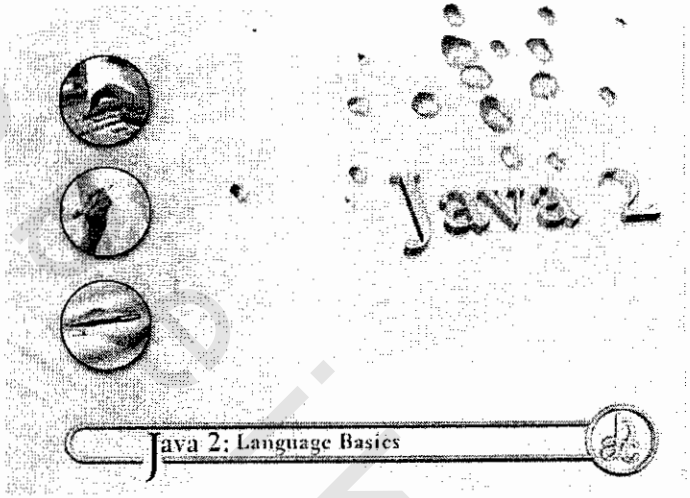
# الفصل الرابع عشر




# مجموعات Collections

في هذا الفصل سوف نتناول كيفية التعامل مع المجموعات Collections. سنتناول في هذا الفصل أشهر وأكثر المجموعات Collections استعمالاً وهم:

- الفصيلة ArrayList.
- الفصيلة Vector.
- الفصيلة LinkedList.
- الفصيلة LinkedHashSet.
- الفصيلة TreeSet.
- الفصيلة HashSet.
- الفصيلة TreeMap.
- الفصيلة LinkedHashMap.
- الفصيلة HashMap.
- الفصيلة Hashtable.

obeikan.com

## مقدمة:

- إن المجموعات Collections هي ببساطة عبارة عن هدف object يقوم بتجميع عدة عناصر في وحدة واحدة فقط.
- تستخدم المجموعات Collections للتعامل مع بيانات عناصرها حيث يمكنها إضافة أو حذف أو تعديل أي من العناصر التي تحتويها.
- ويعتبر أبسط مثال لتخيل المجموعات Collections هو المصفوفة Array ، حيث يتم إنشاء مصفوفة Array وتكون هذه المصفوفة Array محتوية علي عدد من العناصر المرتبطة ببعضها البعض.
- ولكن هناك اختلاف جوهري بين المصفوفة Array والمجموعات Collections حيث أن المصفوفة Array تحتوي علي عناصر جميعها من نفس النوع وذلك بعكس المجموعات Collections التي لا يلزم أن تكون عناصرها من نفس النوع ، كما أن المصفوفة Array لها حجم ثابت لا يتغير وذلك بعكس المجموعات Collections التي يمكن تغيير حجمها أثناء عمل البرنامج.
- في هذا الفصل سوف نتناول أشهر فصائل المجموعات Collections حيث سنتعرض إلي ثلاثة من أنواع المجموعات Collections وهم:
  1. القائمة List.
  2. المجموعة Set.
  3. الخريطة Map.
- وتحت كل مجموعة توجد العديد من الفصائل Classes التي سنتناولها كما هو واضح في الجدول التالي:

| النوع | اسم الفصيلة Class |
|---|---|
| القائمة List | ArrayList |
| القائمة List | Vector |
| القائمة List | LinkedList |
| المجموعة Set | LinkedHashSet |

| النوع | اسم الفصيلة Class |
|---|---|
| المجموعة Set | TreeSet |
| المجموعة Set | HashSet |
| الخريطة Map | TreeMap |
| الخريطة Map | LinkedHashMap |
| الخريطة Map | HashMap |
| الخريطة Map | Hashtable |

- لاحظ أن هذه الفصائل classes الموجودة في الجدول السابق هي المطلوب معرفتها لدخول امتحان المبرمج للإصدار 1.4 والمعروف باسم (Sun Certified Java Programmer SCJP 1.4).
- قبل الدخول في هذه الفصائل classes فلا بد من معرفة بعض المفاهيم الأساسية.

### المجموعات المفهرسة Ordered Collections:

عندما تكون المجموعة Collection مفهرسة Ordered فهذا يعني أنه يمكن قراءة بيانات العناصر الموجودة للمجموعة Collection عن طريق ترتيب معين ، فمثلاً إذا تخيلنا المصفوفة array فنجد أنه يوجد فهرس المصفوفة array index الذي يمكننا من خلاله قراءة عناصر المصفوفة array.

### المجموعات المرتبة Sorted Collections:

عندما تكون المجموعة Collection مرتبة Sorted فهذا يعني أن عناصرها مرتبة ترتيباً أبجدياً بالنسبة للنصوص ومرتباً تصاعدياً بالنسبة للأرقام.

- سنبدأ الآن في شرح كل فصيلة class من الفصائل classes السابقة بالتفصيل كما يلي في النقاط القادمة.

### الفصيلة ArrayList:

- تنتمي إلي القائمة List.
- تعتبر فصيلة مفهرسة Ordered وليست مرتبة Unsorted.

