

Past-paper walk-through

ArrayList Traversals ■ 4.9

eXercise

Questions in this set

- 2024 Q3
- 2023 Q3
- 2019 Q3
- 2018 Q2
- 2017 Q1
- 2016 Q4
- 2015 Q3
- 2015 Q4
- 2014 Q1
- 2014 Q3

Reading the mark scheme

- **B1** a mark that stands on its own – the point is either made or not.
- **M1** a *method* mark: the right approach, even if the number is wrong.
- **A1** an *answer* mark, and it usually needs its M mark first.
- **C1** a working mark on the way to the answer.
- **//** separates answers the examiner accepts equally.
- **ecf** = error carried forward: a later part is marked on your own earlier answer.
- **owtte** = “or words to that effect” – the wording need not match.

Question 3

2024 ■ Q3

This question involves the manipulation and analysis of a list of words. The following `WordChecker` class contains an `ArrayList<String>` to be analyzed and methods that are used to perform the analysis. You will write two methods of the `WordChecker` class.

```

k+kdpublic+w k+kdclass n+ncWordChecker
p
+w      c+cm/** Initialized in the constructor and contains no null elements */
+w      k+kdprivate+w nArrayListonStringo+w nwordListp;

+w      c+cm/**
c+cm      * Returns true if each element of wordList (except the first) contains the previous
c+cm      * element as a substring and returns false otherwise, as described in part (a)
c+cm      * Precondition: wordList contains at least two elements.
c+cm      * Postcondition: wordList is unchanged.
c+cm      */
+w      k+kdpublic+w k+ktboolean+w n+nfisWordChainp(p)
+w      p+w      c+cm/* to be implemented in part (a) */+w      p
+w      c+cm/**
c+cm      * Returns an ArrayListString based on strings from wordList that start
c+cm      * with target, as described in part (b). Each element of the returned ArrayList has had

```

Question 3

(a) Write the `isWordChain` method, which determines whether each element of `wordList` (except the first) contains the previous element as a substring. The following table shows two sample `isWordChain` method calls.

wordList	isWordChain Return Value	Explanation
<code>["an", "band", "band", "abandon"]</code>	<code>true</code>	Each element contains the previous element as a substring.
<code>["to", "too", "stool", "tools"]</code>	<code>false</code>	"tools" does not contain the substring "stool".

Question 3

Complete the `isWordChain` method.

```
c+cm/**
```

```
c+cm * Returns true if each element of wordList (except the first) contains the previous
```

```
c+cm * element as a substring and returns false otherwise, as described in part (a).
```

```
c+cm * Precondition: wordList contains at least two elements.
```

```
c+cm * Postcondition: wordList is unchanged.
```

```
c+cm */
```

```
k+kdpublic+w k+ktboolean+w n+nfisWordChainp(p)
```

Question 3

2024 ■ Q3

This question involves the manipulation and analysis of a list of words. The following `WordChecker` class contains an `ArrayList<String>` to be analyzed and methods that are used to perform the analysis. You will write two methods of the `WordChecker` class.

```
k+kdpublic+w k+kdclass n+ncWordChecker
p
+w      c+cm/** Initialized in the constructor and contains no null elements */
+w      k+kdprivate+w nArrayListonStringo+w nwordListp;

+w      c+cm/**
c+cm      * Returns true if each element of wordList (except the first) contains the previous
c+cm      * element as a substring and returns false otherwise, as described in part (a)
c+cm      * Precondition: wordList contains at least two elements.
c+cm      * Postcondition: wordList is unchanged.
c+cm      */
+w      k+kdpublic+w k+ktboolean+w n+nfisWordChainp(p)
+w      p+w      c+cm/* to be implemented in part (a) */+w      p

+w      c+cm/**
c+cm      * Returns an ArrayListString based on strings from wordList that start
```

Question 3

(b) Write the `createList` method, which creates and returns an `ArrayList<String>`. The method identifies strings in `wordList` that start with `target` and returns a new `ArrayList` containing each identified string without the starting occurrence of `target`. Elements must appear in the returned list in the same order as they appear in `wordList`. Consider an example where `wordList` contains the following strings.

```
["catch", "bobcat", "catchacat", "cat", "at"]
```


Question 3

The following table shows the `ArrayList` returned by some calls to `createList`. In all cases, `wordList` is unchanged.

Method Call	<code>ArrayList</code> Returned by <code>createList</code>	Explanation
<code>createList("cat")</code>	<code>["ch", "chacat", ""]</code>	Only "catch", "catchacat", and "cat" begin with "cat".
<code>createList("catch")</code>	<code>["", "acat"]</code>	Only "catch" and "catchacat" begin with "catch".
<code>createList("dog")</code>	<code>[]</code>	None of the words in <code>wordList</code> begin with "dog".

