

40009 ExerciseTypes.PPT12

Kotlin Search Engine

Submitters

anb122

Adithya Narayanan

Emarking

Final Tests TestSummary.txt: 1/1 Adithya Narayanan - anb122:j1:24

```
1: Final Tests: Summary for anb122 of j1
2: PPT 24
3: -----
4:
5:   Public Tests:
6:     Compiles:    1 / 1
7:     Tests Pass:  1 / 1
8:     Style Check: 1 / 1 ✓
9:
10: Git Repo: git@gitlab.doc.ic.ac.uk:lab2324_autumn/kotlinsearchengine_anb122.git
11: Commit ID: 9004c
```

Tests: 3/4
Quality: 0/3
Correctness: 2/3

5/10

- 1 for no tests added.
- 2 for incorrectly using maps as for loops.
 - 1 for no comments explaining code.
- 1 for no rank.

Final Tests

JsoupExamples.kt: 1/1 Adithya Narayanan - anb122:j1:24

```

1: package websearch
2:
3: import org.jsoup.Jsoup.connect
4: import org.jsoup.Jsoup.parse
5: import org.jsoup.nodes.Document
6: import org.jsoup.select.Elements
7:
8: fun main() {
9:
10:     val htmlString =
11:         """<html>
12:             <head>
13:                 <title>A Simple HTML Document</title>
14:             </head>
15:             <body>
16:                 <p>This is a very simple <a href="https://
17: //en.wikipedia.org/wiki/HTML">HTML</a> document.</p>
18:                 <p>It only has two paragraphs.</p>
19:             </body>
20:         </html>"""
21:
22:     // parse a string into a Document using Jsoup.parse()
23:     val htmlDocument: Document = parse(htmlString)
24:
25:     // get all the text from the document
26:     println(htmlDocument.text())
27:
28:     // extract all the <a >...</a> tags (which create links)
29:     val aTags: Elements = htmlDocument.getElementsByTag("a")
30:     println(aTags)
31:
32:     // get the value of the href attribute (where the link points to)
33:     val firstTag = aTags.first()
34:
35:     // aTags.first() could return null, so we have to check that before using it.
36:     if (firstTag != null) {
37:         val link: String = firstTag.attr("href")
38:         println(link)
39:     }
40:
41:     // A different approach to dealing with the nullable value is to use the "
42:     // safe-call" operator ?.
43:     // This says, if firstTag is null, don't call .attr(), just return null.
44:     println(firstTag?.attr("href"))
45:
46:     // You can also use the "Elvis" operator ?: to use a specific value if the
47:     // safe-call returns null
48:     println(firstTag?.attr("href") ?: "")
49:
50:     // do the same thing using a CSS selector
51:     val selection: Elements = htmlDocument.select("a[href]")
52:     val firstResult = selection.first()
53:
54:     // download a document from the internet using Jsoup.connect()
55:     val downloadedDocument = connect("https://kotlinlang.org/").get()
56:     println(downloadedDocument.body())
57: }

```

Final Tests

SearchEngine.kt: 1/2 Adithya Narayanan - anb122:j1:24

```

1: package websearch
2:
3: import org.jsoup.Jsoup
4:
5: class SearchEngine(private val mapURLToWebPage: Map<URL, WebPage>) {
6:
7:     private var index = emptyMap<String, List<SearchResult>>()
8:     fun getIndex(): Map<String, List<SearchResult>> {
9:         return index
10:     }
11:
12:     // I have created the function without the need for the helper rank().
13:     // Please let me know if this is ok? it's better to have a separate rank function.
14:     // I have also purposefully used only maps as I am trying to practice using
15:     // Haskell esque HOFs. Only do this in Haskell, it's very unreadable in Kotlin.
16:     fun compileIndex() {
17:         var URLToStrings = LinkedHashMap<URL, List<String>>()
18:         for (i in mapURLToWebPage) {
19:             URLToStrings.put(i.key, i.value.extractWords())
20:         }
21:         var URLToStringsAndCounts = URLToStrings.mapValues { it.value.map { x ->
22: Pair(x, it.key) } }
23:         .mapValues { it.value.groupingBy { x -> x.first }.eachCount() }
24:         var indexMutableMap = mutableMapOf<String, ArrayList<SearchResult>>()
25:         URLToStringsAndCounts
26:             .map { (url, stringcountpair) ->
27:                 stringcountpair
28:                     .map { (string, count) ->
29:                         if (string in indexMutableMap.keys) {
30:                             var searchResultList = indexMutableMap[string]
31:                             searchResultList?.add(SearchResult(url, count))
32:                             indexMutableMap.set(string, searchResultList)
33:                         } else {
34:                             indexMutableMap.put(string, arrayListOf(SearchResult(url, count)))
35:                         }
36:                     }
37:             }
38:         var indexMutableMapHolder = indexMutableMap.mapValues { x ->
39: x.value?.sortedByDescending { y -> y.numRefs } }
40:         index = indexMutableMapHolder.mapValues { y -> y.value?.toList() }.toMap()
41:     }
42:
43:     fun searchFor(query: String): SearchResultsSummary = SearchResultsSummary(
44:         query,
45:         if (this.getIndex().getValue(query) != null) {
46:             this.getIndex().getValue(query)
47:         } else {
48:             emptyList()
49:         }
50:     )
51:
52:     class SearchResult(val url: URL, val numRefs: Int)
53:
54:     class SearchResultsSummary(val query: String, val results: List<SearchResult>) {
55:         override fun toString(): String {
56:             var stringOut = "Results for $query:\n"
57:             results.map { stringOut += "${it.url} - ${it.numRefs} references\n" }
58:             return stringOut
59:         }
60:     }
61:
62:     fun main() {
63:         var list = mutableListOf(1, 1, 2, 2, 3, 4, 5, 4, 5, 6, 7, 8, 9)
64:         var arrlist = mutableListOf<Pair<Int, String>>()