- تعتبر فصيلة غير متزامنة Unsynchronized.
- تقبل القيم null فيها.
- تقبل تكرار العناصر فيها.
- تعتبر سريعة في قراءة البيانات وبطيئة في إدخال ومسح البيانات.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

**الدالة add(Object o):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة add(int index, Object element):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة في الترتيب المحدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة clear():**

تقوم بمسح جميع العناصر الموجودة.

**الدالة contains(Object elem):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في العناصر الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object elem: العنصر الذي نريد البحث عنه ضمن العناصر الموجودة.

**الدالة get(int index):**

تقوم بإرجاع قيمة العنصر الذي ترتيبه محدد في المعامل Parameter.



obeikan.com

**المعاملات Parameters:**

int index: ترتيب العنصر الذي نريد إرجاع قيمته.

**الدالة isEmpty ():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة remove(int index):**

تقوم بمسح العنصر الذي ترتيبه محدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الذي نريد مسحه من العناصر الموجودة.

**الدالة remove(Object o):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجوده فيها.

**المعاملات Parameters:**

Object o: العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجوده فيها.

**الدالة removeRange(int fromIndex, int toIndex):**

تقوم بمسح جميع العناصر بداية من العنصر الذي ترتيبه محدد في المعامل Parameter الأول إلي العنصر الذي يأتي قبل العنصر الذي ترتيبه محدد في المعامل Parameter الثاني.

**المعاملات Parameters:**

int fromIndex: ترتيب العنصر الذي نريد بدء المسح من عنده.

int toIndex: ترتيب العنصر الذي نريد الانتهاء من المسح قبله مباشرة.

**الدالة set(int index, Object element):**

تقوم بتعديل العنصر الموجود في الترتيب المحدد في المعامل Parameter لتصبح قيمته نفس القيمة المحددة في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الجديد الذي نريد تعديل قيمته.



obeikan.com

Object o: قيمة العنصر الجديدة.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة ArrayList.

**مثال (1): الفصيلة ArrayList:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة ArrayList لإضافة وعرض عناصرها.

**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyArrayList
4:  {
5:    public static void main(String[]args)
6:    {
7:        ArrayList list = new ArrayList();
8:
9:        list.add("Mostafa");
10:       list.add("Maged");
11:       list.add("Mahmoud");
12:       list.add("Fakhry");
13:
14:       for ( int i = 0 ; i < list.size() ; i ++ )
15:       {
16:           System.out.println("Element number " + i + " = "
    + list.get(i));
17:       }
18:   }
19: }
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyArrayList.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع ArrayList.
- في السطور من 9 إلي 12 يتم إضافة بعض العناصر الجديدة.
- في السطور من 14 إلي 17 يتم إنشاء دوارة Loop لعرض جميع عناصر المجموعة Collection.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-1).

**الشكل (14-1) الفصيلة ArrayList**

**الفصيلة Vector:**

- تنتمي إلي القائمة List.
- تعتبر فصيلة مفهرسة Ordered وليست مرتبة Unsorted.
- تعتبر فصيلة متزامنة Synchronized.
- تقبل القيم null فيها.
- تقبل تكرار العناصر فيها.
- تعتبر بطيئة في قراءة وإدخال ومسح البيانات.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

**الدالة add(Object o):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة add(int index, Object element):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة في الترتيب المحدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة clear():**

تقوم بمسح جميع العناصر الموجودة.

**الدالة contains(Object elem):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في العناصر الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object elem: العنصر الذي نريد البحث عنه ضمن العناصر الموجودة.

**الدالة firstElement():**

تقوم بإرجاع قيمة أول عنصر.

**الدالة get(int index):**

تقوم بإرجاع قيمة العنصر الذي ترتيبه محدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الذي نريد إرجاع قيمته.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة lastElement():**

تقوم بإرجاع قيمة أخر عنصر.



obeikan.com

**الدالة remove(int index):**

تقوم بمسح العنصر الذي ترتيبه محدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الذي نريد مسحه من العناصر الموجودة.

**الدالة remove(Object o):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجوده فيها.

**المعاملات Parameters:**

Object o: العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجوده فيها.

**الدالة removeRange(int fromIndex, int toIndex):**

تقوم بمسح جميع العناصر بداية من العنصر الذي ترتيبه محدد في المعامل Parameter الأول إلي العنصر الذي يأتي قبل العنصر الذي ترتيبه محدد في المعامل Parameter الثاني.

**المعاملات Parameters:**

int fromIndex: ترتيب العنصر الذي نريد بدء المسح من عنده.

int toIndex: ترتيب العنصر الذي نريد الانتهاء من المسح قبله مباشرة.

**الدالة set(int index, Object element):**

تقوم بتعديل العنصر الموجود في الترتيب المحدد في المعامل Parameter لتصبح قيمته نفس القيمة المحددة في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الجديد الذي نريد تعديل قيمته.

Object o: قيمة العنصر الجديدة.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة Vector.

