

جامعة حماة

كلية العلوم في مصيف / السنة الأولى

المادة: لغات البرمجة

المحاضرة الثالثة: الأوامر الأساسية و الشرطية و الحلقات التكرارية في لغة باسكال

Conditional Statements and Loops in Pascal



بنية برنامج باسكال

Program name (input, output);

Var



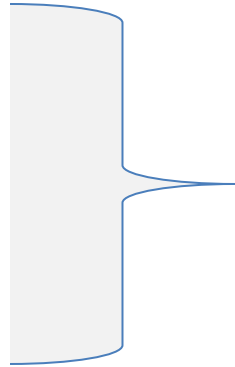
...

Begin



....

{Your program is here}



....

End.



١. اسم البرنامج

٢. الإعلان عن متغيرات البرنامج

٣. بداية البرنامج

٤. العبارات البرمجية

٥. نهاية البرنامج

قسم رأس البرنامج Program

- السطر الأول لأي برنامج مكتوب بلغة باسكال مخصص لكتابه اسم البرنامج و يجب أن يكون للاسم علاقة بمضمون البرنامج، فمثلا يمكن تسمية برنامج الآلة الحاسبة `calculate`:

Program calculate (input, output);

- هناك قواعد لكتابه أسماء البرامج فلا يجوز أن يبدأ الاسم برقم أو أن يحتوي على مسافات، و في حالة اسم البرنامج مؤلف من كلمتين فمن الممكن أن تكتب الكلمتان متلاصقتان مع مراعاة تكبير الحرف الأول من كل كلمه أو وضع علامة الشرطة – للفصل بين الاسم الأول و الثاني.
- الكلمتين `input` و `output` لتوضيح أن البرنامج يحتوي على مدخلات (أي يقوم مستخدم البرنامج بإدخال بيانات) أو مخرجات (أي طباعة النتائج على الشاشة فور تنفيذ البرنامج) .

• يمكن الاكتفاء باسم البرنامج كالتالي **Program calculate (i,o);**

• أو الاكتفاء باسم البرنامج فقط **Program calculate;**

قسم الإعلان عن المتغيرات Var

- الكلمة المحجوزة **Var** (عند كتابه الكلمات المحجوزة في المترجم ، فإنه يقوم بكتابتها بالخط العريض **Bold**) و تشير إلى أسماء المتغيرات التي سيستخدمها البرنامج.
- المتغيرات: هي مساحة في ذاكرة الحاسب يستخدمها برنامج الباسكال لتخزين بعض المعلومات، و من الممكن أن تتغير المعلومات على حسب الأوامر التي تعالج في لحظه تنفيذ البرنامج .
- قواعد تسميه المتغيرات: يجب ألا يبدأ برقم أو أن يحتوي على مسافات، وألا يكون من الأسماء المحجوزة في لغة باسكال..الخ.

Var

```
Num1: integer; // Num1, Num2, Num3 : integer;
```

```
Result: real; // R1, R2, R3 : real;
```

```
Name: char; // Name, Family, Address : char;
```

أوامر إدخال القيم للمتغيرات المعرّفة (١ - الإسناد :=)

- الإسناد: نقوم بكتابة اسم المتغير يليه نقطتان وعلامة مساواة تليهما القيمة المراد وضعها في المتغير ثم ننهي السطر بفاصلة منقوطة (دليل على نهاية التعليمة البرمجية).
- أمثلة على ذلك :

number := 1404;

number2 := -2.07;

text := 'Hello Mr.Reader';

character := 'Y';

bool := false;

ملاحظة هامة: يجب التمييز بين إشارتي المساواة (=) والإسناد (:=)، لأن إشارة المساواة تستخدم للمقارنة بين قيمتين أو بين متغيرين، بينما الإسناد تمكننا من وضع القيمة الموجودة على اليمين في المتغير الموجود على يسار إشارة الإسناد

أوامر إدخال القيم للمتغيرات المعرّفة (٢ - إدخال Read)

١. تستخدم تعليمة `read()` لإدخال قيم للمتغيرات التي تم تعريفها في بداية البرنامج والبقاء على نفس السطر.