```

no pointer
of int

Just use a for
loop. This is not
functional as it has
side effects.

can use get on default.

don't use map as a for loop alternative.
use it if you want change the original var.

```

63: println(list.groupBy { it })
64: var arrlist2 = list.map { x -> Pair(x, "Cats") }
65: println(arrlist2)
66:
67: val docHomePageHtml =
68:     """<html>
69:         <head>
70:             <title>Department of Computing</title>
71:         </head>
72:         <body>
73:             <p>Welcome to the Department of Computing at <a href="https://www.imperial.ac.uk">Imperial College London</a></p>
74:         </body>
75:     </html>"""
76:
77: val imperialHomePageHtml =
78:     """<html>
79:         <head>
80:             <title>Imperial College London</title>
81:         </head>
82:         <body>
83:             <p>Imperial people share ideas, expertise and technology to find
84:             answers to the big scientific questions and tackle global challenges</p>
85:             <p>See the latest news about our research on the <a href="https://www.bbc.co.uk/news">BBC</a></p>
86:         </body>
87:     </html>"""
88:
89: val bbcNewsPageHtml =
90:     """<html>
91:         <head>
92:             <title>BBC News</title>
93:         </head>
94:         <body>
95:             <p>Here is all the latest news about science and other things.</p>
96:         </body>
97:     </html>"""
98:
99: val docHomePage = WebPage(Jsoup.parse(docHomePageHtml))
100: val imperialHomePage = WebPage(Jsoup.parse(imperialHomePageHtml))
101: val bbcNews = WebPage(Jsoup.parse(bbcNewsPageHtml))
102:
103: val downloadedPages = mapOf(
104:     URL("https://www.imperial.ac.uk/computing") to docHomePage,
105:     URL("https://www.imperial.ac.uk") to imperialHomePage,
106:     URL("https://www.bbc.co.uk/news") to bbcNews
107: )
108:
109: val searchEngine = SearchEngine(downloadedPages)
110: println(searchEngine.compileIndex())
111: for (i in searchEngine.getIndex()) {
112:     println(i.value?.map { x -> x.url })
113:     println(i.value?.map { x -> x.numRefs })
114: }
115: println(searchEngine.getIndex())
116: }

```

```

1: package websearch
2:
3: import org.jsoup.nodes.Document
4:
5: class URL(private val URL: String) {
6:     override fun toString(): String {
7:         return "$URL"
8:     }
9: }
10:
11: class WebPage(private val jsoupDoc: Document) {
12:     fun extractWords(): List<String> {
13:         var text = jsoupDoc.text().filter { it != ' ' && it != '.' }
14:         .toLowerCase().trim()
15:         var word = ""
16:         var arrayListOfString = ArrayList<String>()
17:         for (i in 0..text.length - 1) {
18:             if (text[i].equals(' ')) {
19:                 arrayListOfString.add(word)
20:                 word = ""
21:             } else {
22:                 word += text[i]
23:             }
24:         }
25:         arrayListOfString.add(word)
26:         return arrayListOfString.toList()
27:     }
28: }

```

Can use
".split(" ")
instead.