**مثال (2): الفصيلة Vector:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة Vector لإضافة وعرض عناصرها.

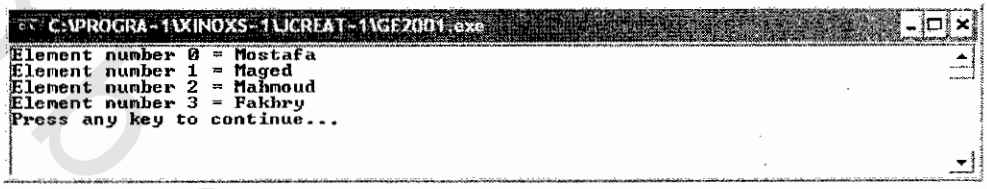
**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyVector
4:  {
5:    public static void main(String[]args)
6:    {
7:          Vector vector = new Vector();
8:
9:          vector.add("Mostafa");
10:         vector.add("Maged");
11:         vector.add("Mahmoud");
12:         vector.add("Fakhry");
13:
14:         for ( int i = 0 ; i < vector.size() ; i ++ )
15:         {
16:             System.out.println("Element number " + i + " = "
    + vector.get(i));
17:         }
18:  }
19:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyVector.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع Vector.
- في السطور من 9 إلي 12 يتم إضافة بعض العناصر الجديدة.

- في السطور من 14 إلي 17 يتم إنشاء دوارة Loop لعرض جميع عناصر المجموعة Collection.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-2).




**الشكل (14-2) الفصيلة Vector**

### الفصيلة LinkedList:

- تنتمي إلي القائمة List.
- تعتبر فصيلة مفهرسة Ordered وليست مرتبة Unsorted.
- تعتبر فصيلة غير متزامنة Unsynchronized.
- تقبل القيم null فيها.
- تقبل تكرار العناصر فيها.
- تعتبر بطيئة في قراءة البيانات وسريعة في إدخال ومسح البيانات.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

### الدالة add(Object o):

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

### الدالة add(int index, Object element):

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة في الترتيب المحدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة addFirst(Object o):**

تقوم بإضافة العنصر الجديد كأول عنصر في العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة addLast(Object o):**

تقوم بإضافة العنصر الجديد كأخر عنصر في العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة clear():**

تقوم بمسح جميع العناصر الموجودة.

**الدالة contains(Object elem):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في العناصر الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object elem: العنصر الذي نريد البحث عنه ضمن العناصر الموجودة.

**الدالة get(int index):**

تقوم بإرجاع قيمة العنصر الذي ترتيبه محدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الذي نريد إرجاع قيمته.

**الدالة getFirst():**

تقوم بإرجاع قيمة أول عنصر.

**الدالة getLast():**

تقوم بإرجاع قيمة أخر عنصر.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة lastElement():**

تقوم بإرجاع قيمة أخر عنصر.

**الدالة remove(int index):**

تقوم بمسح العنصر الذي ترتيبه محدد في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الذي نريد مسحه من العناصر الموجودة.

**الدالة remove(Object o):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجوده فيها.

**المعاملات Parameters:**

Object o: العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجوده فيها.

**الدالة removeFirst():**

تقوم بمسح قيمة أول عنصر.

**الدالة removeLast():**

تقوم بمسح قيمة أخر عنصر.

**الدالة set(int index, Object element):**

تقوم بتعديل العنصر الموجود في الترتيب المحدد في المعامل Parameter لتصبح قيمته نفس القيمة المحددة في المعامل Parameter.

**المعاملات Parameters:**

int index: ترتيب العنصر الجديد الذي نريد تعديل قيمته.

Object o: قيمة العنصر الجديدة.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة LinkedList.

