

جامعة حماة

كلية العلوم في مصيف / السنة الأولى

المادة: لغات البرمجة - العام الدراسي ٢٠١٩ - ٢٠٢٠
المحاضرة السادسة: المجموعات و السجلات في لغة باسكال

Set's and Records in Pascal



السلسلة المحرفية String

- السلسلة المحرفية هي مصفوفة أحادية مضمونها رموز char و دليلها Integer يبدأ من الواحد .
- لا تتجاوز السلسلة المحرفية ٢٥٥ حرفاً، إذا لم نحدد الطول الأعظمي للسلسلة عند تعريفها فإنه يكون ٢٥٥.
- عند تعريف سلسلة يحجز لها خانات بطولها + خانتين إضافيتين هما الخانة رقم ٠ و تحوي طول السلسلة و الخانة التي تزيد عن طولها ب ١ و تضم رمز لاغي مهمته الدلالة على طول السلسلة و يتم تعريفها إما بتحديد طولها :

S:string [20];

- أو بدون تحديد الطول عندها يكون ٢٥٥

S:string ;

- كما أن السلسلة تمتاز بأشياء إضافية فمثلاً لكتابتها أو قراءتها يمكن معاملة string كما يلي :
يمكن الاكتفاء بكتابة أسمها دون الحاجة لتعليمة for

WriteLn(s);

ReadLn(s);

السلسلة المحرفية String

- من التوابع مسبقا التعريف التي تتعامل مع السلاسل المحرفية String.
- التابع **Length**: يأخذ كوسيط سلسلة محرفية و يعطي طولها (عدد محارفها)
- مثال :

```
S:= 'Hello world'
```

```
WriteLn (Length(s));
```

الخرج : 11

- التابع **Concat**: يقوم بضم مجموعة من السلاسل الممررة إليه كوسطاء في سلسلة واحدة
- مثال :

```
S1:='Hello ';
```

```
S2:='world' ;
```

```
S3:=concat(s1,s2);
```

```
WriteLn (s3);
```

الخرج : Hello world

الجميل المعرفة من نوع Type

- هي عبارة عن قيم يتم تخزينها تحت اسم (متغير) .
- فمثلا الأعداد الصحيحة integer: هي الأعداد من ٣٢٧٦٨- إلى ٣٢٧٦٧ + يتم تخزينها تحت اسم متغير ما، وبإمكانك أن تعرف مجموعه وتضعها تحت اسم (متغير) .
- يمكن استخدام هذا المتغير في برنامجك بشكل طبيعي، باستثناء أنك لا تستطيع استخدام جمل الطباعة write و writeln أيضا لا تستطيع استخدام جمل القراءة read و readln.
- مثال: نستطيع تعريف مجموعه من الأسماء تحت متغير name

Type

name= (Samer, Essam, Ahmed);

- الآن أصبح لدينا متغير من النوع **name** مثله مثل (char و real و integer) و باقي المتغيرات يتم الإعلان عنه في قسم المتغيرات تماما مثل بقية المتغيرات

Var

N: name;

مثال على الجمل المعرفة Type

Type

```
car = (bmw , mazda ,honda ,ford);  
color = (red ,green ,blue);  
str20 = string[20];
```

var

```
mycar: car;  
mycolor: color;  
name:str20;
```

- هذه العبارات تسمى Enumerated Types أي الجمل المعرفة مسبقا من قبل المبرمج، فنجد أن المتغير من النوع mycolor والمتغير mycar من النوع car و color لا يمكن أن يخرج خارج إطار قيمة المتغير mycolor ، بمعنى أن المتغير أيضا (عند إسناده مثلا) يجب أن تكون قيمة الإسناد هي احد القيم التي يحملها المتغير الجديد color وهي (blue و red و green).
- فكما أن المتغير من النوع الصحيح integer لا يقبل الأعداد الكسرية ٢.٦ ، فالمتغير mycar لا يقبل إلا احد عناصره وهي (bmw, mazda, honda, ford) وهكذا

بعض الأوامر التي تتعامل مع الجمل المعرفة Type

١- الأمر **ord**: له وظيفتان الأولى هي تحديد موقع الجمل المعرفة (في حالة ASCII و الثانية هي إعطاء قيمه الحرف بصيغه الأسكي).

