

جامعة حماة

كلية العلوم في مصيف / السنة الأولى

المادة: لغات البرمجة

المحاضرة الرابعة: المتجهات و المصفوفات في لغة باسكال

Array Structures in Pascal

العام الدراسي ٢٠١٨ - ٢٠١٩



استخدام الأحرف في كتابة برنامج باسكال

- مترجم لغة باسكال Pascal Compiler (بعكس اللغات الأخرى) يفضّل الطرف عن حالة الأحرف (عالية أو منخفضة). لذلك فإن التعريفات التالية `myname` ، `MyName` ، `Myname` ، `myName` و `MYNAME` كلها متساوية. و بشكل عام يعدّ هذا أمرا ايجابيا، ففي اللغات الحساسة لحالة الأحرف قد تحدث العديد من الأخطاء اللغوية بسبب الإهمال في مراعاة حالة الأحرف.
- **ملاحظة:** ربما الحالة الاستثنائية الوحيدة لقاعدة حساسية الأحرف في باسكال هي إجراء `Register` في حزمة المكونات، لا بد لها أن تبدأ بحرف `R` العالي، وذلك للمحافظة على التوافقية مع `C++` `Builder`.
- إلا أنه توجد بعض السلبيات.
- **أولا**، يجب أن ننتبه لأن تكون هذه التعريفات متساوية بالفعل، لذا يجب أن نتجنب استعمالها كعناصر مختلفة.
- **ثانيا**، يجب أن نكون حذرين قدر الإمكان عند استخدامنا للأحرف العالية (لتحسين قراءة البرنامج).
- توحيد استخدام حالة الأحرف ليس ملزما من قبل المترجم ولكنها عادة حسنة يحدّ إتباعها. الأسلوب المتبع هو تكبير الحرف الأول فقط من كل معرفّ `identifier` و عندما يكون المعرفّ مركّبا من عدة كلمات (لا يمكننا حشر فراغ في المعرفّ) ، فإن كل أول حرف من كل كلمة يجب أن يكون عاليا:
`MyLongIdentifier`
`MyVeryLongAndAlmostStupidIdentifier`

استخدام الأحرف في كتابة برنامج باسكال

- هناك حالات أخرى لا يأبه لها المترجم كالفراغات والأسطر الفارغة والمسافات (tabs) التي نقوم بوضعها في البرنامج. كل هذه الحالات مجتمعة تسمى بالفراغ الأبيض white space الفراغات البيضاء تستخدم فقط لتحسين قراءة البرنامج؛ و لا تؤثر في عملية الترجمة .
- لغة باسكال تسمح لنا بكتابة تعليمة واحدة موزعة على عدة أسطر، فالتعليمة الطويلة يمكن تجزئتها لتكون في سطرين أو أكثر. السلبية الوحيدة لإمكانية أن تكون التعليمات في أكثر من سطر هي أنه عليك أن تتذكر بأن تضع فاصلة منقوطة آخر كل تعليمة، أو بدقة أكثر، أن تفصل بين التعليمة والتي تليها. لاحظ أن القيد الوحيد هنا أن الجملة النصية الواحدة لا يمكن مدها لعدة أسطر.
- لا توجد قواعد ثابتة لاستخدام الفراغات و التعليمات متعددة الأسطر، إنما توجد بعض الأعراف التي تجعل برنامجك سوف يبدو أفضل عندما تقوم بطباعته على الورق. غير هذا فإن الأسطر الطويلة قد يتم تجزئتها من أي موضع حتى من منتصف الكلمة عند طباعة البرنامج .
- عندما يكون للوظيفة أو الإجراء عدة محددات Parameters ، فإن العادة المتبعة هي وضع هذه المحددات في سطر آخر.
- نستطيع ترك سطر بكامله أبيضاً (فارغاً) قبل وضع أي تعليق أو ملاحظة، أو لتقسيم جزء كبير من التعليمات إلى أجزاء أصغر. هذه الفكرة البسيطة بإمكانها تحسين قراءة البرنامج، سواء على الشاشة أو عند طباعتها.
- استخدم الفراغات لفصل محددات استدعاء وظيفة function call ، وربما حتى فراغ قبل فتح الأقواس، أيضاً حافظ على فراغات لفصل رموز العمليات في التعبيرات البرمجية.

