سنبدأ أولاً بالتكلم عن المصفوفات ومن ثم حل تمرين الوظيفة 😊

المصفوفات: بعد أن تحدثنا عن المتجه أصالة البعد في المحاضرات السابقة فإن من السهل طرح مفهوم المقولة ثنائية البعد (المصفوفات) هنالك تشابه كبير بين المفهومين والفرق الوحيد هو إضافة بعد جديد.

♥ التصريح عن المصفوفة: نقوم بالتصريح عن المصفوفة كما يلي:

نوع المصفوفة [][] اسم مصفوفة ;

أو

نوع المصفوفة اسم مصفوفة [][] ;

• مثال: مصفوفة أعداد صحيحة // int [][] Matrix ;

مصفوفة محارف // char [][] Matrix ;

♥ حجز المصفوفة: لحجز المصفوفة نستخدم الكلمة الهامة new وذلك كالتالي

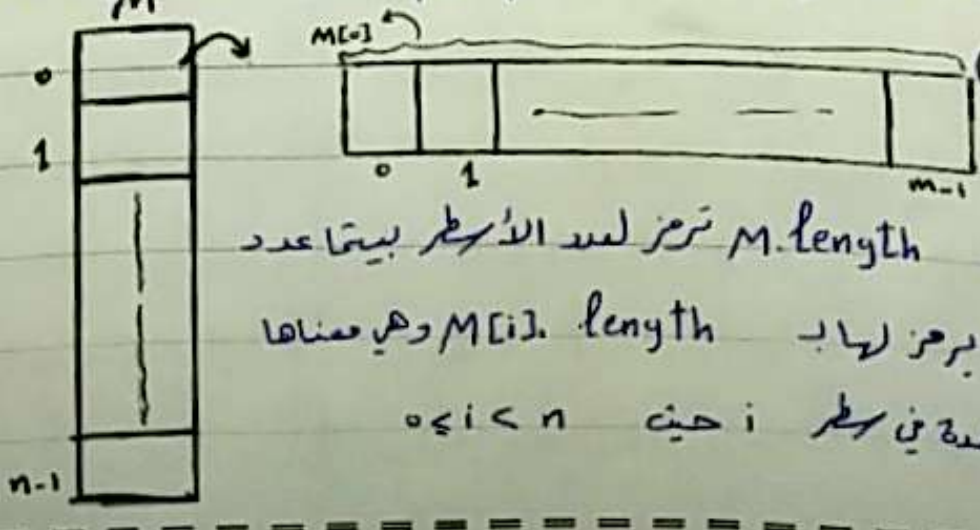
اسم المصفوفة = new نوع المصفوفة [عدد الأعمدة وهو عدد مدخل] [عدد الأسطر وهو عدد مدخل] ;

• مثال: Matrix = new char [n][m] ;

هنا حجزنا لمصفوفة محارف عدد أسطرها n وعدد أعمدتها m.

Note: لا يمكن الحجز إلا بعد التصريح.

* يعبر عن المصفوفة في الذاكرة على أنها متجهة عناصرها متجهات سطرية (كما هو موضح في الشكل التالي فإننا نعتبر M هي المتجهة العامودية وكل خلية من تلك المتجهة هي متجهة بحد ذاتها)



إذاً فإن M.length ترمز لعدد الأسطر بينما عدد الأعمدة يرمز لها بـ M[i].length وهي عددها عدد الأعمدة في سطر i حيث $0 \le i < n$

Scanned by CamScanner

Note: إن عدد الأعمدة ثابتة من أجل كل سطر لذلك لا فرق إذا استخدمنا M[0].length أو M[i].length لكن يجب الانتباه عند استخدام التعبير الأول يجب أن يكون المقول i معرف أي مصرح عنه.

♥ إذا كانت قيم عناصر المصفوفة معلومة مسبقاً قبل التنفيذ فيمكن إسنادها تلقائياً إلى المصفوفة، مثل من أجل المصفوفة $M=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix}$

نكتب في بيئة الـ Java الآتي:

```
int M[][]={{1,2,3},{4,5,6}};
```

هي مصفوفة التي عدد أسطرها 2 وعدد أعمدتها 3.