**مثال (3): الفصيلة LinkedList:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة LinkedList لإضافة وعرض عناصرها.

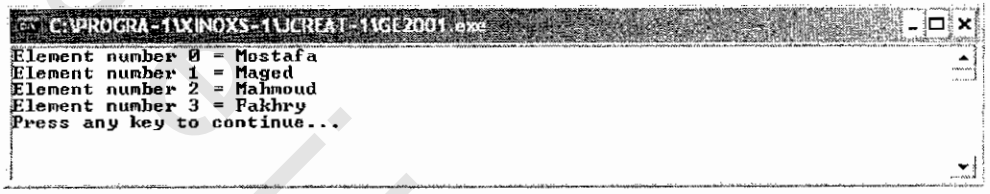
**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyLinkedList
4:  {
5:    public static void main(String[]args)
6:    {
7:         LinkedList list = new LinkedList();
8:
9:         list.add("Mostafa");
10:        list.add("Maged");
11:        list.add("Mahmoud");
12:        list.add("Fakhry");
13:
14:        for ( int i = 0 ; i < list.size() ; i ++ )
15:        {
16:             System.out.println("Element number " + i + " = "
    + list.get(i));
17:        }
18:  }
19:}
```

**ثالثاً: شرح الكود:**

في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.

- في السطر رقم 3 يتم إنشاء الفصيلة MyLinkedList.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع LinkedList.
- في السطور من 9 إلي 12 يتم إضافة بعض العناصر الجديدة.
- في السطور من 14 إلي 17 يتم إنشاء دوارة Loop لعرض جميع عناصر المجموعة Collection.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-3).




**الشكل (14-3) الفصيلة LinkedList**

### الفصيلة LinkedHashSet:

- تنتمي إلي المجموعة Set.
- تعتبر فصيلة مفهرسة Ordered وليست مرتبة Unsorted.
- تعتبر فصيلة غير متزامنة Unsynchronized.
- تقبل القيم null فيها.
- لا تقبل تكرار العناصر فيها.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

### الدالة add(Object o):

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

### الدالة clear():

تقوم بمسح جميع العناصر الموجودة.

**الدالة contains(Object elem):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في العناصر الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object elem: العنصر الذي نريد البحث عنه ضمن العناصر الموجودة.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة iterator():**

تقوم بإرجاع هدف object من نوع الفصيلة Iterator التي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection حيث تحتوي الفصيلة Iterator علي دالتين Methods تمكننا من عرض العناصر كما هو واضح في الجدول التالي:

| اسم الدالة Method | الاستخدام |
|---|---|
| next() | تقوم بإرجاع قيمة العنصر التالي في المجموعة Collection |
| hasNext() | تقوم بتحديد ما إذا كانت المجموعة Collection تحتوي علي عناصر أخري أم لا |

**الدالة remove(Object o):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجوده فيها.

**المعاملات Parameters:**

Object o: العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجوده فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة LinkedHashSet.

**مثال (4): الفصيلة LinkedHashSet:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة LinkedHashSet لإضافة وعرض عناصرها.

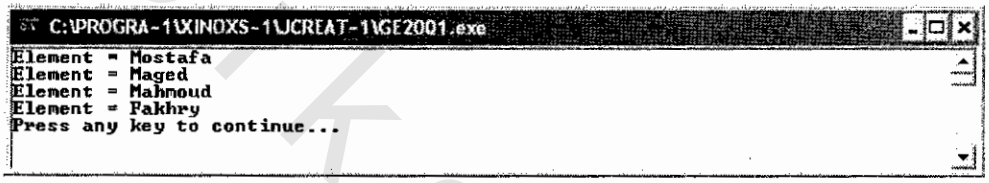
**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyLinkedHashSet
4:  {
5:    public static void main(String[]args)
6:    {
7:         LinkedHashSet set = new LinkedHashSet();
8:
9:         set.add("Mostafa");
10:        set.add("Maged");
11:        set.add("Mahmoud");
12:        set.add("Fakhry");
13:        set.add("Mostafa");
14:
15:        Iterator iterator = set.iterator();
16:
17:        while ( iterator.hasNext() )
18:        {
19:             System.out.println("Element = " + iterator.next());
20:        }
21:  }
22:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyLinkedHashSet.

- في السطر رقم 7 يتم إنشاء هدف Object من نوع LinkedHashSet.
- في السطور من 9 إلي 13 يتم إضافة بعض العناصر الجديدة.
- في السطر رقم 15 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة iterator() حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 17 إلي 20 يتم استخدام الدالة hasNext() و الدالة next() لعرض جميع عناصر المجموعة Collection. لاحظ أنه تم عرض القيمة Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-4).




**الشكل (14-4) الفصيلة LinkedHashSet**

### الفصيلة TreeSet:

- تنتمي إلي المجموعة Set.
- تعتبر فصيلة مفهرسة Ordered ومرتبة Sorted.
- تقبل القيم null فيها.
- لا تقبل تكرار العناصر فيها.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

### الدالة add(Object o):

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

### الدالة clear():

تقوم بمسح جميع العناصر الموجودة.

**الدالة contains(Object elem):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في العناصر الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object elem: العنصر الذي نريد البحث عنه ضمن العناصر الموجودة.

**الدالة first():**

تقوم بإرجاع قيمة أول عنصر في المجموعة المرتبة. تذكر أن الفصيلة TreeSet تقوم بترتيب العناصر المضافة إليها.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة iterator():**

تقوم بإرجاع هدف object من نوع الفصيلة Iterator والتي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection كما سبق لنا شرحه في هذا الفصل.

**الدالة last():**

تقوم بإرجاع قيمة أخر عنصر في المجموعة المرتبة. تذكر أن الفصيلة TreeSet تقوم بترتيب العناصر المضافة إليها.

**الدالة remove(Object o):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجوده فيها.

**المعاملات Parameters:**

Object o: العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجوده فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة TreeSet.

**مثال (5): الفصيلة TreeSet:**

**أولاً: هدف المثال:**

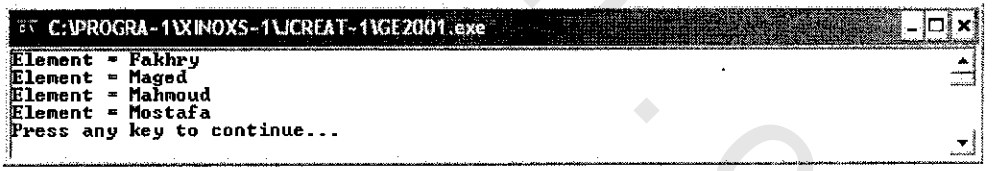
توضيح كيفية استخدام الفصيلة TreeSet لإضافة وعرض عناصرها.

**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyTreeSet
4:  {
5:    public static void main(String[]args)