تعليمات لغة باسكال و الكلمات المفتاحية و الكلمات المحجوزة

- تمكنا لغة باسكال من تحديد أسماء بعض المعرفّات (المتغيرات و الثوابت) التي سنستخدمها لاحقاً في جسم البرنامج ضمن التعليمات أو المعادلات لأنها تصبح جزء منها، تقدم باسكال مجموعة من التعليمات والتعبيرات.
- **الكلمات المفتاحية:** هي كل المعرفّات المحجوزة من قبل (اوبجكت) باسكال، و التي لها معاني في لغة باسكال. دليل لغة دلفي (Help) يميّز بين الكلمات المحجوزة والتوجيهات كالتالي:
- الكلمات المحجوزة لا يمكن استخدامها كمعرفّات، بينما التوجيهات لا يجب استخدامها لنفس الغرض، حتى لو قبلها المترجم Compiler. عند كتابة البرنامج عليك تجنّب استخدام أية كلمة محجوزة كمعرفّ.
- في الجداول التالية يمكننا رؤية قائمة كاملة بالمعرفّات التي لها دورا خاصا في لغة اوبجكت باسكال (في دلفي ٤)، بما في ذلك الكلمات و الكلمات المحجوزة الأخرى.

الكلمة الرئيسية	الدور
absolute	directive (variables) (موجّه متغير)
abstract	directive (method) (موجّه مسار)
and	operator (boolean) (معامل بولي)
array	type نوع
as	operator (RTTI) (معامل RTTI)
asm	statement تعليمية
assembler	backward compatibility (asm) (توافقية مع السابق)
at	statement (exceptions) (تعليمية استثناءات)
automated	access specifier (class) (محدد دخول طبقة)
begin	block marker علامة حيز
case	statement تعليمية
cdecl	function calling convention قاعدة استدعاء وظيفة
class	type نوع
const	declaration or directive (parameters) (تحريف أو توجيه محددات)
constructor	special method مسار خاص
contains	operator (set) (عامل فئة)
default	directive (property) (توجيه سمة)
destructor	special method مسار خاص
dispid	dispinterface specifier محدد واجهة اطلاق
dispinterface	type نوع
div	operator عامل
do	statement تعليمية
downto	statement (for) (for) تعليمية
dynamic	directive (method) (توجيه مسار)
else	statement (if or case) (case او if) تعليمية
end	block marker تعليم حيز

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

except	statement (exceptions) نُعلِمة (استثناءات)
export	backward compatibility (class) نوافِية مع السابق
exports	declaration نُعرِيف
external	directive (functions) نُوجِبه
far	backward compatibility (class) نوافِية مع السابق
file	type نوع
finalization	unit structure بنية وحدة
finally	statement (exceptions) نُعلِمة
for	statement نُعلِمة
forward	function directive نُوجِبه وظيفه
function	declaration نُعرِيف
goto	statement نُعلِمة
if	statement نُعلِمة
implementation	unit structure بنية وحدة
implements	directive (property) نُوجِبه (مسار)
in	operator (set) - project strucure عامل (فئة) - بنية مشروع
index	directive (dipinterface) نُوجِبه
inherited	statement نُعلِمة
initialization	unit structure بنية وحدة
inline	backward compatibility (see asm) نوافِية مع السابقة
interface	type نوع
is	operator (RTTI) عامل
label	declaration نُعرِيف
library	program structure بنية برنامج
message	directive (method) نُوجِبه (مسار)
mod	operator (math) عامل (رياضي)
name	directive (function) نُوجِبه (وظيفة)
near	backward compatibility (class) نوافِية مع السابق