- مثال على الوظيفة الأولى (في البرنامج السابق)

```
ord(bmw); {=0}
```

```
ord(mazda); {=1}
```

- مثال على الوظيفة الثانية

```
N:= ord(a); {n=65}
```

٢- الأمر **chr**: أي عكس الدالة ASCII يقوم بإعطاء الرقم للحرف المدخل ، طبقا لمعايير ord

- مثال/ مع ملاحظه ان قيمه الحرف small تختلف عن capital

```
chr(65); {=a}
```

٣- الأمرين **pred** و **succ**: الأول يقوم بإعطاء القيمة أو الجمل التالية (في الجمل المعرفة)

الثاني يقوم بإعطاء القيمة أو الجمل السابقة (في الجمل المعرفة)

- مثال

```
succ(bmw); {=mazda}
```

```
pred(mazda); {=bmw}
```

أمثلة عن الجمل المعرفة Type

Type

```
car = (bmw=1,mazda ,Honda  
      ,ford=7,Volvo);
```

Begin

```
ord (bmw); {=1}  
ord (mazda); {=2}  
ord (honda); {=3}  
ord (ford); {=7}  
ord (volvo); {=8}
```

End.

Type

```
color= (red ,green ,blue);
```

Var

```
Mc : color;
```

Begin

```
succ(red); {green}  
pred(blue); {green}  
ord(blue); {2}
```

End.

بنية المجموعة Set Structure

- المجموعة: هي بنية معطيات نظامية تكون عناصرها من نوع قاعدي واحد و تتكون المجموعة من عدد منتهي من العناصر المختلفة و المرتبة من نوع واحد.
- المجموعة Set -

العنصر 1	العنصر 2	العنصر 3		العنصر n
----------	----------	----------	------	----------

- و يتم التعريف عنها بواسطة التعليمة Type على الشكل التالي في بداية البرنامج:

Type T = Set of To;

Var

x : T ;

- التطبيقات العملية للمجموعات المستخدمة في البرمجة يجب أن يكون النوع القاعدي To منتهياً، و عدده الأساسي غير كبير نسبياً، و يعد النوع البدائي البسيط من أفضل الأنواع المستخدمة كقاعدة للنوع المجموعاتي.
- و بشكل عام لا يمكن قراءة عناصر مجموعة أو طباعتها، كما هو الحال بالنسبة للنوع البدائي البسيط.

بنية المجموعة Set Structure

- مثال: مجموعة تحتوي على أسماء الموظفين، أو درجات الطلاب، أو ألوان الكرات في سلة الألعاب، الخ ...

Type Date = Set of 1.. 31;

Type Marks = Set of 0 .. 100;

Type Class = Set of String [45];

Type Color = (White, Red, blue, yellow, green) ;

Type Boll = Set of Color;

Var D : Date ;

M: Marks;

Name: array [1 .. 50] of Class;

B: array [1 .. 5] of Boll;

العمليات الأساسية المعرفة على المجموعات

• يعرف على متحولات النوع المجموعاتي العمليات الأساسية التالية:

١. التقاطع و يرمز له بالعملية (*). تدعى " ضرب مجموعتين "

٢. الاتحاد و يرمز له بالعملية (+). " جمع مجموعتين "

٣. الفرق و يرمز له بالعملية (-).

٤. الانتماء و يرمز له بالعملية (in).

٥. الإسناد: يمكن أن نسند قيمة متحول مجموعاتي إلى آخر من نفس النوع .

٦. المقارنة بين متحولين بالإشارة (=) أو (≠).

٧. عملية الاحتواء و رمزها (≤)، و لا تستخدم عملية الاحتواء التام (⊆).

• تعطى أفضلية التنفيذ الأولى لعملية التقاطع، تليها عمليتا الاتحاد و الفرق و لهما نفس الأفضلية، و أخيراً عملية الانتماء، حيث تنسب عملية الانتماء إلى العمليات المنطقية لأنها تحمل القيمتين True " ينتمي للمجموعة " أو False " لا ينتمي للمجموعة ".

$$R * S + T = (R * S) + T$$

$$R - S * T = R - (S * T)$$

$$R - S + T = (R - S) + T$$

$$X \text{ in } S + T = X \text{ in } (S + T)$$

مثال ١: اكتب برنامجاً للحصول على أسماء جميع أيام الأسبوع من السبت إلى الجمعة؟

Program ex (input, output);

Type

days = (Sunday, Monday, Tuesday,
Wednesday, Thursday, Friday, Saturday);

Var

allday : set of days;

workday : set of Monday .. Friday;

thisday : days;

Begin

thisday:= Monday;

If thisday in workday then

writeln('This day I'm in work')

Else

Writeln ('This day I'm on holiday');

End.