٢. تستخدم تعليمة `readln ()` لإدخال قيم للمتغيرات التي تم تعريفها في بداية البرنامج والانتقال إلى سطر جديد.

• أمثلة:

- `read (Num);` لإدخال القيم إلى المتغيرات والبقاء على نفس السطر
- `readln(Num);` لإدخال القيم إلى المتغيرات ثم الانتقال إلى سطر جديد
- `Readln();` لإيقاف شاشة الحاسب عند خطوة معينة

أوامر إظهار قيم المتغيرات (طباعة) Write

1. **تعلية write:** تستخدم لإظهار وطباعة نص معيّن على شاشة الحاسب أو إظهار قيم المتغيرات (محتواها) التي تم تعريفها في بداية البرنامج والبقاء على نفس السطر.
 - كما يمكن تنفيذ عمليات حسابية ضمن تعليمة write و طباعة ناتج تنفيذها.
 - أمثلة:

Write (Sum); لطباعة محتوى المتغير

Write ('Sum='); لطباعة النص الموجود ضمن علامتي التنصيص

Write ('Sum=', Sum); لطباعة اسم المتغير ومحتواه

Write ('Sum=', Sum+5); لطباعة اسم المتغير ومحتواه بعد جمعه مع العدد 5

أوامر إظهار قيم المتغيرات (طباعة ٢) WriteIn

٢- **تعلية writeIn:** تستخدم لإظهار وطباعة نص معيّن على شاشة الحاسب أو إظهار قيم المتغيرات (محتواها) التي تم تعريفها في بداية البرنامج والانتقال إلى سطر جديد.

• مثال:

WriteIn (Sum) لطباعة محتوى المتغير و الانتقال إلى سطر جديد

WriteIn('Sum='); لطباعة النص الموجود ضمن علامتي التنصيص ثم الانتقال إلى سطر جديد

WriteIn('Sum=', Sum); لطباعة اسم المتغير ومحتواه و الانتقال إلى سطر جديد

أمثلة على تعليمة write

Program Name1 (i ,o);

Var

X: char;

Begin

write(' Input your name : ');

Read(x);

Writeln (x);

End.

Input your name : Haysam الناتج:

Haysam

Program name2 (i,o);

Begin

write(' My name is : ');

writeln(' Haysam');

write(' My First Program');

End.

My name is : Haysam الناتج:

My First Program

مثال على تعليمتي read و write

```
program lesson2;  
  
var  
name : string;  
  
begin  
  
writeln('Please write your name : ');  
readln(name);  
write('Hello Mr. ');  
write(name);  
  
end.
```

اكتب برنامج يطلب منك إدخال اسمك
و يقوم بطباعة عبارة Hello Mr
قبل اسمك على نفس السطر

النتائج:

```
Please write your name:  
AHMAD  
Hello Mr. AHMAD
```

التعليمات الشرطية Conditional Statement

- التعليمات الشرطية Conditional Statement تستعمل لتنفيذ إحدى التعليمات التي تتضمنها أو عدم تنفيذ ولا واحدة منها. بالاعتماد على شيء من الاختبار (مقارنة منطقية).
- يوجد شكلين من التعليمات الشرطية:

١. تعليمات if :

– تعليمات if الشرطية

– تعليمات if – then – else

– تعليمات if الشرطية المتداخلة

٢. تعليمات case :

تعلیمة If - then الشرطیة البسیطة

- if ... then
statement;
- if ... then
begin
statement1;
statement2;
.....
end;

١. تعلیمة If – then یمكن استخدامها لتنفیذ تعلیمة برمجیة واحدة فقط إذا تحقق شرط معین (if-then) و فی حال عدم تحقق الشرط یتم الانتقال إلى السطر البرمجي التالي.

