

Answers to End of Section and Review Exercises for Chapter 5

Exercises 5.1

1. The items in a bag are unordered.
2. The `isEmpty`, `len`, `str`, `iter`, `add`, `==`, and `+` operations appear in the interface of any collection.
3. The `__init__` method is responsible for creating a collection object.
4. Interfaces are separated from implementations because
 - a. The code for an implementation can be modified without rewriting the code that uses it.
 - b. A completely different implementation can be chosen, as long as it conforms to the interface.
 - c. The user of an interface does not have to be concerned with any details of its implementation.

Exercises 5.2

1. The `__init__` method of a collection class is responsible for creating an instance of a container object, setting the logical size to 0, and copying data items from a source collection to the new collection.
2. It's better to call methods than to refer to instance variables in a class, because methods are part of an interface, which never changes, whereas variables are not, so they may change. Thus, the use of method calls rather than variable references makes code more stable and easier to share.
3. Here is the code for the revised `__init__` method:

```
def __init__(self, sourceCollection = None):
    self.clear()
    if sourceCollection:
        for item in sourceCollection: self.add(item)
```
4. The `__iter__` method might be the most useful method in a collection class, because many other methods in that class use the `for` loop on that collection.
5. When Python does not see a `__contains__` method in a collection class, it automatically creates one, using the `__iter__` method to perform a linear search on the collection.

Exercises 5.3

1. The array bag uses 10 cells of memory for the array and 1 cell of memory for each of the two instance variables. The linked bag uses 1 cell of memory for each of the two instance variables.
2. A linked bag could do without a separate instance variable to track its logical size, because the number of items always equals the number of nodes in the linked structure. However, the `len` function to count these nodes would be linear in running time. Therefore, a separate variable to track this value is more desirable.
3. Because each node removed is automatically recycled to system memory, the removal of a given item leaves no memory unused in a linked bag.

Answers to Review Questions

1. b
2. a
3. b
4. b
5. b
6. a
7. a