المجموعة **Set** في البرمجة مشابهة للرياضيات،

$$S = \{1, 5, 8, 9\}$$

أما في البرمجة فيتم تعريف المجموعة كالتالي:

Type

days = (sut, sun, mon, tues, wed, thurs, fri);

Var

Allday : set of days ;

المتغير **allday** عناصره جميع الأيام (السبت – الجمعة)

إذا قمنا بتغيير **thisday:= Monday** إلى **thisday:=**

Saturday سيتغير الخرج.

Exclude(myday, Friday); لحذف احد الأيام لسبب ما

و بذلك يتم استبعاد يوم الجمعة من أيام الأسبوع.

Include(myday, Friday); ولإرجاع أو أضافه يوم

ملاحظة مهمة: الأمرين **include** و **exclude** لا يعملان

في نسخه تيربو باسكال .

مثال ٢: اكتب برنامجاً للحصول على مجموعة مربعات الأعداد الصحيحة التي لا تزيد على ٣٠، و مجموعة أسس العدد ٢ التي لا تزيد أيضاً على العدد ٣٠، ثم اطبع المجموعة الحاصلة من تقاطعهما ؟

Program Numbers (output);

Type intset= set of 0 .. 30;

Var P, set1, set2 : intset;
x, i : integer;

Begin

set1 := [];

i := 1;

x := sqr (i);

while x <= 30 Do

Begin

set1 := set1 + [x];

i = i + 1;

x := sqr (i);

end;

set2 := [];

x := 1;

while x <= 30 Do

Begin

set2 := set2 + [x];

x := x * 2;

end;

P := set1*set2;

For i = 1 to 30 Do

IF i in P then Writeln (i);

End.

السجلات Record

Program progame (input, output);

Uses unit1 , unit2

Type

:

Var

:

Procedure & function

:

Begin {main}

:

End.

كما في المصفوفات ، السجلات هي عبارة عن عدد معين من المتغيرات تحت اسم واحد ، لكن المصفوفات تستطيع أن تحمل نوع واحد من البيانات ، أما السجلات من الممكن أن تحمل جميع أنواع البيانات (المتغيرات) بما فيها المصفوفات.

- فعليا السجلات هي نوع من أنواع البيانات المعرفة وتتطوي تحت قسم type وفي اغلب الأحيان تكون قبل الإعلان عن المتغيرات . نظره حول برنامج شبه متكامل بباسكال :

بنية السجل Record Structure

- غالباً ما نواجه في حياتنا العملية حاجة إلى التعامل مع بيانات من أنواع مختلفة و مترابطة " متعلقة بشخص أو حالة معيّنة "، مثل البيانات الخاصة بذاتية الطلاب (اسم، كنية، تاريخ ميلاد، عنوان، رقم هاتف .. الخ)، لذا يتوجب علينا استخدام بنية بيانات برمجية نظامية يطلق عليها اسم السجلات، و يتم عادة بناء بنوك المعلومات اعتماداً عليها.
- السجل: هو بنية معطيات نظامية مترابطة يمكن أن تكون عناصرها من عدة أنواع قاعدية و يتكون السجل من عدة حقول، و كل حقل يحتوي على بيانات من نوع قاعدي واحد بسيط أو مركب " أي يمكن أن يكون نوع بنيوي معرّف عليه في بداية البرنامج أو له نوع متجهي "، و يتميز بإمكانية الوصول المباشر إلى العنصر عن طريق اسم الحقل و اسم السجل يفصل بينهما نقطة .

- السجل Record -

الحقل 1 Field 1	الحقل 2 Field 2	الحقل 3 Field 3		الحقل n Field n
-----------------	-----------------	-----------------	------	-----------------

بنية السجل Record Structure

و يتم التعريف عنه بواسطة التعليمة Type على الشكل التالي في بداية البرنامج:

Type T = record

S1 : T1 ;

S2 : T2 ;

.