٢. یمكن استخدام تعلیمة If - then لتنفیذ عدة تعلیمات إذا تحقق شرط معین، و ذلك عن طریق حصر التعلیمات ضمن (begin...end;) و فی حال عدم تحقق الشرط یتم الانتقال إلى السطر البرمجي الموجود بعد .end;

مثال على تعليمة If - then الشرطية البسيطة

Program clean (i , o);

Var

X: integer;

Begin

write(' Input Number x = ');

Readln(x);

If ($x > 10$) then writeln (' bigger then 10 ');

Readln ();

End.

- برنامج لإدخال عدد ما و طباعة جملة العدد أكبر من عشرة إذا تحقق الشرط .

تعليمة If – then - else الشرطية

- if ... then
 statement1;
else
 statement2;
- if ... then
 begin
 statement1;
 statement2;
 end;
else
 statement3;

- للاختيار بين بدليين (if-then-else) أي يتم التحكم بمسرى تنفيذ العمليات في كلتا حالتى تحقق الشرط أو عدم تحققه (في حال تحققه يتم تنفيذ تعليمة واحدة أو عدة تعليمات و في حال عدم تحقق الشرط يتم تنفيذ تعليمة واحدة أو مجموعة تعليمات أخرى)، و يتم وصف الشرط بتعبير بولياني Boolean مثال باسكال بسيط سيوضح كيفية كتابة تعليمات شرطية.

مثال على تعليمة if – then – else الشرطية

Program CLEAN1 (i , o);

Var

X: integer;

Begin

write(' Input Number x = ');

Readln(x);

If ($x > 10$) then writeln (' bigger than 10 ')

else Writeln (' less than 10 ');

Readln();

End.

- برنامج لإدخال عدد ما و طباعة عبارة العدد أكبر من عشرة إذا تحقق الشرط أو طباعة العدد أصغر من عشرة إذا لم يتحقق الشرط

تعليمات If الشرطية المتداخلة

- if ... then
begin
if ... then
statement1;
statement2;
end;

- يمكن استخدام تعليمة if لتنفيذ تعليمة واحدة أو عدة تعليمات محصورة بين begin و end; إذا تحقق أكثر من شرط (if-then .. if-then) ضمن (begin-end;)

تعليمة Case الشرطية

- تستخدم عبارة (الحالة) Case في المواقع التي يراد فيها الاختيار بين عدد من البدائل.
- تستخدم عبارة CASE كبديل عن استعمال عدد من عبارات IF - ELSE المتتالية، والتي غالباً ما تكون أكثر تعقيداً، وصعبة التتبع والفهم، أما باستعمال CASE فإن البرنامج يصبح أكثر وضوحاً، وسهل التتبع. تكتب عبارة case كالتالي:

```
CASE selector OF
  constant1 :
    statement1;
  constant2 :
    statment2;
.
.
.
END;
```

- selector هو عبارة عن متغير من أي نوع معرّف مسبقاً في بداية البرنامج.

- constant1 و constant2 هي قيم يمكن أن يأخذها المتغير selector و من نفس النوع.

- إذا كانت قيمة selector تساوي constant1 يتم تنفيذ العبارة statement1.

- إذا كانت قيمته تساوي constant2 فتنفذ العبارة statment2 وهكذا دواليه حتى نهاية هيكل العبارة CASE عند تعليمة END;

مثال ١ على تعليمة Case الشرطية

```
Program Test3 (i , o);  
Var  
  Num : integer;  
Begin  
  write('Input Number = ');  
  readln(Num);  
  case Num of  
    1: writeln ( 'One');  
    2: writeln ('Two');  
    3: writeln ('Three');  
  end;  
End.
```

- عندما تصبح تعليمة if أكثر تعقيدا (متداخلة)، يمكنك استبدالها بتعليمة case
- تعليمة case عبارة عن تعبير يستخدم لاختيار تنفيذ تعليمة واحدة أو أكثر من ضمن قائمة شرطية بعدة قيم محتملة (عدة شروط)، أو مدى من القيم.
- مثال: أدخل رقم Num و اطبع اسم العدد حسب قيمته.