Final Tests

testResults.txt: 1/1

Adithya Narayanan - anb122:j1:24

```
1: ----- Test Output -----
2: Running LabTS build... (Wed 29 Nov 01:15:09 UTC 2023)
3:
4: Submission summary...
5: You made 5 commits
6:   - f30a780 PPT completed [3 files changed, 150 insertions, 8 deletions]
7:   - faf4ebe style corrections [2 files changed, 13 insertions, 12 deletions]
8:   - 6aa7612 style corrections [2 files changed, 2 insertions, 1 deletion]
9:   - 9bf18dd Corrected style stuff [1 file changed, 2 insertions, 3 deletions]
10:  - 9004cb4 PLEASE TELL ME STYLE CHECK PASSED [1 file changed, 1 deletion]
11:
12: Preparing...
13:
14: BUILD SUCCESSFUL in 843ms
15:
16: Compiling...Path for java installation '/usr/lib/jvm/openjdk-17' (Common Linux Locations) does not contain a java executable
17:
18: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/JsoupExamples.kt:36:9 Variable 'link' is never used
19: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/JsoupExamples.kt:49:7 Variable 'firstResult' is never used
20: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/JsoupExamples.kt:55:7 Variable 'downloadedDocument' is never used
21: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:37:80 Unchecked cast: Map<String, List<SearchResult>?> to Map<String, List<SearchResult>>
22: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:42:9 Condition 'this.getIndex().getValue(query) != null' is always 'true'
23: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:62:7 Variable 'arrrlist' is never used
24: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:111:20 Unnecessary safe call on a non-null receiver of type List<SearchResult>
25: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:112:20 Unnecessary safe call on a non-null receiver of type List<SearchResult>
26: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/Web.kt:13:66 'toLowerCase(): String' is deprecated. Use lowercase() instead.
27:
28:
29: BUILD SUCCESSFUL in 14s
30:
31: Running tests...Path for java installation '/usr/lib/jvm/openjdk-17' (Common Linux Locations) does not contain a java executable
32:
33:
34: websearch.SearchEngineTest > can index downloaded pages PASSED
35:
36: websearch.WebPageTest > extracts words from page PASSED
37:
38: websearch.WebPageTest > converts case and removes punctuation PASSED
39:
40: BUILD SUCCESSFUL in 2s
41:
42: Checking code style...
43: BUILD SUCCESSFUL in 2s
44: Finished auto test. (Wed 29 Nov 01:15:56 UTC 2023)
45:
46: ----- Test Errors -----
47:
```

Test Preview**TestSummary.txt: 1/1 Adithya Narayanan - anb122:j1:24**

```
1: Test Preview: Summary for anb122 of j1
2: PPT 24
3: -----
4:
5:   Public Tests:
6:     Compiles:    1 / 1
7:     Tests Pass:  1 / 1
8:     Style Check: 1 / 1
9:
10: Git Repo: git@gitlab.doc.ic.ac.uk:lab2324_autumn/kotlinsearchengine_anb122.git
11: Commit ID: 9004c
```

Test Preview

JsoupExamples.kt: 1/1 Adithya Narayanan - anb122:j1:24

```

1: package websearch
2:
3: import org.jsoup.Jsoup.connect
4: import org.jsoup.Jsoup.parse
5: import org.jsoup.nodes.Document
6: import org.jsoup.select.Elements
7:
8: fun main() {
9:
10:     val htmlString =
11:         """<html>
12:             <head>
13:                 <title>A Simple HTML Document</title>
14:             </head>
15:             <body>
16:                 <p>This is a very simple <a href="https://
//en.wikipedia.org/wiki/HTML">HTML</a> document.</p>
17:                 <p>It only has two paragraphs.</p>
18:             </body>
19:         </html>"""
20:
21:         // parse a string into a Document using Jsoup.parse()
22:         val htmlDocument: Document = parse(htmlString)
23:
24:         // get all the text from the document
25:         // println(htmlDocument.text())
26:
27:         // extract all the <a >...</a> tags (which create links)
28:         val aTags: Elements = htmlDocument.getElementsByTag("a")
29:         // println(aTags)
30:
31:         // get the value of the href attribute (where the link points to)
32:         val firstTag = aTags.first()
33:
34:         // aTags.first() could return null, so we have to check that before using it.
35:         if (firstTag != null) {
36:             val link: String = firstTag.attr("href")
37:             // println(link)
38:         }
39:
40:         // A different approach to dealing with the nullable value is to use the "✓
safe-call" operator ?.
41:         // This says, if firstTag is null, don't call .attr(), just return null.
42:         // println(firstTag?.attr("href"))
43:
44:         // You can also use the "Elvis" operator ?: to use a specific value if the ✓
safe-call returns null
45:         // println(firstTag?.attr("href") ?: "")
46:
47:         // do the same thing using a CSS selector
48:         val selection: Elements = htmlDocument.select("a[href]")
49:         val firstResult = selection.first()
50:
51:         // println(firstResult?.attr("href"))
52:         // println(firstResult?.attr("href") ?: "")
53:
54:         // download a document from the internet using Jsoup.connect()
55:         val downloadedDocument = connect("https://kotlinlang.org/").get()
56:         // println(downloadedDocument.body())
57: }