.

Sn : Tn ;

end ;

Var x : T ;

- يقبل كل حقل (S1, S2,.. Sn) من حقول السجل T، أي نوع من أنواع البيانات القياسية البسيطة (أعداد صحيحة، حقيقية، بوليانية، محارف و رموز) أو أي نوع آخر يتم التعريف عنه من قبل المبرمج في بداية البرنامج، كما يمكن أن يحتوي الحقل على بنية بيانات مركبة (متجهة أو مصفوفة ذات أبعاد). و يعيّن العدد الأساسي لهذا النوع بالعلاقة:

$$\text{Card (T)} = \text{Card (T1)} \times \text{Card (T2)} \times \dots \times \text{Card (Tn)}$$

بنية السجل Record Structure

- تعتبر عملية الوصول إلى البيانات الموجودة في حقول السجلات من العمليات الأساسية التي يتم تنفيذها على السجلات، و كما ذكرنا سابقاً أنه يمكننا الوصول المباشر إلى العنصر عن طريق كتابة اسم الحقل و اسم السجل يفصل بينهما نقطة، مثال لدينا سجل T و يحتوي على مجموعة حقول (S1, S2, .. Sn)، فعندما نريد الوصول إلى أي حقل يتوجب علينا كتابة (T.S1, T.S2, .. T.Sn) للتعامل معه.

T.S1:= Etc

If T.S2 ≤ Etc

... If T.Sn ≠ Etc

و لتسهيل التعامل مع البيانات الموجودة في حقول السجلات يمكننا لغة باسكال من استخدام تعليمة with التي ترتبط باسم السجل الذي سيتم التعامل معه، بعدئذٍ نقوم بكتابة اسم الحقل بدون اسم السجل (S1, S2, .. Sn)، و ذلك على الشكل التالي:

with T Do

S1:= etc

If S2 ≤ etc

If Sn ≠ etc

السجلات Records - مثال ١

Program example (I,O);

Type

data = **record**

name : string;

age : integer;

tel : integer ;

end;

Var

da: data ;

Begin

da.name := 'Haysam' ;

da.age := 40;

da.tel := 0932119415 ;

End.

- اسم البرنامج

- الإعلان عن السجلات

- الإعلان عن حقول السجلات

- نهاية الإعلان عن السجلات

- الإعلان عن المتغيرات : السجل لن يستخدم

بعد الإعلان عنه ، إلا إذا كان هناك متغير

يحمل اسم السجل da.

- **جمله التعيين**

- نأتي إلى البرنامج الرئيسي ، وكيفيه إدخال

القيم إليه ، عن طريق جمل التعيين أو جمل

الإدخال read و readln

السجلات Records (طريقة ثانية) - مثال ١

Program example (I,O);

Type data = record
 name : string;
 age : integer;
 tel : integer ;

end;

Var da: data ;

Begin

With da do

 name := 'Haysam' ;

 age := 40;

 tel := 0932119415 ;

end;

End.

- اسم البرنامج

- الإعلان عن السجلات

- الإعلان عن حقول السجلات

- نهاية الإعلان عن السجلات

- السجل لن يستخدم بعد الإعلان عنه ، إلا إذا

- كان هناك متغير يحمل اسم السجل da ،
هذه هي أهم نقط في الإعلان عن السجلات .

- طريقة ثانية لجملته التعيين

- نأتي إلى البرنامج الرئيسي ، وكيفيه إدخال
القيم إليها ، عن طريق جملة with

السجلات (جمل الإدخال) - مثال ١

Begin

Read(da.name);

Read(da.age);

Read(da.tel);

End.

Begin

With da do

Begin

Read(name);

Read(age);

Read(tel);

End;

End.

- **جمله الإدخال:** يتم كتابه جمله القراءة ثم المتغير (الذي يحمل السجل) ثم نقطه ثم اسم الحقل المراد وهكذا ..

- **طريقة ثانية لإدخال قيم المتغيرات:** يتم في البداية كتابة الكلمة **with** ثم نضع المتغير الذي يحمل السجل ، ، ونضع بداخله الجمل المراد **block** بعدها نقوم بفتح **do** ثم نضع كلمه إدخالها . إما عمليات إسناد ، أو جمل إدخال قيم .