6:    {
7:          TreeSet set = new TreeSet();
8:
9:          set.add("Mostafa");
10:         set.add("Maged");
11:         set.add("Mahmoud");
12:         set.add("Fakhry");
13:         set.add("Mostafa");
14:
15:         Iterator iterator = set.iterator();
16:
17:         while ( iterator.hasNext() )
18:         {
19:             System.out.println("Element = " + iterator.next());
20:         }
21:  }
22:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyTreeSet.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع TreeSet.
- في السطور من 9 إلي 13 يتم إضافة بعض العناصر الجديدة.
- في السطر رقم 15 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة iterator() حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 17 إلي 20 يتم استخدام الدالة hasNext() و الدالة next() لعرض جميع عناصر المجموعة Collection. لاحظ أنه تم عرض القيمة Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط. لاحظ أيضاً طباعة القيم مرتبة أبجدياً.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (5-14).




**الشكل (14-5) الفصيلة TreeSet**

**الفصيلة HashSet:**

- تنتمي إلي المجموعة Set.
- تعتبر فصيلة غير مفهرسة Unordered وليست مرتبة Unsorted.
- تقبل القيم null فيها.
- لا تقبل تكرار العناصر فيها.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

**الدالة add(Object o):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object o: العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة clear():**

تقوم بمسح جميع العناصر الموجودة.

**الدالة contains(Object elem):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في العناصر الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object elem: العنصر الذي نريد البحث عنه ضمن العناصر الموجودة.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة iterator():**

تقوم بإرجاع هدف object من نوع الفصيلة Iterator والتي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection كما سبق لنا شرحه في هذا الفصل.

**الدالة remove(Object o):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجوده فيها.

**المعاملات Parameters:**

Object o: العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجوده فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة HashSet.

**مثال (6): الفصيلة HashSet:**

**أولاً: هدف المثال:**

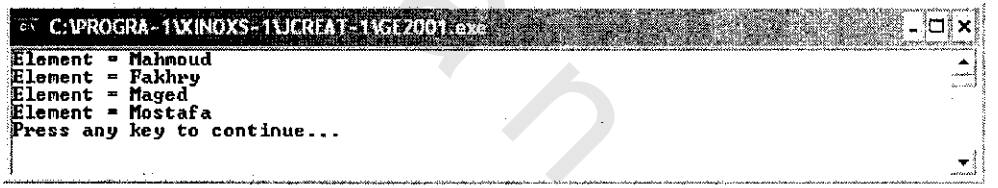
توضيح كيفية استخدام الفصيلة HashSet لإضافة وعرض عناصرها.

**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyHashSet
4:  {
5:    public static void main(String[]args)
6:    {
7:          HashSet set = new HashSet();
8:
9:          set.add("Mostafa");
10:         set.add("Maged");
11:         set.add("Mahmoud");
12:         set.add("Fakhry");
13:         set.add("Mostafa");
14:
15:         Iterator iterator = set.iterator();
16:
17:         while ( iterator.hasNext() )
18:         {
19:             System.out.println("Element = " + iterator.next());
20:         }
21:  }
22:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyHashSet.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع HashSet.
- في السطور من 9 إلي 13 يتم إضافة بعض العناصر الجديدة.
- في السطر رقم 15 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة iterator() حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 17 إلي 20 يتم استخدام الدالة hasNext() و الدالة next() لعرض جميع عناصر المجموعة Collection. لاحظ أنه تم عرض القيمة Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط. لاحظ أيضاً طباعة القيم بدون ترتيب معين.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-6).




**الشكل (14-6) الفصيلة HashSet**

**الفصيلة TreeMap:**

- تنتمي إلي الخريطة Map.
- تعتبر فصيلة مفهرسة Ordered ومرتبة Sorted.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

**الدالة clear():**

تقوم بمسح جميع العناصر الموجودة.

**الدالة containsKey(Object key):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في المفاتيح Keys الموجودة

وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة containsValue(Object value):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في القيم Values الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object value: العنصر الذي نريد البحث عنه ضمن القيم Values الموجودة.

**الدالة firstKey():**

تقوم بإرجاع قيمة أول مفتاح key.

**الدالة get(Object key):**

تقوم بإرجاع القيمة Value المناظرة للمفتاح key الموجود في المعامل Parameter.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة keySet():**

تقوم بإرجاع هدف object من نوع المجموعة Set والتي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection عن طريق استدعاء الدالة iterator() ثم استخدام دوال الفصيلة Iterator كما سبق لنا شرحه في هذا الفصل.

**الدالة lastKey():**

تقوم بإرجاع قيمة آخر مفتاح key.

**الدالة put(Object key, Object value):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object value: قيمة value العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة remove(Object key):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجود المفتاح key الموجود في المعامل في هذه العناصر.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجود المفتاح key فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة TreeMap.

**مثال (7): الفصيلة TreeMap:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة TreeMap لإضافة وعرض عناصرها.

**ثانياً: كود البرمجة:**

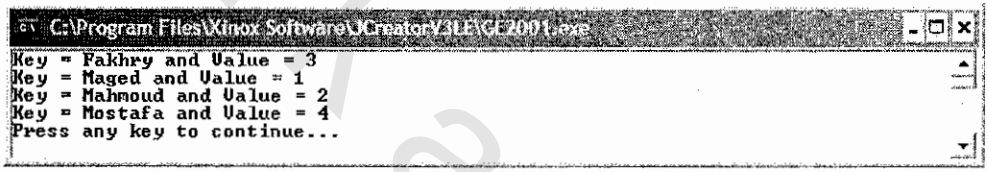
```
1:  import java.util.*;
2:
3:  public class MyTreeMap
4:  {
```