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

nil	قيمة خيالية
nodefault	directive (property) توجيه (مساير)
not	operator (boolean) عامل (بولي)
object	backward compatibility (class) توافقية مع السابق
of	statement (case) تعليمة
on	statement (exceptions) تعليمة
or	operator (boolean) معامل (بولي)
out	directive (parameters) توجيه (محددات)
overload	function directive توجيه وظيفة
override	function directive توجيه وظيفة
package	program structure (package) بنية برنامج (حزمة)
packed	directive (record) توجيه (تسجيل)
pascal	function calling convention طريقة استدعاء وظيفة
private	access specifier (class) محدد لوصول (طبقة)
procedure	declaration تعريف
program	program structure بنية برنامج
property	declaration تعريف
protected	access specifier (class) محدد لوصول (طبقة)
public	access specifier (class) محدد لوصول (طبقة)
published	access specifier (class) محدد لوصول (طبقة)
raise	statement (exceptions) تعليمة (اعتراضات)
read	property specifier محدد سمة
readonly	dispatch interface specifier محدد واجهة ارسال
record	type نوع
register	function calling convention طريقة لاستدعاء وظيفة
reintroduce	function directive توجيه وظيفة
repeat	statement تعليمة

الكلمات المفتاحية و الكلمات المحجوزة في لغة باسكال

requires	بنية برنامج (حزمة) (package) program structure
resident	توجيه (وظيفة) (directive) (functions)
resourcestring	نوع type
safecall	طريقة لاستدعاء وظيفة function calling convention
set	نوع type
shl	عامل (رياضة) (operator) (math)
shr	عامل (رياضة) (operator) (math)
stdcall	طريقة لاستدعاء وظيفة function calling convention
stored	توجيه (سمة) (directive) (property)
string	نوع type
then	statement (if)
threadvar	تحريف declaration
to	تعليمات (for) statement
try	تعليمات (استثناءات) (statement) (exceptions)
type	تحريف declaration
unit	بنية وحدة unit structure
until	تعليمات statement
uses	بنية وحدة unit structure
var	تحريف declaration
virtual	توجيه (مسار) (directive) (method)
while	تعليمات statement
with	تعليمات statement
write	محدد سمة property specifier
writeln	محدد واجهة ارسال dispatch interface specifier
xor	عامل (بولي) (operator) (boolean)

بنية المتجهة Array Structure

المتجهة: هي بنية معطيات نظامية متجانسة تكون جميع عناصرها من نوع قاعدي واحد بسيط قياسي أو بدائي و تتميز بإمكانية الوصول المباشر إلى أي عنصر داخلها دون المرور بالعناصر الأخرى و ذلك بمساعدة دليل المتجهة Index الذي يحدد موقع العنصر الذي نرغب بتنفيذ العمليات عليه ضمن المتجهة $x [i]$ ، و يتم التعريف عنها بواسطة التعليمة Type على الشكل التالي في بداية البرنامج:

Type T = array [I] of To;

بناءً على ما سبق نرى أن تعريف المتجهة (T) يتم بدلالة نوعين مصممين هما:

- نوع القاعدة (To) الذي يحدد نوع عناصر المتجهة
- النوع المعرّف لدليل المتجهة [I].

بنية المتجهة Array Structure

أمثلة:

Type Row = array [1 .. 80] of integer ;

Type c = [80 .. 100] of real ;

تعريف
المتجهة

Type Alfa = array ['a' .. 'z'] of char;

كما يتم حجز متحولات من النوع المعرّف في قسم التصريح Var و كذلك يتم التعريف عن الدليل Index الذي يعتبر المفتاح الأساسي للوصول المباشر إلى أي عنصر من عناصر المتجهة، و تعتبر المتجهات من أكثر بنى المعطيات استخداماً في البرمجة.

Var x : Row ;

y : c ;

تعريف
المتغيرات

z : Alfa ;

١٠ i : integer ;

مثال: أوجد مجموع عناصر المتجهة التالية SUM

Program SUM;

تعريف المتجهة

Type MAT = array [1..n] of integer;

Var

M: MAT;
i, n, Sum: integer;

تعريف
المتغيرات

Begin

write ('Input n =');
readln (n);
Sum:= 0;

إدخال عدد
عناصر المتجهة

for i:=1 to n do

begin

إدخال عناصر المتجهة

write('Input M [, i ,] =');
readln(M[i]);
Sum:= Sum +M [i];

end;

write ('Sum= ', Sum);
Readln();

طباعة
مجموع
عناصر
المتجهة

End.