مثال ٢ على تعليمة Case الشرطية

- مثال: اكتب برنامج يقوم بإدخال عدد صحيح Number (من ١ إلى ١٠) و أطلع عبارة odd إذا كان العدد فردي أو عبارة even إذا كان زوجي.

Program ex2 (input, output);

```
VAR          number : Integer;
```

```
BEGIN
```

```
  writeln('Enter a number between 1 - 10: ');
```

```
  readln(number);
```

```
  CASE number OF
```

```
    1, 3, 5, 7, 9 : writeln('The number is odd');
```

```
    2, 4, 6, 8, 10: writeln(' the number is even');
```

```
  END;
```

```
END.
```

مثال ٣ على تعليمة Case الشرطية

Program TEST4 (i , o);

Var

MyChar : char;

Begin

write('Input Symbol= '); readln(MyChar);

case MyChar of

'+' : writeln ('Plus sign');

'-' : writeln ('Minus sign');

*, '/': writeln ('Multiplication or division');

'0'..'9': writeln ('Number');

'a'..'z': writeln ('Lowercase character');

'A'..'Z': writeln ('Uppercase character');

else

writeln ('Unknown character');

end;

End.

- القيم هي ثوابت Constants ، ويجب أن تكون فريدة ومن نوع ترانبي Ordinal أحيانا، قد يوجد بها تعليمة else و التي يتم تنفيذها إذا لم تتوافق أي من الاحتمالات مع القيمة المعطاة.
- مثال: أطبع اسم القيمة التي يتم إدخالها عن طريق لوحة المفاتيح إلى المتغير Mychar و إلا أطبع أن الرمز غير معروف Unknown character

الحلقات التكرارية في باسكال

- تستعمل الحلقات التكرارية Loops لتنفيذ تعليمة أو مجموعة من التعليمات عدة مرات وذلك بالاعتماد على شيء من الاختبار .
- تحتوي لغة باسكال التعليمات التكرارية النمطية التي تمتلكها معظم لغات البرمجة.
- يوجد ثلاثة أنواع أساسية من الحلقات التكرارية:

١ . حلقة For

٢ . حلقة While..do

٣ . حلقة Repeat ...until

حلقة For التكرارية

- تعتبر حلقة for في لغة باسكال أقل مرونة مقارنة بلغات البرمجة الأخرى (ليس بالإمكان تحديد معدّل الزيادة أو النقصان إلا بواحد)، لكنها بسيطة و سهلة الفهم.
- إذا أردنا اختبار شرط بتشعب أكبر، أو أردنا إيجاد عدّاد بمواصفات خاصة، فنحن بحاجة إلى استخدام تعليمات while أو repeat ، عوضاً عن حلقة for.
- ملاحظة: عدّاد حلقة for ليس بالضرورة أن يكون رقماً. يمكن له أن يكون قيمة من نوع تراتبي، مثل حرف أو نوع سردي.
- تكتب حلقة for على الشكل التالي كمثل لطباعة الرمز @ ستة مرات متتابة

```
for i:=1 to 6 do
```

```
write ('@'); → @ @ @ @ @ @ ← سيكون ناتج التنفيذ
```

مثال ١ على حلقة For

```
Program ADDS1 (i , o);
```

```
Var
```

```
Sum, i: Integer;
```

```
begin
```

```
Sum:= 0;
```

```
for i := 1 to 10 do
```

```
Sum := Sum + i;
```

```
writeln (Sum);
```

```
End.
```

- حلقة for في لغة باسكال مبنية بصورة مقيدة على عدّاد Counter ، يمكن زيادته أو تخفيضه بقيمة واحد فقط في كل مرة يتم تنفيذ الحلقة ($i := i + 1$) أو ($i := i - 1$) فقط.
- فيما يلي مثال بسيط لحلقة for مستخدمة لجمع و طباعة مجموع الأعداد من واحد إلى عشرة.

