الدرس السابع

درسنا الان يتصف بالصفات التالية فهو سهل الفهم ممتع بالاضافة لانه بسيط ايضا جد مختصر

العبارة

تنويه27 يجب على المبرمج الحديث العهد نعلم الفرق بين التعبير والعبارة التعريف الرسمي للعبارة هو التعبير المتبوع بفاصلة منقوطة اما التعبير فهو وحدة من الشيفرة تعطي مقدار ما .....

انا لم افهم التعريف هههههه

مثال **c := a + b ;**

في مثالنا السابق الجزء الموجود الى يمين معامل الاسناد **a + b** هو تعبير بينما السطر باكمله هو عبارة يمكنكم القول ان التعبير هو جزء من عبارة والعبارة يمكن تشكلها من عدة تعابير

ان جميع العمليات الحسابية الموجودة في الجدول الاتي هي عمليات ثنائية وقد اوردت مثالا على ذالك في الدرس السابق

تنويه28 تعطي عملية التقسيم للاعداد الصحيحة كنتيجة عددا صحيحا مثال

```
z:=11;
y:=3;
x:=z div y ;
write(x);
```

ان نتيجة هذه القسمة عددا ليس صحيحا ولكنها مع عملية القسمة هذه يعطي قيمة صحيحة فقط انظر

```
3
```

لاحظ انه تم استبعاد اي جزء عشري اثناء عملية حساب رقمين صحيحين على شكل عدد صحيح وهي لاتستعمل الا مع الاعداد الصحيحة

تعطي عبارة **(z mod y)** باقي قسمة **z**على **y** فنتيجة عبارة **11 mod 3** هي 2 انظر

```
z:=11;
y:=3;
x:=z mod y ;
write(x);
```

```
2
```

تنويه 29 يعتبر استخدام عملية باقس القسمة الصحيحة **mod**مع الاعداد الغير صحيحة خطأ قواعديا

تنويه 30 يجب كتابة التعابير الحسابية في لغة الدلفي على خط مستقيم بحيث تظهر كافة الثوابت والمتحولات في نفس السطر اما الشكل الجبري فهو غير مقبول من قبل المترجم

تنويه31 ان المثالان السابقان تم بعد تعريف متغير من نوع **Integer**

اما اذا كنا نريد ان تظهر لنا النتائج كاملة ويقبل بها اي عدد كان اشارة سلبية او بفاصلة عائمة يجب ان نستخدم متحول من نوع **Double** مثال

```
{$APPTYPE CONSOLE}

uses
  SysUtils;
 var
  z : Double ;
  x : Double ;
  y : Double ;
begin
  { TODO -oUser -c
  z := 11 ;
  y := 3 ;
  x := z/y ;
  write(x);
end.
```

انظر الخرج قيمة حقيقية

```
3.66666666666667E+0000
```

يمكن استخدام الاقواس بتفس الطريقة التي نستعملها في العبارات الجبرية
مثال اذا اردنا ضرب **a**بالكمية **b+c** يجب ان نكتب
a*(b+c)

عاملات أحادية (أسبقية عليا)	
@	عنوان المتغير أو الوظيفة (ترجّع مؤشّر)
not	بولي أو ما يخص الجزيئات
العاملات الضربية و ما يخص الجزئيات	
*	ضرب حسابي أو تقاطع فئة
/	تقسيم نقطة عائمة
div	تقسيم عدد صحيح
mod	البواقي (باقي تقسيم عدد صحيح)
as	تلبيس نوع آمن (RTTI)
and	بولي أو ما يخص الجزيئات
shl	ازاحة لليسار فيما يخصّ الجزئيات
shr	ازاحة لليمين فيما يخصّ الجزئيات
العاملات الجمعية	
+	جمع حسابي، اتحاد فئة، ربط جمل، اضافة رصيف مؤشّر
-	طرح حسابي، طرح وتخالف فئة، طرح صف مؤشّر
or	بولي أو ما يخص الجزيئات

xor	بولي أو ما يخص الجزيئات
عاملات العلاقة والمقارنة (أسبقية دنيا)	
=	اختبار مساواة
<>	اختبار عدم مساواة
>	اختبار أقل من
<	اختبار اكبر من
=<	اختبار أقل من أو يساوي، أو فئة فرعية من فئة
>=	اختبار أكبر من أو يساوي، أو فئة عليا تعلو فئة
in	اختبار اذا ما عنصر عضو في فئة
is	اختبار توافقية نوع لكائن(عامل RTTI آخر)