مثال: أوجد مجموع و متوسط عناصر AVG المتجهة التالية

Program AVG ;

تعريف المتجهة

Type MAT = array [1..n] of integer;

Var

M: MAT;

تعريف المتغيرات

I, n, Sum : integer;

Avg: real;

Begin

Sum:= 0;

إدخال عدد عناصر المتجهة

write ('Input n ='); readln (n);

for i:=1 to n do

begin

إدخال عناصر المتجهة

write('Input M [', i , '] =');

readln(M[i]);

Sum:= Sum + M [i];

end;

Avg := Sum / n;

طباعة مجموع

writeln ('Sum= ', Sum);

عناصر

writeln ('Avg= ', Avg);

المتجهة

ومتوسطها

Readln();

End.

مثال: أوجد العنصر الأصغر MIN في المتجهة التالية

Program MIN;

تعريف المتجهة

Type MAT = array [1..n] of integer;

Var

M: MAT;
l, n, min : integer;

تعريف
المتغيرات

Begin

write ('Input n =');
readln (n);

for i:=1 to n do

إدخال عناصر
المتجهة

begin

write('Input M [, i ,] =');
readln(M[i]);

end;

min:=M[1];

for i :=2 to n do

if M [i] < min then

min := M [i];

إيجاد العنصر
الأصغر

write ('Min = ', min);

Readln();

طباعة العنصر
الأصغر

End.

مثال: أطبع عناصر المتجهة التالية والعنصر الأكبر MAX

Program MAX;

Type MAT = array [1..n] of integer;

تعريف
المتجهة

Var

M: MAT;
l, n, max : integer;

تعريف
المتغيرات

Begin

write ('Input n =');

readln (n);

for i:=1 to n do

begin إدخال عناصر المتجهة

write('Input M [, i ,] =');

readln(M[i]);

end;

max :=M[1];

for i :=2 to n do

if M [i] > max then

max := M [i];

إيجاد العنصر
الأكبر

for i :=1 to n do

write (M [i]) ;

طباعة عناصر
المتجهة

write ('Max = ' , max) ;

Readln();

طباعة العنصر
الأكبر

End.

وظيفة Homework

- ١- اكتب برنامجاً يقوم بإدخال عشرة أرقام صحيحة عشوائياً، ثم يرتبها تصاعدياً بمساعدة المتجهات ؟
- ٢- اكتب برنامجاً يقوم بإدخال أسماء عشرة طلاب، ثم يرتبها أبجدياً بشكل تصاعدي بمساعدة المتجهات ؟
- ٣- اكتب برنامجاً يقوم بإدخال درجات الحرارة في مدينة حمص لمدة أسبوع، ثم يظهر متوسط درجات الحرارة لهذا الأسبوع و درجة الحرارة العظمى و ذلك بمساعدة المتجهات ؟

متجهة المتجهات و المصفوفة Array of Array and Matrix Structure

- **المصفوفة:** هي بنية معطيات نظامية متجانسة تكون جميع عناصرها من نوع قاعدي واحد مركب أي يمكن أن يكون نوع بنيوي معرّف عليه في بداية البرنامج أو له نوع متجهي " متجهة متجهات " وتتميز بإمكانية الوصول المباشر إلى أي عنصر فيها دون المرور بالعناصر الأخرى و ذلك بمساعدة مجموعة الأدلة للمتجهات Index's التي تحدد موقع العنصر الذي نرغب بتنفيذ العمليات عليه ضمن المتجهة .. $x[i][j][k]$ ، أو ... $x[i, j]$ ، و يتم التعريف عنها بواسطة التعليمة Type على الشكل التالي في بداية البرنامج:

Matrix : array [i] of To;

تعريف المصفوفة ذات البعد واحد

Mat : array [i , j] of To;

تعريف المصفوفة ذات البعدين – طريقة (١)

أو

Mat : array [i] of array [j] of To;

تعريف المصفوفة ذات البعدين – طريقة (٢)

مثال: برنامج يقرأ ١٠ أحرف ، ثم يقوم بطباعتها بالعكس

Program ex4 (input,output);

Var

i: integer;

x: array [1..10] of char;

Begin

For i:= 1 to 10 do

Read(x[i]);

For i:=10 downto 1 do

Write(x[i]);

End.