مثال ٢ على حلقة For

```
Program ADDS2 (i , o);  
var  
    Sum, j : Integer;  
begin  
    Sum := 0;  
    for j := 10 down to 1 do  
        Sum := Sum + j ;  
        writeln ('Sum = ',Sum);  
End.
```

- نفس المثال السابق يمكن كتابته بطريقة ثانية:
- فيما يلي مثال لحلقة for مستخدمة لجمع و طباعة مجموع الأعداد من عشرة إلى واحد باستخدام عدّاد معكوس:

أمثلة ٣ و ٤ على حلقة For

- برنامج لطباعة جداول ضرب الأعداد من ١ إلى ١٠ (عشرة جداول)

Program multiplic (i, o);

var

i , j : integer;

begin

for i :=1 to 10 do

for j :=1 to 10 do

writeln(i , ' * ', j , '=' , i * j);

end.

- برنامج لحساب مجموع قواسم عدد باستخدام لغة الباسكال

Program Div (i , o);

var

x,i,k: integer;

begin

write('x='); readln(x);

k:=0;

for i:=1 to x do

if x mod i = 0 then

k:=k+i;

writeln('k=',k);

end.

تعليمات Repeat و While

- الفرق الرئيسي بين هاتين الحلقتين هو أن حلقة repeat-until لديها شرط محجوز و سيتم تنفيذ التعليمات الموجودة بين repeat و until طالما الشرط غير محقق وتتوقف الحلقة عندما يتحقق الشرط.
- على العكس من ذلك فإن حلقة while-do تنفذ ما دام الشرط محققاً.
- لهذا السبب علينا أن نعكس الشرط في حال استخدام التعليمة أعلاه للحصول على تعليمة مشابهة.

While ... do تعليمة

- while (I < 100) and (J < 100) do
begin
// use I and J to compute something...
statement1;
statement2;
.....
I := I + 1;
J := J + 1;
end;

- الفرق بين حلقة while-do و حلقة repeat-until هو أن التعليمات الموجودة بعد تعليمة repeat تنفذ دائماً، مرّة واحدة على الأقل. نستطيع بسهولة فهم السبب بالإطّلاع على المثال التالي:

Program FACTORIAL (i , o);

var

N, F, i : integer ;

begin

write('N='); readln(N);

i := 1; F := 1;

while (i <= N) do

begin

F := F* i ; i := i + 1;

End;

writeln(N, ' ! =',F);

end.

مثال على تعليمة While ... do

- تمكنا تعليمة While من تكرار تنفيذ مجموعة من التعليمات المحصورة بين do ... end; الواقعة ضمن الشرط المذكور.
- اكتب برنامجا لحساب قيمة عاملي عدد N! باستخدام حلقة While

تعليمة Repeat ... Until

- repeat

// use I and J to compute something...

statement1;

statement2;

.....

I := I + 1;

J := J + 1;

until (I > 100) or (J > 100);

- تمكنا تعليمة Repeat من تكرار تنفيذ مجموعة من التعليمات الموجودة قبل الشرط ... until
- إذا كانت القيمة الابتدائية في I أو J أكبر من 100، فإن التعليمات داخل حلقة repeat-until سيتم تنفيذها مرة على كل حال.

Repeat ... Until مثال على تعليمة

Program factorialN (i , o);

var

M, F, i : integer ;

begin

write('M =');

readln(M);

i := 1; F := 1;

Repeat

F := F* i ;

i := i + 1;

until (i > M);

writeln(M, ' ! =',F);

end.

• اكتب برنامجا لحساب قيمة عاملي

عدد M! باستخدام حلقة Repeat