تتحكم العبارات **statem** في تسلسل تنفيذ البرنامج او تقوم بتقييم التعبير او لاتفعل اي شيء
تنويه32 يجب ان تنتهي كل عبارة بفاصلة منقوطة ... طبعا كما ذكرت سابقا الا في حالات قليلة وساتي على ذكرها كلما مررت عليها
الان ناتي على ذكر الاقواس ان الاقواس في الباسكال كما هي الحال في السي++ تعطي اهمية القصوى حيث يتم تنفيذ التعبير الذي يكون بين الاقواس اولا في العبارة زمن ثم يتم تنفيذ باقي التعليمات وفي الخالات التي مرت معنا في درسنا السابق فانه في العبارة الواحدة لم ياتي بعد قيمة الاسناد سزى تعبير واحد لذالك كتبتها لتحسين المقرؤية فقط
لنتسائل عن اولوية تنفيذ تعليمات العمليات الحسابيةاو المعاملات في العبارة الواحدة
تقوم الباسكال كما في السي ++ بتطبيق العمليات الحسابية وفق ترتيب معين تبعا لقواعد الاولوية

كما قلنا ان البرنامج في الباسكال كما في اغلب اللغات ينفذ بالتسلسل حسب ورود التعليمة في نص البرنامج من الاعلى للاسفل كما عرفنا في درسنا السابق
يجري اولا انجاز الحسابات بالنسبة للازواج الموجودة بين قوسين لذالك نستخدم الاقواس لتحديد الحسابات التي نرغب كمبرمجين لذالك نقول ان الاقواس انها تحدد اعلى مستى للاولوية وفي حال وجود مجموعات للاقواس متداخلة بعضها ببعض فإن الحساب يبدأ انطلاقا من الزوج الداخلي الاول الذي لايحوي اي مجموعة للاقواس ضمنه
مثال

z:= (a*(b+c))/x

فانه سوف يتم جمع **b**مع **c** ومن ثم يتم ضرب كميتها مع **a** ثم تقسم الكمية الناتجة على **x**
بعد الاقواس يتم حساب عمليات الضرب والقسمة وباقي القسمة الصحيحة اذا كانت العبارة تحتوي على عدة عمليات ضرب او تقسيم او حساب لباقي قسمة صحيحة فيجري حسابها كما هو الحال في السي++ من اليسار الى اليمين نقول عن العمليات الضرب والقسمة وباقي القسمة الصحيحة انها من نفس مستوى الاولوية مثال
ففي مثالنا السابق لو كتبناه بدون اقواس **z:=a*b+c/x** فان التنفيذ سوف يبدأ بضرب **a**مع **b**ومن ثم سوف يقوم بقسمة **c**على **x**وبعدها يقوم بجمع الكميتين الناتجتين
اخيرا وبحسب الاولوية نقوم بحساب عمليات الجمع والطرح واذا حوت العبارة على عدة عمليات جمع او طرح عندها تبدأ عملية الحساب من اليسار الى اليمين ايضا نقول ان عمليتي الجمع والطرح لهما نفس المستوى
ومن ثم تاتي عملية الاسناد =
تعالو لنطبق قواعد الاولوية السابقة على مجموعة من الامثلة والتعابير الحسابية
اذا اردنا حساب المتوسط الحسابي لخمسة حدود
جبريا

$$m = \frac{a+b+c+d+e}{5}$$

حسب الباسكال

m = (a+b+c+d+e)/5

تنويه33 هنا من الضروري وضع داخل اقواس طريقة التعبير لان عملية القسمة لها اولوية التنفيذ على عملية الجمع وبالتالي وجود الاقواس يعني حساب ناتج قسمة الكمية **(a+b+c+d+e)**على الكمية 5 اما غيابها فيعني انه سيتم قسمة **e**على 5 ومن ثم جمع هذه الكمية مع الكميات الاخرى وهذا لايتفق مع المطلوب حسابه

لناخذ مثال معادلة خط مستقيم جبريا **m=yx+b**

حسب الباسكال **m=y*x+b**

لاحظ هنا لاحاجة لاستخدام الاقواس حيث يتم حساب اولا عملية الضرب لان لها الاولوية

الى هنا ينتهي الدرس السابع وتنتهي معه الماضرة الاولى

المحاضرة الثانية سوف نبدا ببنى التحكم وسيكون الدرس الاول لدراسة بنية التحكم **if**