- التعريف باسم البرنامج

- التعريف بالمتغيرات

- بداية البرنامج

- إدخال عناصر المصفوفة

- طباعة عناصر المصفوفة عكسياً

- نهاية البرنامج

مثال: أوجد العنصر الأصغر MIN في المصفوفة التالية

Program MIN;

VAR

M: array [1..n] of integer;
i, n, min : integer;

تعريف
المصفوفة
والمتغيرات

Begin

write ('Input n =');
readln (n);

إدخال عدد
عناصر المصفوفة

for i:=1 to n do

begin

write('Input M [', i , '] =');
readln(M[i]);

إدخال عناصر المصفوفة

end;

min:=M[1];
for i :=2 to n do
if M [i] < min then
min := M [i];

إيجاد العنصر
الأصغر

write ('Min = ', min);
Readln();

طباعة العنصر
الأصغر

End.

مثال: أوجد العنصر الأكبر MAX في المصفوفة التالية و أطلع عناصرها

Program MAX;

Var

M: array [1..n] of integer;
i, n, max : integer;

تعريف
المصفوفة
والمتغيرات

Begin

write ('Input n ='); إدخال عدد
readln (n); عناصر المصفوفة

for i:=1 to n do

begin

إدخال عناصر المصفوفة
write('Input M [, i ,] =');
readln(M[i]);

end;

max :=M[1];
for i :=2 to n do
if M [i] > max then
max := M [i];

إيجاد
العنصر
الأكبر

for i :=1 to n do
write (M [i]);

طباعة عناصر
المصفوفة

writeln();

write ('Max = ', max);
Readln();

طباعة
العنصر
الأكبر

End.

مثال: رتب المصفوفة التالية تصاعدياً

Program Sort 1;

Var

M: array [1..n] of integer;
i, j, n, min ,temp: integer;

تعريف
المصفوفة
والمغيرات

Begin

write ('Input n =');
readln (n);

إدخال عدد
عناصر المصفوفة

for i:=1 to n do

إدخال عناصر
المصفوفة

begin

write('Input M [, i ,] =');
readln(M[i]);

end;

for j:=1 to n do

begin

إيجاد
العنصر
الأصغر

min:=M[j];
for i:= j+1 to n do

if M [i] < min then

begin

temp := M [i];
M [i]:= min;
min := temp;

ترتيب
عناصر
المصفوفة
تصاعدياً

end;

end;

for i:=1 to n do

write (M [i]);

طباعة
عناصر
المصفوفة

End.

مثال: رتب المصفوفة التالية تنازلياً

Program Sort 2;

Var

M: array [1..n] of integer;
i, j, n, max ,temp: integer;

تعريف
المصفوفة
والمغيرات

Begin

write('Input n =');

readln(n);

إدخال عدد
عناصر المصفوفة

for i:=1 to n do

begin

write('Input M [, i ,] =');

readln(M[i]);

end;

إدخال عناصر
المصفوفة

for j:=1 to n do

begin

max :=M[j];

for i:= j+1 to n do

if M [i] > max then

begin

temp := M [i];

M [i]:= max ;

max := temp;

end;

end;

for i:=1 to n do

write (M [i]);

إيجاد
العنصر
الأكبر

ترتيب
عناصر
المصفوفة
تنازلياً

طباعة
عناصر
المصفوفة

End.

المصفوفة ذات البعدين 2 X Array (Matrix)

- المصفوفة ذات البعدين: هي بنية معطيات نظامية متجانسة لها بعدان (n,m) و تكون جميع عناصرها من نوع قاعدي واحد بسيط أو مركب، أي يمكن أن يكون من نوع بنيوي معرّف عليه في بداية البرنامج أو له نوع متجهي " متجهة متجهات " و تتميز بإمكانية الوصول المباشر إلى أي عنصر فيها دون المرور بالعناصر الأخرى و ذلك بمساعدة مجموعة الأدلة (i , j) Index's التي تحدد موقع العنصر الذي نرغب بتنفيذ العمليات عليه ضمن المصفوفة [i , j] x، و يمكن التعريف عنها بعد تعليمة Var على الشكل التالي في بداية البرنامج:

Var

Matrix: array [i , j] of To; **تعريف المصفوفة ذات البعدين**

- نستخدم في المصفوفات ذات البعدين حلقتين متداخلتين ليتم قراءة البعد الأول i والبعد الثاني j .