```
5:  public static void main(String[]args)
6:  {
7:      TreeMap map = new TreeMap();
8:
9:      int x = 0;
10:
11:     map.put("Mostafa",new Integer(x++));
12:     map.put("Maged",new Integer(x++));
13:     map.put("Mahmoud",new Integer(x++));
14:     map.put("Fakhry",new Integer(x++));
15:     map.put("Mostafa",new Integer(x++));
16:
17:     Iterator iterator = map.keySet().iterator();
18:
19:     while ( iterator.hasNext() )
20:     {
21:         String key = (String)iterator.next();
22:         Integer value = (Integer)map.get(key);
23:         System.out.println("Key = " + key + " and
    Value = " + value);
24:     }
25: }
26:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyTreeMap.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع TreeMap.
- في السطر رقم 9 يتم إنشاء متغير من نوع int بحيث تكون قيمته الابتدائية Initial Value تساوي صفراً.
- في السطور من 11 إلي 15 يتم إضافة بعض العناصر الجديدة.

- في السطر رقم 17 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة keySet() ثم iterator() حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 19 إلي 24 يتم استخدام الدالة hasNext() و الدالة next() والدالة get() لعرض جميع عناصر المجموعة Collection حيث يتم عرض جميع قيم المفاتيح keys والقيم values. لاحظ أنه تم عرض المفتاح Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط. لاحظ أيضاً أن القيمة الموجودة للمفتاح Mostafa هي أخر قيمة تم تحديدها. لاحظ أيضاً طباعة القيم مرتبة ترتيباً أبجدياً.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (7-14).




الشكل (14-7) الفصيلة TreeMap

### الفصيلة LinkedHashMap:

- تنتمي إلي الخريطة Map.
- تعتبر فصيلة مفهرسة Ordered وليست مرتبة Unsorted.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

### الدالة clear():

تقوم بمسح جميع العناصر الموجودة.

### الدالة containsKey(Object key):

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في المفاتيح Keys الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة containsValue(Object value):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في القيم Values الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object value: العنصر الذي نريد البحث عنه ضمن القيم Values الموجودة.

**الدالة get(Object key):**

تقوم بإرجاع القيمة Value المناظرة للمفتاح key الموجود في المعامل Parameter.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة keySet():**

تقوم بإرجاع هدف object من نوع المجموعة Set والتي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection عن طريق استدعاء الدالة iterator() ثم استخدام دوال الفصيلة Iterator كما سبق لنا شرحه في هذا الفصل.

**الدالة put(Object key, Object value):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object value: قيمة value العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة remove(Object key):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجود المفتاح key الموجود في المعامل في هذه العناصر.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجود المفتاح key فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة LinkedHashMap.

**مثال (8): الفصيلة LinkedHashMap:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة LinkedHashMap لإضافة وعرض عناصرها.

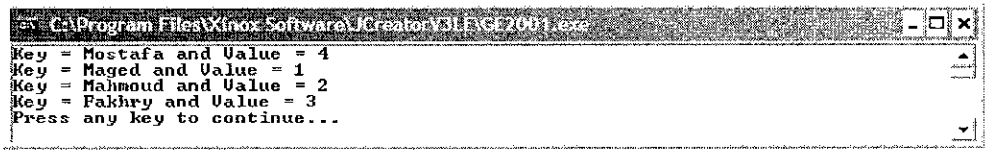
**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyLinkedHashMap
4:  {
5:   public static void main(String[]args)
6:   {
7:        LinkedHashMap map = new LinkedHashMap();
8:
9:        int x = 0;
10:
11:       map.put("Mostafa",new Integer(x++));
12:       map.put("Maged",new Integer(x++));
13:       map.put("Mahmoud",new Integer(x++));
14:       map.put("Fakhry",new Integer(x++));
```