```

Test Preview

SearchEngine.kt: 1/2 Adithya Narayanan - anb122:j1:24

```

1: package websearch
2:
3: import org.jsoup.Jsoup
4:
5: class SearchEngine(private val mapURLToWebPage: Map<URL, WebPage>) {
6:
7:     private var index = emptyMap<String, List<SearchResult>>()
8:     fun getIndex(): Map<String, List<SearchResult>> {
9:         return index
10:     }
11:
12:     // I have created the function without the need for the helper rank().
13:     // Please let me know if this is ok?
14:     // I have also purposefully used only maps as I am trying to practice using ✓
Haskell esque HOFs.
15:     fun compileIndex() {
16:         var URLToStrings = LinkedHashMap<URL, List<String>>()
17:         for (i in mapURLToWebPage) {
18:             URLToStrings.put(i.key, i.value.extractWords())
19:         }
20:         var URLToStringsAndCounts = URLToStrings.mapValues { it.value.map { x -> ✓
Pair(x, it.key) } }
21:         .mapValues { it.value.groupingBy { x -> x.first }.eachCount() }
22:         var indexMutableMap = mutableMapOf<String, ArrayList<SearchResult>>?()
23:         URLToStringsAndCounts
24:             .map { (url, stringcountpair) ->
25:                 stringcountpair
26:                     .map { (string, count) ->
27:                         if (string in indexMutableMap.keys) {
28:                             var searchResultList = indexMutableMap[string]
29:                             searchResultList?.add(SearchResult(url, count))
30:                             indexMutableMap.set(string, searchResultList)
31:                         } else {
32:                             indexMutableMap.put(string, arrayListOf(SearchResult(url, count)))
33:                         }
34:                     }
35:             }
36:         var indexMutableMapHolder = indexMutableMap.mapValues { x -> ✓
x.value?.sortedByDescending { y -> y.numRefs } }
37:         index = indexMutableMapHolder.mapValues { y -> y.value?.toList() }.toMap() as ✓
Map<String, List<SearchResult>>
38:     }
39:
40:     fun searchFor(query: String): SearchResultsSummary = SearchResultsSummary(
41:         query,
42:         if (this.getIndex().getValue(query) != null) {
43:             this.getIndex().getValue(query)
44:         } else {
45:             emptyList()
46:         }
47:     )
48: }
49:
50: class SearchResult(val url: URL, val numRefs: Int)
51:
52: class SearchResultsSummary(val query: String, val results: List<SearchResult>) {
53:     override fun toString(): String {
54:         var stringOut = "Results for $query:\n "
55:         results.map { stringOut += "${it.url} - ${it.numRefs} references\n" }
56:         return stringOut
57:     }
58: }
59:
60: fun main() {
61:     var list = mutableListOf(1, 1, 2, 2, 3, 4, 5, 4, 5, 6, 7, 8, 9)
62:     var arrlist = mutableListOf<Pair<Int, String>>()

```

Test Preview

SearchEngine.kt: 2/2 Adithya Narayanan - anb122:j1:24

```

63: println(list.groupBy { it })
64: var arrlist2 = list.map { x -> Pair(x, "Cats") }
65: println(arrlist2)
66:
67: val docHomePageHtml =
68:     """<html>
69:         <head>
70:             <title>Department of Computing</title>
71:         </head>
72:         <body>
73:             <p>Welcome to the Department of Computing at <a href="https://www.imperial.ac.uk">Imperial College London</a></p>
74:         </body>
75:     </html>"""
76:
77: val imperialHomePageHtml =
78:     """<html>
79:         <head>
80:             <title>Imperial College London</title>
81:         </head>
82:         <body>
83:             <p>Imperial people share ideas, expertise and technology to find answers to the big scientific questions and tackle global challenges</p>
84:             <p>See the latest news about our research on the <a href="https://www.bbc.co.uk/news">BBC</a></p>
85:         </body>
86:     </html>"""
87:
88: val bbcNewsPageHtml =
89:     """<html>
90:         <head>
91:             <title>BBC News</title>
92:         </head>
93:         <body>
94:             <p>Here is all the latest news about science and other things.</p>
95:         </body>
96:     </html>"""
97:
98: val docHomePage = WebPage(Jsoup.parse(docHomePageHtml))
99: val imperialHomePage = WebPage(Jsoup.parse(imperialHomePageHtml))
100: val bbcNews = WebPage(Jsoup.parse(bbcNewsPageHtml))
101:
102: val downloadedPages = mapOf(
103:     URL("https://www.imperial.ac.uk/computing") to docHomePage,
104:     URL("https://www.imperial.ac.uk") to imperialHomePage,
105:     URL("https://www.bbc.co.uk/news") to bbcNews
106: )
107:
108: val searchEngine = SearchEngine(downloadedPages)
109: println(searchEngine.compileIndex())
110: for (i in searchEngine.getIndex()) {
111:     println(i.value?.map { x -> x.url })
112:     println(i.value?.map { x -> x.numRefs })
113: }
114: println(searchEngine.getIndex())
115: }

