GladLibs

Quiz, 19 questions

1 point

1. Consider the first version of GladLibs we saw in this lesson, which stores label substitutions in ArrayLists. Assume an ArrayList named **wordsUsed** will keep track of words used as replacements so no replacement word will be used more than once. The code below was used as part of a program by a learner in the method **processWord**. The learner's program runs but still results in duplicate words sometimes.

```java
String sub = getSubstitute(w.substring(first+1,last));
while (true) {
    if (wordsUsed.contains(sub)) {
        sub = getSubstitute(w.substring(first+1,last));
        break;
    }
    else {
        wordsUsed.add(sub);
    }
}
```

Which one of the following best explains why this code still returns duplicates sometimes?

- ○ The “if condition” is always false the first time in the loop, so the else part is executed the first time through the loop. This means the while loop always executes its body at least twice so some words may be used a second time.
- ○ Repeated words are also put into **wordsUsed**, so the call to **getSubstitute** may choose a repeated word.
- ○ The “if condition” is always false in the loop, so the else part is always executed. This always results in a second random word.
- ● If a word is a repeated word, then this code gets another random word and uses that second word without checking to see if it is a repeated word.

---

1 point

2. Consider the first version of GladLibs we saw in this lesson, which you modified so there would not be duplicate words chosen for the story. Assume an instance variable is used to keep track of the total number of word tags that are replaced.

Which one of the following methods is most likely where that variable is updated?

- ○ **getSubstitute**
- ● **myStory**
- ○ **processWord**
- ○ The GladLibs constructor.

---

1 point

3. Consider the class **WordFrequencies**, which you wrote in an assignment, that can determine facts about words in a file.

How many unique words are in the file **errors.txt**?

(You should lowercase all words and include the punctuation as part of a word. Thus, “end.” is different than “end”, but “All” is the same as “all”.)

3721

---

1 point

4. Consider the class **WordFrequencies**, which you wrote in an assignment, that can determine facts about words in a file.

Which word occurs the most often in the file **errors.txt**?

(You should lowercase all words and include the punctuation as part of a word. Thus, “end.” is different than “end”, but “All” is the same as “all”.)

of

---

1 point

5. Consider the class **WordFrequencies**, which you wrote in an assignment, that can determine facts about words in a file.

Find the word that occurs the most often in the file **errors.txt**.

(You should lowercase all words and include the punctuation as part of a word. Thus, “end.” is different than “end”, but “All” is the same as “all”.)

How many times does the most common word occur?

609

---

1 point

6. Consider the class **CharactersInPlay**, which you wrote in an assignment, that determines who the characters were in one of Shakespeare’s plays and also how many lines they had.

Of the characters who have fewer than 100 lines in the file **errors.txt**, which of these characters has the most speaking parts?

ADRIANA

---

1 point

7. Consider the class **CharactersInPlay**, which you wrote in an assignment, that determines who the characters were in one of Shakespeare’s plays and also how many lines they had.

Find the name of the character with the <u>third</u> most speaking parts in the file **errors.txt**. How many speaking parts does this person have?

79

---

1 point

8. Consider the class **CharactersInPlay**, which you wrote in an assignment, that determines who the characters were in one of Shakespeare’s plays and also how many lines they had.

How many characters in the file **errors.txt** have at least 10 speaking parts, but no more than 15 speaking parts?

3

---

1 point

9. Consider the class you wrote to find out how many times each codon occurs in a strand of DNA based on reading frames. The file **dnaMystery2** represents a long strand of DNA.

How many unique codons are there if you use a reading frame that starts at position 1?

32

---

1 point

10. Consider the class you wrote to find out how many times each codon occurs in a strand of DNA based on reading frames. The file **dnaMystery2** represents a long strand of DNA.

What is the number of occurrences of the codon that occurs the most often using a reading frame that starts at position 2?

12

---

1 point

11. Consider the class you wrote to find out how many times each codon occurs in a strand of DNA based on reading frames. The file **dnaMystery2** represents a long strand of DNA.

Using a reading frame that starts at position 0, which of the following codons occur 7 times? (Select all that are correct.)

- ☑ **CAG**
- ☐ **TGT**
- ☐ **ATG**
- ☑ **CAA**
- ☐ **GCC**
- ☐ **GAT**

---

1 point

12. Consider the class **WordsInFiles**, which you wrote in an assignment, that determines which words occur in several files, and for each word, which files they occur in.