```
15:       map.put("Mostafa",new Integer(x++));
16:
17:       Iterator iterator = map.keySet().iterator();
18:
19:       while ( iterator.hasNext() )
20:       {
21:            String key = (String)iterator.next();
22:            Integer value = (Integer)map.get(key);
23:            System.out.println("Key = " + key + " and
    Value = " + value);
24:       }
25:  }
26:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyLinkedHashMap.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع LinkedHashMap.
- في السطر رقم 9 يتم إنشاء متغير من نوع int بحيث تكون قيمته الابتدائية Initial Value تساوي صفراً.
- في السطور من 11 إلي 15 يتم إضافة بعض العناصر الجديدة.
- في السطر رقم 17 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة keySet() ثم iterator() حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 19 إلي 24 يتم استخدام الدالة hasNext() و الدالة next() والدالة get() لعرض جميع عناصر المجموعة Collection حيث يتم عرض جميع قيم المفاتيح keys والقيم values. لاحظ أنه تم عرض المفتاح Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط. لاحظ أيضاً أن القيمة الموجودة للمفتاح Mostafa هي أخر قيمة تم تحديدها.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (14-8).




**الشكل (14-8) الفصيلة LinkedHashMap**

### الفصيلة HashMap:

- تنتمي إلي الخريطة Map.
- تعتبر فصيلة غير مفهرسة Unordered وليست مرتبة Unsorted.
- تقبل قيمة null واحدة فقط في المفتاح Key وعدة قيم null في القيمة Value.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

### الدالة clear():

تقوم بمسح جميع العناصر الموجودة.

### الدالة containsKey(Object key):

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في المفاتيح Keys الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

### الدالة containsValue(Object value):

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في القيم Values الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object value: العنصر الذي نريد البحث عنه ضمن القيم Values الموجودة.

**الدالة get(Object key):**

تقوم بإرجاع القيمة Value المناظرة للمفتاح key الموجود في المعامل Parameter.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة keySet():**

تقوم بإرجاع هدف object من نوع المجموعة Set والتي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection عن طريق استدعاء الدالة iterator() ثم استخدام دوال الفصيلة Iterator كما سبق لنا شرحه في هذا الفصل.

**الدالة put(Object key, Object value):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object value: قيمة value العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة remove(Object key):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجود المفتاح key الموجود في المعامل في هذه العناصر.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجود المفتاح key فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة HashMap.

**مثال (9): الفصيلة HashMap:**

**أولاً: هدف المثال:**

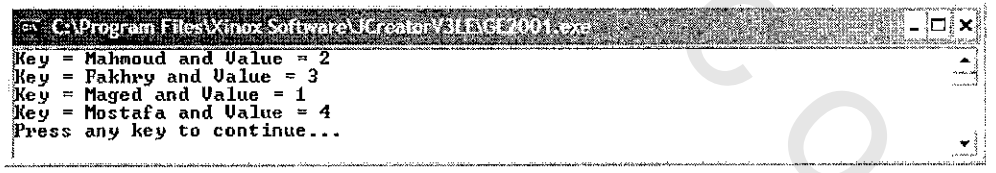
توضيح كيفية استخدام الفصيلة HashMap لإضافة وعرض عناصرها.