السجلات (جمل الطباعة) - مثال ١

```
writeln (da.name);  
writeln (da.age);  
writeln (da.tel);
```

With da do

Begin

```
writeln (name);  
writeln (age);  
writeln (tel);  
End;
```

- **جمله الطباعة:** يتم كتابه جمله القراءة ثم المتغير (الذي يحمل السجل) ثم نقطه ثم اسم الحقل المراد وهكذا ..

- **طريقة ثانية لطباعة قيم المتغيرات:** يتم في البداية كتابة الكلمة **with** ثم نضع المتغير الذي يحمل السجل ونضع بداخله الجمل المراد **block** بعدها نقوم بفتح **do** ثم نضع كلمه إدخالها . إما عمليات إسناد ، أو جمل إدخال قيم .

السجلات و المصفوفات

- أيضا من الممكن أن تستخدم مصفوفة من السجلات ، نرجع إلى المثال الأول في قسم المتغيرات سيكون الإعلان بعد أن عرفنا السجل `data`

`da : array [1..10] of data ;`

- الطريقة الأولى باستخدام المصفوفات مباشرة:

`da[1].name := 'Haysam';`

- الطريقة الثانية باستخدام `with ... do`

With `da[1]` **do**

Begin

`name := 'Haysam';`

End;

ومن الممكن استخدام حلقة (تكرار) لقراءة جميع قيم السجل (بعد وضعه في مصفوفة).

مثال ٢: اكتب برنامج يقرأ معلومات ٣ طلاب ومن ثم يطبع هذه المعلومات على الشاشة ؟ علماً بأن معلومات الطالب هي (الاسم ، العمر ، الهاتف).

Program database (input, output);

Type

data = record
name: string;
age, tel: integer;

End;

Var

da: array [1..3] of data ;
l:integer;

Begin

Writeln('please fill the information : ');

For i:= 1 to 3 do

Begin

Writeln('NOW .. student no ',i);

With da [i] do

Begin

Write ('Name: '); Readln(name);

Write ('age: '); Readln(age);

Writeln('tel: '); Readln(tel);

End;

End; {for}

Circsr;

For i:= 1 to 3 do

Begin

With da[i] do

Begin

Writeln(' student no ',i);

Write ('name: '); Writeln(name);

Write ('age: '); Writeln(age);

Write ('tel: '); Writeln(tel);

End;

End; {for}

Writeln;

End.

مثال ٢: اكتب برنامج يقرأ معلومات ٣ طلاب ومن ثم يطبع هذه المعلومات على الشاشة ؟ علماً بأن معلومات الطالب هي (الاسم ، العمر ، الهاتف).

- **شرح البرنامج:** في البداية تم تعريف متغير معرف مسبقاً (سجل : data)، أي أنه هو أحد أنواع type من نوع record و تم وضع ثلاث حقول وهي (name الاسم ، tel الهاتف ، age العمر) المطلوبة في المثال.
- بعدها أصبح عرفنا متغير da من النوع الاسم Data في قسم الإعلان عن المتغيرات Var ، ثم تم الإعلان على المتغير عبارة عن مصفوفة مكونة من ثلاثة عناصر (data) ومتغير i من النوع الصحيح (يستخدم في التكرار).
- بعد بداية البرنامج الرئيسي begin:
- السطر الأول: جملة طباعه نص، سيطبع على الشاشة فور تنفيذ البرنامج .
- السطر التالي حلقه ستستخدم لإدخال بيانات (الاسم ،العمر،الهاتف) لثلاث طلاب .
- السطر التالي طباعة جملة نصية + قيمه متغير i ثم البدء بإدخال قيم للمتغيرات (الاسم و الهاتف و العمر) و لثلاث مرات.
- الجملة clrscr معناها تنظيف الشاشة clear screen.
- الحلقة الثانية لطباعه بيانات الطلاب الثلاث، وهي شبيهة بالحلقة السابقة (الأولى).
- توقف البرنامج مع تعليمة writeln ثم إنهاء البرنامج بتعليمة End

