6byte)

- double [$5.0 \times 10^{-324} \dots 1.7 \times 10^{+308}$] (8byte)

- comp [$-2^{63} + 1 \dots 2^{63} - 1$] (8byte) يتعامل مع قيم صحيحة فقط ضمن مجال قيمته

- extended [$3.4 \times 10^{-4932} \dots 1.1 \times 10^{4932}$] (10byte)

• تعرّف المتحولات الحقيقية على الشكل التالي ووفقاً للقيم التي ستحفظ فيها:

Var a: single; b: real; c: double; d: comp; e: extended;

• يتم تنفيذ العمليات الحسابية القياسية التالية على البيانات الصحيحة:

✓ الجمع (+) والطرح (-) والضرب (*)

✓ القسمة الحقيقية (/): ناتج القسمة عدد حقيقي يتم تقريبه وتدويره $c := a / b$

تصريح البيانات المتغيرة Var

١- البيانات البدائية القياسية Standard Primitive Type

٣- ١- النوع البوليني Boolean Type:

- يحوي هذا النوع قيمتين فقط (true, false) ويمكن تنفيذ العمليات التالية على قيم هذه المتحولات المنطقية (or, and, not). نتيجة أي من العمليات المنطقية (=, ≠, >, <, ≤, ≥) تعطي قيمة بوليانية.

٤- ١- النوع المحرفي Char Type:

- يحوي هذا النوع كافة الرموز والأرقام وبعض رموز التحكم الموجودة على لوحة مفاتيح جهاز الحاسب، ويتم تنفيذ مجموعة من العمليات الحسابية والمنطقية على هذا النوع من البيانات.
- كل رمز موجود على لوحة المفاتيح يقابله عدد "وزن" في جدول الترميز ASCII الموضوع من قبل المنظمة العالمية للقياس ISO. علماً أنه يوجد على لوحة المفاتيح ٩٥ رمز بالإضافة إلى ٣٣ رمز تحكم.
- الأرقام من ٠ - ٣١ محجوزة لمفاتيح التحكم في وسائط الإدخال والإخراج والوحدات الملحقة بالحاسبات
- الرمز " " فراغ يقابله الرقم ٣٢
- الأحرف والأرقام وباقي الرموز من ٣٣ - ٢٥٥
- يمكن احتساب الرقم المقابل لأي رمز بواسطة التعليمة ord وذلك على الشكل التالي:

- ord ('a') := 97
- ord ('o') := 48
- ord ('A') := 65

أنواع البيانات Data Types في لغة باسكال

٢- تصريح البيانات الثابتة Const

- ويتم في هذا القسم من البرنامج التصريح عن البيانات الثابتة قبل استخدامها ويخضع اسم الثابت إلى شروط وهي:

١- أن يبدأ بحرف (ولا تقبل الأسماء التي تبدأ بأرقام)

٢- أن يحوي الاسم حروفا وأرقاما فقط

٣- ألا يكون الاسم من ضمن الكلمات المحجوزة بالذاكرة لأي من كلمات اللغة .

٤- يفضل دائما أن يكون اسم له معنى للتسهيل

٥- إسناد قيمة الثابت إلى اسم الثابت

- مثال:

Const

```
P := 3.14159;
```

```
Week := 7; Month:= 30;
```

- تم تعريف مكانين بالذاكرة اسماهما P & week ويقبلان الأعداد المسندة إليهما فقط ، ولا يقبلان أن تغيير لقيمهما أثناء تنفيذ البرنامج.

أنواع البيانات Data Types في لغة باسكال

٣-١- البيانات البدائية البسيطة Primitive Data Types

يمكن تمثيل نوع مخصص من البيانات على شكل متحول له قيم محددة يتم التعريف عنها في بداية البرنامج بواسطة تعليمة `type` ، ثم يعرف هذا المتحول على أنه من النوع المعلن عنه كي يتم حجزه في ذاكرة الحاسب.

• أمثلة: - أيام الأسبوع يمكن تمثيلها ضمن متحول من النوع المحدد الذي نعرّفه على الشكل التالي:

- **Program** week (input, output);
- **Type** weekday := (SAT,SUN,MON,TUE,WED,THU,FRI);
- **Var** day: weekday;

-جنس الطالب:

- **Program** Students (input, output);
- **Type** sex := (male, female);
- **Var** stud: sex;

- أنواع العملات:

- **Program** Money (input, output);
- **Type** currency := (frank, euro, dollar, pound, lira, yen);
- **Var** unit: currency;

أنواع البيانات Data Types في لغة باسكال

٢-٣- الأنواع المجالية أو المحدودة Subrang Types

المتحولات التي تأخذ قيمة في مجال محدد، يجب تعريفها وفق قواعد لغة باسكال على الشكل التالي:

- ✓ تعريف نوع محدد من البيانات بواسطة تعليمة Type
 - ✓ التعريف بحدّي المجال المعرّف (الأول والأخير) Type t:= min .. max;
 - ✓ التأكد من أن كافة قيم المجال من نوع واحد (صحيح أو حقيقي أو محرفي أو بولياني)
 - ✓ التصريح عن أسماء المتحولات التي سيتم استخدامها في تنفيذ البرنامج بواسطة تعليمة Var
- أمثلة:

Type year := 1900 .. 1990;

Type letter := 'a' .. 'z';

Type number := '0' .. '9';

Type day := 1 .. 31;

Var y: year; l: letter; n: number; d: day;

أنواع البيانات Data Types في لغة باسكال

٤- المتحولات المطلقة (المؤشرات) Absolute Variables

- تمكنا لغة باسكال من تحديد موقع معين في ذاكرة الحاسب لتخزين المتحول الذي نحجزه بواسطة `var`، عندئذ نسمّيه متحولاً مطلقاً (مؤشراً)، ونستخدم الصيغة `Segment` لتحديد المقطع و `offset` لانزياح مكان المتحول.

```
Var m : byte absolute $0040 : $0049 ;
```

```
Var str : string [ 32 ] ;
```

```
Ln : byte absolute str ;
```