**ثانياً: كود البرمجة:**

```
1: import java.util.*;
2:
3: public class MyHashMap
4: {
5:   public static void main(String[]args)
6:   {
7:        HashMap map = new HashMap();
8:
9:        int x = 0;
10:
11:       map.put("Mostafa",new Integer(x++));
12:       map.put("Maged",new Integer(x++));
13:       map.put("Mahmoud",new Integer(x++));
14:       map.put("Fakhry",new Integer(x++));
15:       map.put("Mostafa",new Integer(x++));
16:
17:       Iterator iterator = map.keySet().iterator();
18:
19:       while ( iterator.hasNext() )
20:       {
21:            String key = (String)iterator.next();
22:            Integer value = (Integer)map.get(key);
23:            System.out.println("Key = " + key + " and
   Value = " + value);
24:       }
25:  }
26:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyHashMap.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع HashMap.
- في السطر رقم 9 يتم إنشاء متغير من نوع int بحيث تكون قيمته الابتدائية Initial Value تساوي صفراً.
- في السطور من 11 إلي 15 يتم إضافة بعض العناصر الجديدة.
- في السطر رقم 17 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة keySet() ثم iterator()حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 19 إلي 24 يتم استخدام الدالة hasNext() و الدالة next() والدالة get() لعرض جميع عناصر المجموعة Collection حيث يتم عرض جميع قيم المفاتيح keys والقيم values. لاحظ أنه تم عرض المفتاح Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط. لاحظ أيضاً أن القيم الموجودة للمفتاح Mostafa هي أخر قيمة تم تحديدها. لاحظ أيضاً عدم وجود ترتيب معين للعناصر.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (9-14).




**الشكل (14-9) الفصيلة HashMap**

## الفصيلة Hashtable:

- تنتمي إلي الخريطة Map.
- تعتبر فصيلة غير مفهرسة Unordered وليست مرتبة Unsorted.
- لا تقبل القيم null فيها لا في المفتاح Key ولا في القيمة Value.

تتوافر العديد من الدوال Methods لهذه الفصيلة class كما يلي.

**الدالة clear():**

تقوم بمسح جميع العناصر الموجودة.

**الدالة containsKey(Object key):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في المفاتيح Keys الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة containsValue(Object value):**

تقوم بالبحث عن العنصر الموجود في المعامل Parameter في القيم Values الموجودة وتقوم بإرجاع القيمة true في حالة وجوده في العناصر الموجودة وتقوم بإرجاع القيمة false في حالة عدم وجوده في العناصر الموجودة.

**المعاملات Parameters:**

Object value: العنصر الذي نريد البحث عنه ضمن القيم Values الموجودة.

**الدالة get(Object key):**

تقوم بإرجاع القيمة Value المناظرة للمفتاح key الموجود في المعامل Parameter.

**المعاملات Parameters:**

Object key: العنصر الذي نريد البحث عنه ضمن المفاتيح Keys الموجودة.

**الدالة isEmpty():**

تقوم بإرجاع القيمة true في حالة عدم وجود أي عناصر موجودة في المجموعة Collection (أي أن المجموعة Collection خالية من أي عناصر) وتقوم بإرجاع القيمة false في حالة وجود أي عناصر.

**الدالة keySet():**

تقوم بإرجاع هدف object من نوع المجموعة Set والتي تمكننا من عرض جميع العناصر الموجودة في المجموعة Collection عن طريق استدعاء الدالة iterator() ثم استخدام دوال الفصيلة Iterator كما سبق لنا شرحه في هذا الفصل.

**الدالة put(Object key, Object value):**

تقوم بإضافة العنصر الجديد إلي العناصر الموجودة.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

Object value: قيمة value العنصر الجديد الذي ستتم إضافته إلي العناصر الموجودة.

**الدالة remove(Object key):**

تقوم بمسح العنصر من العناصر الموجودة في حالة وجود المفتاح key الموجود في المعامل في هذه العناصر.

**المعاملات Parameters:**

Object key: مفتاح key العنصر الذي سيتم مسحه من العناصر الموجودة في حالة وجود المفتاح key فيها.

**الدالة size():**

تقوم بإرجاع عدد العناصر الموجودة في المجموعة Collection.

المثال التالى يوضح الفصيلة Hashtable.

**مثال (10): الفصيلة Hashtable:**

**أولاً: هدف المثال:**

توضيح كيفية استخدام الفصيلة Hashtable لإضافة وعرض عناصرها.

**ثانياً: كود البرمجة:**

```
1:  import java.util.*;
2:
3:  public class MyHashTable
4:  {
5:    public static void main(String[]args)
6:    {
7:        Hashtable table = new Hashtable();
8:
9:        int x = 0;
10:
11:       table.put("Mostafa",new Integer(x++));
12:       table.put("Maged",new Integer(x++));
13:       table.put("Mahmoud",new Integer(x++));
14:       table.put("Fakhry",new Integer(x++));
15:       table.put("Mostafa",new Integer(x++));
16:
17:       Iterator iterator = table.keySet().iterator();
18:
19:       while ( iterator.hasNext() )
20:       {
21:           String key = (String)iterator.next();
22:           Integer value = (Integer)table.get(key);
23:           System.out.println("Key = " + key + " and
    Value = " + value);
24:       }
25:  }
26:}
```

**ثالثاً: شرح الكود:**

- في السطر رقم 1 يتم تضمين الحزم packages اللازمة لعمل البرنامج.
- في السطر رقم 3 يتم إنشاء الفصيلة MyHashTable.
- في السطر رقم 7 يتم إنشاء هدف Object من نوع Hashtable.

- في السطر رقم 9 يتم إنشاء متغير من نوع int بحيث تكون قيمته الابتدائية Initial Value تساوي صفراً.
- في السطور من 11 إلي 15 يتم إضافة بعض العناصر الجديدة.
- في السطر رقم 17 يتم إنشاء هدف Object من نوع Iterator باستخدام الدالة keySet() ثم iterator() حتي يمكننا عرض عناصر المجموعة Collection.
- في السطور من 19 إلي 24 يتم استخدام الدالة hasNext() و الدالة next() والدالة get() لعرض جميع عناصر المجموعة Collection حيث يتم عرض جميع قيم المفاتيح keys والقيم values. لاحظ أنه تم عرض المفتاح Mostafa مرة واحدة فقط دلالة علي عرض القيم الفريدة Unique فقط. لاحظ أيضاً أن القيم الموجودة للمفتاح Mostafa هي أخر قيمة تم تحديدها. لاحظ أيضاً عدم وجود ترتيب معين للعناصر.
- قم بترجمة البرنامج وتنفيذه لتظهر لك شاشة البرنامج كما هو واضح في الشكل (10-14).

**الشكل (14-10) الفصيلة Hashtable**

## ملخص الفصل:

- تعلمنا في هذا الفصل كيفية استخدام العديد من فصائل المجموعات Collections حيث تعرفنا علي القائمة List والمجموعة Set والخريطة Map.
- في الفصل القادم نتعلم - بإذن الله - كيفية تصميم واجهة المستخدم الرسومية Graphical User Interface (GUI) من خلال أدوات مكتبة Swing ، فتابع معنا الفصل القادم.