مثال Example

- مثال: لدينا مجموعة من الطلاب عددهم (١٠٠) مئة طالب، و لنفرض لدينا سجل الطالب Student يتألف من عدة حقول (الاسم الأول First Name، أسم الأب Father Name، الكنية Family Name، تاريخ الولادة Birth Date، الجنس Sex، درجات المقررات الدراسية لثلاثة مواد "رياضيات، فيزياء، لغة"، المطلوب كتابة برنامج يقوم بإدخال بيانات الطلاب المذكورة أعلاه تمهيداً لتنفيذ العمليات التالية عليها:
١. حساب عدد الطلاب الذكور MSum و عدد الطالبات الإناث FSum ؟
 ٢. حساب عدد الطلاب الذكور الناجحين في المقررات الدراسية الثلاثة SSm ؟
 ٣. طباعة اسم الطالبة الأصغر عمراً Fmin ؟
 ٤. طباعة قائمة بأسماء الطلاب و معدل درجاتهم Average في المواد الثلاثة ؟
 ٥. طباعة قائمة بأسماء الطلاب و الطالبات المتفوقين و الذين معدلهم ٨٠% فما فوق ؟
 ٦. طباعة قائمة بأسماء الطلاب و الطالبات تكون مرتبة وفق التسلسل الأبجدي لأسمائهم ؟

حل المثال ٢ Example

سنقوم بكتابة جزئية البرنامج التي يمكننا من تعريف البيانات في ذاكرة الحاسب فقط

Program School (input, output);

Type n max = 1 .. 100;

BirthDate = record

Day: 1 .. 31;

Month: 1 .. 12;

Year: 1970 .. 2100;

End;

Student = record

FirstN, FatherN, FamilyN: string[25];

Date: BirthDate;

Sex: (Male, Female);

Math: 0 .. 100;

Phys: 0 .. 100;

Eng: 0 .. 100;

End;

Stud = array [n max] of Student;

Var

St: Stud;

Fmin: BirthDate;

n: n max;

i , j , MSum , FSum , SSm: Integer;

Average: Real;

Name: String [25];

الطريقة الأولى: سنقوم بكتابة جزئية البرنامج التي تمكننا من إدخال البيانات إلى ذاكرة الحاسب فقط

```
Begin
Write ( ' Please Enter the Number of Students n = ' );
Readln ( n );
For i = 1 to n do
    Begin
        Write ( ' Enter First Name := ' );
        Write ( ' Father Name := ' );
        Write ( ' Family Name := ' );
        Write ( ' Enter Day of BirthDate :=');
        Write ( ' Month of Birth Date :=');
        Write ( ' Year of Birth Date := ');
        Write ( ' Enter Sex of Student :=');
        Write ( ' Enter Mathematic Mark :=');
        Write ( ' Enter Physic Mark :=');
        Write ( ' Enter English Mark :=');
        readln (St [ i ]. FirstN);
        readln (St [ i ]. FatherN);
        readln (St [ i ]. FamilyN);
        readln (St [ i ].Date.Day);
        readln (St [ i ].Date.Month);
        readln (St [ i ].Date.Year);
        readln ( St [ i ].Sex );
        readln ( St [ i ].Math );
        readln ( St [ i ].Phys );
        readln ( St [ i ].Eng );
    end; {for}
End.
```

الطريقة الثانية : سنقوم بكتابة جزئية البرنامج التي تمكننا من إدخال البيانات إلى ذاكرة الحاسب فقط، باستخدام تعليمة **with** مع السجلات.

Begin

Write (' Please Enter the Number of Students n = '); Readln (n);

For i = 1 to n do

With St [i] do

Begin

Write (' Enter First Name := ');

readln (FirstN);

Write (' Father Name := ');

readln (FatherN);

Write (' Family Name := ');

readln (FamilyN);

With Date do

Write (' Enter Day of Birth Date :=');

readln (Day);

Write (' Month of Birth Date :=');

readln (Month);

Write (' Year of Birth Date := ');

readln (Year);

Write (' Enter Sex of Student :=');

readln (Sex);

Write (' Enter Mathematic Mark :=');

readln (Math);

Write (' Enter Physic Mark :=');

readln (Phys);

Write (' Enter English Mark :='); readln (Eng);

end ; {with}

end; {for}

End.