表项单元的数组，它们也在图 5-14 中表出。HashEntry引用数组的每一项是下列 3 种情形之一：

```
1  public class QuadraticProbingHashTable<AnyType>
2  {
3      public QuadraticProbingHashTable( )
4        { /* Figure 5.15 */ }
5      public QuadraticProbingHashTable( int size )
6        { /* Figure 5.15 */ }
7      public void makeEmpty( )
8        { /* Figure 5.15 */ }
9
10     public boolean contains( AnyType x )
11       { /* Figure 5.16 */ }
12     public void insert( AnyType x )
13       { /* Figure 5.17 */ }
14     public void remove( AnyType x )
15       { /* Figure 5.17 */ }
16
17     private static class HashEntry<AnyType>
18     {
19         public AnyType  element;  // the element
20         public boolean isActive;  // false if marked deleted
21
22         public HashEntry( AnyType e )
23           { this( e, true ); }
24
25         public HashEntry( AnyType e, boolean i )
26           { element  = e; isActive = i; }
27     }
28
29     private static final int DEFAULT_TABLE_SIZE = 11;
30
31     private HashEntry<AnyType> [ ] array;  //  The array of elements
32     private int currentSize;               //  The number of occupied cells
33
34     private void allocateArray( int arraySize )
35       { /* Figure 5.15 */ }
36     private boolean isActive( int currentPos )
37       { /* Figure 5.16 */ }
38     private int findPos( AnyType x )
39       { /* Figure 5.16 */ }
40     private void rehash( )
41       { /* Figure 5.22 */ }
42
43     private int myhash( AnyType x )
44       { /* See online code */ }
45     private static int nextPrime( int n )
46       { /* See online code */ }
47     private static boolean isPrime( int n )
48       { /* See online code */ }
49 }
```

图 5-14　使用探测方法的散列表的类架构，包括嵌套的 HashEntry 类