

كلية الدراسات المتوسطة - الأزهر
College of Intermediate Studies-Alazhar



تراكيب البيانات

مدرس المساق/أ. م. رائد خضير

القوائم

المادة العلمية إعداد
أ. م. رائد خضير

مفهوم ال ArrayList

- لاشك أنك انتبهت خلال تعاملنا مع المصفوفة في إصدارها التقليدي أنّ حجمها ثابت و لا يمكن زيادته خلال عمل البرنامج, و هذا الأمر جعلها غير مناسبة لكثير من البرامج.

- من الحلول لهذه المشكلة استخدام ما يعرف Dynamic Data Structure حيث الحجم يمكن زيادته خلال تنفيذ النظام, و من هذه التراكيب:

- ✓ Stack (المكدس)
- ✓ Queue (الطابور)
- ✓ Array List

مفهوم الـ ArrayList

- و تعتمد الـ ArrayList على حجم مبدئي قدره 4 عناصر, بمجرد الوصول له يتم تلقائي إضافة 4 عناصر جديدة و هكذا, و بالتالي فلا وجود لخطأ انتهاء الحجم, (يمكننا طلب عدد أكبر من الخانات من بداية العمل).
 - تعتبر الـ ArrayList من تراكيب البيانات المبنية سلفا في لغة الجافا, و لكي يمكننا التعامل معها نحتاج لتضمين المكتبة **Collections**
- ```
import java.util.ArrayList;
```

# مفهوم الـ ArrayList

- و يتم تعريف مصفوفة اسمها names من هذا النوع كما يلي:  
`ArrayList names = new ArrayList();`
- كما تلاحظ في تعريف الـ ArrayList لم نحدد عدد العناصر, و لم نستخدم [ ] و ذلك لأنه يعتبر صنف معرف سلفا Predefined class
- و بالتالي فتعريف أي مصفوفة من هذا النوع يتم كأنك تشتق كائن من صنف.

# مفهوم ال ArrayList

- و يمتلك صنف ArrayList مجموعة من الدوال التي يمكنك من خلالها التعامل مع عناصر المصفوفة, بعضها موضح فيما يلي.


| وظائفها                                                                                                 | الدالة                         |
|---------------------------------------------------------------------------------------------------------|--------------------------------|
| من خلالها يتم إضافة عناصر للمصفوفة في نهايتها.                                                          | add()                          |
| يتم من خلالها إضافة مصفوفة لمصفوفة أخرى من النوع ArrayList و تضاف العناصر في نهاية المصفوفة المضاف لها. | addAll()                       |
| تسمح بحذف كافة العناصر الموجودة داخل المصفوفة                                                           | clear()                        |
| دالة تستخدم لفحص وجود عنصر في المصفوفة من عدمه و هي بالتالي تعيد القيمة الشرطية (T, F)                  | contains()                     |
| تسمح لك بإدراج عنصر ما في موضع محدد داخل المصفوفة و ليس شرطاً في نهايتها                                | add(int index, Object element) |

مستخدماً خدمة البحث عبر الإنترنت , ابحث عن مزيد من الدوال مع مفهوم كل منها

# مفهوم ال ArrayList

```
1. public static void main(String[] args) {
2. ArrayList names = new ArrayList();
3. boolean po1 = names.add("Ali");
4. boolean po2 = names.add("hussam");
5. boolean po3 = names.add("hassan");
6. boolean po4 = names.add("Alaa");
7. boolean po5 = names.add("wael");
8. System.out.println();
9. System.out.println(" Position of PO1: " + po1 +
10. " Position of PO2: " + po2 + " Position of PO5: " + po5);
11. System.out.println(" number of names in the ArrayList: " +
names.size());
12. System.out.println(names.get(0));
13. }
```

متغير مُعرَّف سلفاً داخل  
الصفء و يعيد عدد العناصر  
التي تحتويها المصفوفة



```
Position of PO1: true Position of PO2: true Position of PO5: true
number of names in the ArrayList: 5
```

```
Ali
```

# ArrayList المفهوم

```
1. static void Main(string[] args)
2. {
3. public static void main(String[] args) {
4. ArrayList list1 = new ArrayList();
5. list1.add(40);
6. list1.add(50);
7. list1.add(60);
8. list1.add(70);
9. ArrayList list2 = new ArrayList();
10. list2.add(80);
11. list2.add(90);
12. list1.addAll(list2);
13. System.out.println(" == The elements of list1 are ==");
14. for (int i = 0 ;i<list1.size();i++)
15. System.out.println(" "+list1.get(i));
16. System.out.println();
17. } }
```





run:

```
== The elements of list1 are ==
40
50
60
70
80
90
```

# ArrayList مفهوم ال

```
1. static void Main(string[] args)
2. {
3. ArrayList list1 = new ArrayList ();
4. list1.add(40);
5. list1.add(50);
6. list1.add(60);
7. list1.add(70);
8. for(Object c:list1)
9. System.out.println(c);
10. }
```

Output - JavaApplication6 (run) ☒



```
run:
40
50
60
70
BUILD SUCCESSFUL (total time: 0 seconds)
```