```

Test Preview

Web.kt: 1/1

Adithya Narayanan - anb122:j1:24

```

1: package websearch
2:
3: import org.jsoup.nodes.Document
4:
5: class URL(private val URL: String) {
6:     override fun toString(): String {
7:         return "$URL"
8:     }
9: }
10:
11: class WebPage(private val jsoupDoc: Document) {
12:     fun extractWords(): List<String> {
13:         var text = jsoupDoc.text().filter { it != ',' && it != '.' }
14:         .toLowerCase().trim()
15:         var word = ""
16:         var arrayListOfString = ArrayList<String>()
17:         for (i in 0..text.length - 1) {
18:             if (text[i].equals(' ')) {
19:                 arrayListOfString.add(word)
20:                 word = ""
21:             } else {
22:                 word += text[i]
23:             }
24:             arrayListOfString.add(word)
25:             return arrayListOfString.toList()
26:         }
27:     }

```


Test Preview**testResults.txt: 1/1****Adithya Narayanan - anb122:j1:24**

```
1: ----- Test Output -----
2: Running LabTS build... (Wed 29 Nov 01:15:09 UTC 2023)
3:
4: Submission summary...
5: You made 5 commits
6:   - f30a780 PPT completed [3 files changed, 150 insertions, 8 deletions]
7:   - faf4ebe style corrections [2 files changed, 13 insertions, 12 deletions]
8:   - 6aa7612 style corrections [2 files changed, 2 insertions, 1 deletion]
9:   - 9bf18dd Corrected style stuff [1 file changed, 2 insertions, 3 deletions]
10:  - 9004cb4 PLEASE TELL ME STYLE CHECK PASSED [1 file changed, 1 deletion]
11:
12: Preparing...
13:
14: BUILD SUCCESSFUL in 843ms
15:
16: Compiling...Path for java installation '/usr/lib/jvm/openjdk-17' (Common Linux Locations) does not contain a java executable
17:
18: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/JsoupExamples.kt:36:9 Variable 'link' is never used
19: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/JsoupExamples.kt:49:7 Variable 'firstResult' is never used
20: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/JsoupExamples.kt:55:7 Variable 'downloadedDocument' is never used
21: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:37:80 Unchecked cast: Map<String, List<SearchResult>?> to Map<String, List<SearchResult>>
22: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:42:9 Condition 'this.getIndex().getValue(query) != null' is always 'true'
23: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:62:7 Variable 'arrlist' is never used
24: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:111:20 Unnecessary safe call on a non-null receiver of type List<SearchResult>
25: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/SearchEngine.kt:112:20 Unnecessary safe call on a non-null receiver of type List<SearchResult>
26: w: file:///tmp/d20231129-35-u2e8jp/src/main/kotlin/websearch/Web.kt:13:66 'toLowerCase(): String' is deprecated. Use lowercase() instead.
27:
28:
29: BUILD SUCCESSFUL in 14s
30:
31: Running tests...Path for java installation '/usr/lib/jvm/openjdk-17' (Common Linux Locations) does not contain a java executable
32:
33:
34: websearch.SearchEngineTest > can index downloaded pages PASSED
35:
36: websearch.WebPageTest > extracts words from page PASSED
37:
38: websearch.WebPageTest > converts case and removes punctuation PASSED
39:
40: BUILD SUCCESSFUL in 2s
41:
42: Checking code style...
43: BUILD SUCCESSFUL in 2s
44: Finished auto test. (Wed 29 Nov 01:15:56 UTC 2023)
45:
46: ----- Test Errors -----
47:
```