Consider the seven files: **caesar.txt**, **confucius.txt**, **errors.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt**, and **romeo.txt**.

How many words are there that each occur in all seven files?

570

---

1 point

13. Consider the class **WordsInFiles**, which you wrote in an assignment, that determines which words occur in several files, and for each word, which files they occur in.

Consider the seven files: **caesar.txt**, **confucius.txt**, **errors.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt** and **romeo.txt**.

How many words are there that each occur in <u>four</u> of the seven files?

826

---

1 point

14. Consider the class **WordsInFiles**, which you wrote in an assignment, that determines which words occur in several files, and for each word, which files they occur in.

Consider the seven files: **caesar.txt, confucius.txt, errors.txt, hamlet.txt, likeit.txt, macbeth.txt and romeo.txt**.

In which file does the word “laid” NOT appear?

- ● caesar.txt
- ○ confucius.txt
- ○ errors.txt
- ○ hamlet.txt
- ○ likeit.txt
- ○ macbeth.txt
- ○ romeo.txt

---

1 point

15. Consider the class **WordsInFiles**, which you wrote in an assignment, that determines which words occur in several files, and for each word, which files they occur in.

Consider the seven files: **caesar.txt**, **confucius.txt**, **errors.txt**, **hamlet.txt**, **likeit.txt**, **macbeth.txt** and **romeo.txt**.

In which of the following files does the word “tree” appear? (Choose all that apply.)

(Consider only the exact lowercase string "tree". "TREE" or "tree." would be different words.)

- ☐ caesar.txt
- ☑ confucius.txt
- ☐ errors.txt
- ☐ hamlet.txt
- ☑ likeit.txt
- ☑ macbeth.txt
- ☑ romeo.txt

---

1 point

16. Consider the map version of GladLibs where a map is created that maps a category to a list of words in that category.

In which method are the individual ArrayLists of words for categories created?

- ○ **makeStory**
- ○ They are created in the constructor.
- ○ They are not created in a method but are automatically created as part of the definition of the private HashMap variable of <String> to <ArrayList<String>>.
- ● **readIt**
- ○ **initializeFromSource**

---

1 point

17. Consider the map version of GladLibs where a map is created that maps a category to a list of words in that category. In which method are these individual ArrayLists of words placed into the HashMap?

- ● **initializeFromSource**
- ○ Not in a method, but rather, they are placed in automatically as part of the definition of the private HashMap variable of <String> to <ArrayList<String>>.
- ○ in the constructor
- ○ **readIt**
- ○ **makeStory**

---

1 point

18. Consider the map version of GladLibs and consider the method **totalWordsInMap** that returns the total number of words in all the ArrayLists in the HashMap **myMap**.

Which <u>two</u> of the following code possibilities compute this sum of total number of words in the variable **sum**?

☐

```java
int sum = 0;
for (String category : myMap.keySet()) {
    sum += myMap.get(category).size();
}
```

☐

```java
int sum = 0;
for (String category : myMap.keySet()) {
    ArrayList<String> words = myMap.get(category);
    sum += words;
}
```

☐

```java
int sum = 0;
for (String category : myMap.keySet()) {
    sum += myMap.get(category);
}
```

☐

```java
int sum = 0;
for (String category : myMap.keySet()) {
    ArrayList<String> words = myMap.get(category);
    sum += words.size();
}
```

☑

```java
int sum = 0;
for (ArrayList<String> wordlist : myMap.keySet()) {
    for (String word : wordlist) {
        sum += 1;
    }
}
```

☑

```java
int sum = 0;
for (ArrayList<String> wordlist : myMap.keySet()) {
    sum += wordlist.size();
}
```

---

1 point

19. Consider the map version of GladLibs and consider the method **totalWordsConsidered** that returns the total number of words in the ArrayLists of the categories that were used for a particular GladLib. Assume a private variable of type ArrayList<String> and named **categoriesUsed** is used to store the unique categories found as the GladLib is created.

In which method would we put a category into this ArrayList?

- ○ **makeStory**
- ○ **totalWordsConsidered**
- ○ **getSubstitute**
- ○ **fromTemplate**
- ○ **readIt**
- ● **processWord**

☑ I, **Ning Zheng**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera’s Honor Code

Submit Quiz