Complete the `createList` method.

Question 3

```
c+cm/**
```

```
c+cm * Returns an ArrayListString based on strings from wordList that start
c+cm * with target, as described in part (b). Each element of the returned ArrayL
c+cm * the initial occurrence of target removed.
```

```
c+cm * Postconditions: wordList is unchanged.
```

```
c+cm *      Items appear in the returned list in the same order as they appear in
```

```
c+cm */
```

```
k+kdpublish+w nArrayListonStringo+w n+nfcreateListp(nString+w ntargetp)
```

Question 3

Class information for this question

```
k+kdpublic+w k+kdclass n+ncWordChecker
```

```
k+kdprivate+w nArrayListonStringo+w nwordList
```

```
k+kdpublic+w k+ktboolean+w n+nfisWordChainp(p)
```

```
k+kdpublic+w nArrayListonStringo+w n+nfcreateListp(nString+w ntargetp)
```

Question 3

2023 ■ Q3

This question involves the analysis of weather data. The following `WeatherData` class has an instance variable, `temperatures`, which contains the daily high temperatures recorded on consecutive days at a particular location. The class also contains methods used to analyze that data. You will write two methods of the `WeatherData` class.

```

k+kdpublic+w k+kdclass n+ncWeatherData
p
+w      c+cm/** Guaranteed not to be null and to contain only nonnull entries */
+w      k+kdprivate+w nArrayListonDoubleo+w ntemperaturesp;

+w      c+cm/**
c+cm      * Cleans the data by removing from temperatures all values that are less than
c+cm      * lower and all values that are greater than upper, as described in part (a)
c+cm      */
+w      k+kdpublic+w k+ktvoid+w n+nfcleanDatap(k+ktdouble+w nlowerp,+w k+ktdouble+w nupperp)
+w      p+w      c+cm/* to be implemented in part (a) */+w      p
+w      c+cm/**
c+cm      * Returns the length of the longest heat wave found in temperatures, as described in
c+cm      * part (b)
c+cm      * Precondition: There is at least one heat wave in temperatures based on threshold.

```

Question 3

(a) Write the `cleanData` method, which modifies the `temperatures` instance variable by removing all values that are less than the `lower` parameter and all values that are greater than the `upper` parameter. The order of the remaining values in `temperatures` must be maintained.

For example, consider a `WeatherData` object for which `temperatures` contains the following.

```
99.1 | 142.0 | 85.0 | 85.1 | 84.6 | 94.3 | 124.9 | 98.0 | 101.0 |
102.5
```

The three shaded values shown would be removed by the method call `cleanData(85.0, 120.0)`.

[Table of the same 10 values with three cells shaded gray: 142.0, 84.6, and 124.9]

```
99.1 | 142.0 | 85.0 | 85.1 | 84.6 | 94.3 | 124.9 | 98.0 | 101.0 |
102.5
```

The following shows the contents of `temperatures` after the three shaded values are

Question 3

99.1 | 85.0 | 85.1 | 94.3 | 98.0 | 101.0 | 102.5

Complete method `cleanData`.

```
c+cm/**
```

```
c+cm * Cleans the data by removing from temperatures all values that are less than
```

```
c+cm * lower and all values that are greater than upper, as described in part (a)
```

```
c+cm */
```

```
k+kdpublic+w k+ktvoid+w n+nfcleanDatap(k+ktdouble+w nlowerp,+w k+ktdouble+w nupper
```

Question 3

2023 ■ Q3

This question involves the analysis of weather data. The following `WeatherData` class has an instance variable, `temperatures`, which contains the daily high temperatures recorded on consecutive days at a particular location. The class also contains methods used to analyze that data. You will write two methods of the `WeatherData` class.

```
k+kdpublic+w k+kdclass n+ncWeatherData
p
+w    c+cm/** Guaranteed not to be null and to contain only nonnull entries */
+w    k+kdprivate+w nArrayListonDoubleo+w ntemperaturesp;

+w    c+cm/**
c+cm    * Cleans the data by removing from temperatures all values that are less than
c+cm    * lower and all values that are greater than upper, as described in part (a)
c+cm    */
+w    k+kdpublic+w k+ktvoid+w n+nfcleanDatap(k+ktdouble+w nlowerp,+w k+ktdouble+w nupperp)
+w    p+w    c+cm/* to be implemented in part (a) */+w    p

+w    c+cm/**
c+cm    * Returns the length of the longest heat wave found in temperatures, as described in
c+cm    * part (b)
+w    */
+w    k+ktint+w ngetLongestHeatWaveLengthp()
+w    p
```