حل تمرين الوظيفة، السؤال الأول

قسم الحل

```
class Mymatrix{
public static void main(String[]args)
{ int n,m;
  char M[][];
  do {n=stdin.readInt();}while(n<=0);
  do{m=stdin.readInt();}while(m<=0);
  M=new char[n][m];
for(int i=0;i<M.length;i++)
  for(int j=0;j<M[i].length;j++)
    M[i][j]=Stdin.readChar();
  Char T[][];
T=new char[m][n];
  for(int i=0;i<m.length;i++)
```

قسم الشرح

- Mymatrix ← أنشأنا الصف
- main ← وضعنا داله
- n و m ← صرحنا عن الأبعاد
- M ← صرحنا عن مصفوفة محارف
- do-while ← دخلنا n وفق حلقة
- dowhile ← دخلنا m وفق حلقة
- T ← حجزنا المصفوفة من البعد nxm.
- حلقة الأولى for ← عم نوزع الأسطر فندخل أول عنصر بأول سطر وندخل حلقة ثانية على الأعمدة وهنا نقوم بإدخال عناصر المصفوفة.
- صرحنا عن مصفوفة محارف جديد T وحجزنا لها من البعد mxn ومن ثم أضفنا حلقة for لطباعة

(Mob; 0944-879460 / Pho: 011-2151436) خدمات كلية العلوم – دمشق حرم كلية العلوم Plus

Scanned by CamScanner

```
for (int j =0 ; j< M[i].length;j++)
   T[j][i] = M[i][j];
for (int i=0 ; i <T. length ; i++)
 { for(int j=0; j<T[i].length ; j++)
system.out.print (T[i][j]+" ");
system.out.println (); }
```

وومن ثم أرسلنا قيم المصفوفة M إلى المصفوفة T ومن ثم أخذنا حلقتين for وذلك لطباعة مصفوفة T وضعنا تعليمة طباعة فارغة لكي تنظم لنا كل مصفوفة

طلب الثاني:

```
char min = M[0][0];
int l,k  l=0;  k=0;
for(int i=0 ; i<M. length ; i++)
 for(int j=0; j < M[i].length;j++)
  if (M[i][j] < min)
  { min = M[i][j];
    l=i;  k=j; }
System.out.print ("min=" +min +" "
                  + l + " " +k);
System.out.println ();
```

أرسلنا min إلى العنصر الأول في مصفوفة صرحنا عن متحولات l و k وأرسلنا لهم قيم البداية ثم من حلقتين for اختبر الشرط (من هو عنصر الأصغر) ثم نقوم بعملية إرساد القيمة M[i][j] لـ min وغيرنا الأدلة ثم نقوم بعملية الطباعة.

طلب الثالث و الرابع:

~~for(int i=0 ; i<M.length ; i++)~~
~~for(int j=0 ; j<M[i].length ; j++)~~
~~{~~

```
if(M.length != M[0]. length)
system.out.println ("error");
else {for(int i=0 ; i<M.length ; i++)
 System.out.print (M[i][j]);
```

علماً أن ترتيب الخانات وفق جدول أسكي أرقام < أحرف كبيرة < أحرف صغيرة < أحرف لغة عربية.

اختبرنا شرط إذا كانت مصفوفة مربعة أو لا إذا لا سيطبع البرنامج خطأ وإذا مربعة سنعاير جد قطر رئيسي وقطرنا توفي وهنا طبعاً كلمة للتكلة من قطر رئيسي

(Mob; 0944-879460 / Pho: 011-2151436) خدمات كلية العلوم – دمشق حرم كلية العلوم Plus

Scanned by CamScanner

```
for(int i = M.length-1 ; i>=0 ; i--)
system.out.print( M[i][i-M.length+1]);
} //end of else
```

طلب الخامس:

```
char c = stdin.readChar();
boolean E = false;
for(int i=0; i < M.length ; i++)
  for(int j=0; j < M[i].length; j++)
  if ( M[i][j]==c)  E=true;
if(E) system.out.print (c+"is found");
else system.out.print(c+"is not found");
```

طلب السادس:

```
for(int i=0; i<M.length ; i++) {
  for(int j=0; j<M[i].length; j++)
if(i != 2  || j != M[i].length-2)
system.out.print (M[i][j] + " ");
else system.out.print (" ");
system.out.println( ); }
} // end of main
} // end of class My Matrix
```

لطباعة كلمة متكلة من عناصر قطر ثانوي مكونة برجياً عناصر قطر ثانوي دليلها i+j=n-1 ⇐ j=i-n+1 لذلك طبنا العناصر بالشكل M[i][i-M.length+1] حيث لم نعرفه لذلك استخدمناها بذلك صيغة.

أدخلنا محرف c
أدخلنا قيمة بولينية E وأعطيناها القيمة false.
حلقة for للصفوف
أدخلنا شرط بولياني وفرضناه صحيح
إذا كانت E صحيح سيطبع c موجود
وإلا سيكون c غير موجود.

حلقة for لسطر
حلقة for للأعمدة
إذا كان سطر يساوي 2 أو عامود يساوي ماقبل الأخير سيطبع العناصر
وإلا سيطبع فراغ أي أذا كان سطر يساوي 2 والعامود هو ماقبل الأخير.

انتهت دالة main وصف Mymatrix

انتهت المحاضرة ^^ --- ^^

(Mob; 0944-879460 / Pho: 011-2151436) دمشق حرم كلية العلوم