Program example (input,output);

Var

تعريف المصفوفات

i, j , n, m : integer ;
x,a,b:array [1..n , 1..m] of integer ;

ذات البعدين

Begin

readln (n) ; readln (m) ;

إدخال أبعاد
المصفوفات

For i:= 1 to n do

begin

For j:=1 to m do

readln (a[i , j]) ;

end ;

إدخال عناصر
المصفوفة A

For i:= 1 to n do

begin

For j:=1 to m do

readln (b[i , j]) ;

end ;

إدخال عناصر
المصفوفة B

مثال: برنامج لضرب مصفوفتين A و B
لهما نفس البعدين n و m

ضرب عناصر المصفوفتين

for i:=1 to n do
begin

A و B و إسنادهم إلى
المصفوفة X

for j:= 1 to m do

x [i, j] := (a[i, j]) * (b[i, j]) ;

end ;

for i:= 1 to n do

begin

for j:= 1 to m do

writeln (x [i, j]) ;

end ;

طباعة عناصر
المصفوفة X

End.

Program example (input,output);

Var

i,j :integer; **تعريف المصفوفات ذات البعدين**
x,a,b:array [1..5,1..5] of integer;

Begin

For i:= 1 to 5 do **إدخال عناصر A المصفوفة**
begin
 For j:=1 to 5 do
 readln (a[i, j]);
end;

For i:= 1 to 5 do **إدخال عناصر B المصفوفة**
begin
 For j:=1 to 5 do
 readln (b[i, j]);
end;

مثال: برنامج لطباعة مصفوفة عناصرها
ناتج مجموع العنصرين المقابلين في
مصفوفتين مربعيتين

**جمع عناصر المصفوفتين
و إسنادهم إلى A و B
X المصفوفة**
for i:=1 to 5 do
begin
 for j:= 1 to 5 do
x [i, j] := (a[i, j]) + (b[i, j]);
end;

**طباعة عناصر
X المصفوفة**
for i:= 1 to 5 do
begin
 for j:= 1 to 5 do
 writeln (x [i, j]);
end;

End.

متجهة المتجهات و المصفوفة Array of Array and Matrix Structure

٢- لدينا مكتبة تحوي مجموعة أقسام، وكل قسم يحتوي مجموعة كتب، و كل كتاب مجموعة صفحات، و كل صفحة مجموعة سطور، وكل سطر مجموعة محارف.

```
Type Line = array [ 1 .. 80 ] of char ;  
page = array [ 1 .. 25 ] of Line ;  
Book = array [ 1 .. 500 ] of page ;  
Section = array [ 1 .. 1000 ] of Book ;  
Library = array [ 1 .. 10 ] of Section ;
```

```
V Type Matrix = array [ i , j ] of To;
```

```
Var x : Library ;
```

- حيث تم حجز مكان في ذاكرة الحاسب للمتحول x من النوع المعرّف في قسم التصريح Library و كذلك يجب التعريف عن الأدلة Index's

```
Var i, j, k, l, m : integer ; •
```

Homework وظيفة

- ١- أوجد العنصر الأكبر في مصفوفة من البعد ($n \times m$) و حدد مكان هذا العنصر في تلك المصفوفة ؟
- ٢- أوجد متوسط الهطولات المطرية في خمسة مدن سورية تم أخذ عيناتها على مدار شهر في فصل الشتاء ؟
- ٣- أوجد معدّل درجات كل طالب، إذا علمت أن عدد الخريجين ١٥ طالب و عدد المواد ٣٠ مادة، ثم اكتب اسم و علامة صاحب أعلى معدّل بين الخريجين ؟