Course CSC 340.05

Student: Amit Joshi, SFSU ID: 918210532

Teammate: Nomar Olivas, SFSU ID: 918632343

Assignment Number: 02

Assignment due date and time: 9-21-2020 11:55

Part A-OOP Class design Guidelines

1.Instance Vs Static

An instance variable is a variable and method that only occurs once per instance of the class. Static variables including methods are shared by all objects of the class and you can easily change its value. Providing an example can be, **int number;** which is an example of an instance variable, “number” can be used throughout the class and is also a value which is set to a variable as well. Changes in the variable won’t affect other instance variables in that class. Also, we can set this variable on using methods and constructors. The instance variable is used by objects to store their states.

For Static, basically it’s the same but we add new words including ‘Static’ in front of the data type we are using for any variable used. Example is the same as used, for instance, **Static int number;** now the static variables are part of the whole entire class, while the instance is related to what I have described above. Instead before saying Instance change won’t be affected, the Static change can once you change it will affect all of the objects. Liang mentions that a constructor is always an instance, it can create a specific instance. Static variables also invoked by instance method, but don’t work the other way around.

2. Inheritance vs. Aggregation

On one of the Y. Daniel Lang's guidelines are knowing the difference between inheritance and aggregation is the difference between an is-a and a has-a relationship. Know this because in Java, inheritance is when a class gets other properties and methods from a different class. We can also call this the parent class while the other class is acquiring the traits from the parent class also called up as subclass. This makes the point of is-a relationship because they coincide. Is-a and has-a relation coincide with inheritance and aggregation. For example, **public class Student extends Person{**

This is how a class gets the properties and methods from another class, the main keyword as we can tell is "extends". Students and Persons are related to each other and the class Student would have a name first and last, social security number, etc. Another thing to note is that whatever you do to the parent class can affect the child's class of course. All because of is-a relation are dependent on each other.

Aggregation best describes as classes are not entirely related but a part of each class is related to each other as has-a relation is like an is-relationship to get the properties of a class. Has-a Relationship is an instance class for one which has reference to an instance of another class or another instance of the same class. Another example is called Transcript has-a relationship with the CS Student class but they aren't the same thing but are related. Transcript class gets information from Cs class which can include student ID, grades, and that creates objects in the transcript class.

3. Clarity

Cohesion, consistency, and encapsulation are the parts needed to having a good and clear program that we are coding. Cohesion is when a class is broken down into parts and each class has it's own doing and own operators. Also should have its own responsibilities since it will be organized as well very clear. For consistency is like choosing the right names for classes, data fields and methods. Data declaration before constructor and use the constructors before you choose to do methods. It way more proficient as well because you could run into trouble afterwards. Encapsulation is just a way to keep it more organized. So everything should be better also, it way clear on seeing what is happening and why we're using this in our program we are writing. Liang says many times that we have to write a clear program. That is the way we need to perform and every time we finish we need to see and examine on how we can make this program more better and proficient. So this also has a better clear way of understanding on what is happening and making life easy. For example, in this code, we can see clearly and read on what is happening during this return process.

```
return{
processSystem: func(){
getTheScores();
calcualteGrade()
```

4. Encapsulation

So, Encapsulation in java is a class that uses a private modifier to hide its data from direct access form any clients. More useful to maintain and keep everything organized. The variables or data of a class is also hidden from other classes also it can be accessed by the getter and the setter method we use in the program. The getter methods also provide data to be readable. For the setter method they make data fields ready. Setting up an encapsulation method first we need to make variables private. Which means they can't be accessed from another class outside. Now for an example, this is how we set our private variables up. Just because it hides data from direct access from outside of the class

```
Class Encapsulation{

private int num;

private String myName;

Private int ownAge;
```

Then we do our setter and getter methods,

```
      Public int getNum(){
      return num;
       }
   Public String getMyName(){
   return myName;
  }
   Public String getOwnAge(){
   return ownAge;
  }
  Public void setGetNum(int newValue{
 getNum = newValue;
}
 Public void setMyName(String newValue){
 getName= newValue;
}
```

5. Interfaces vs. Abstract Classes

Liang made another argument which made it very clear to me as well on Interface and Abstract classes. Both can be used to specify common behavior on objects. We have differences as well together, type of method and interface can only have abstract methods only and variables can only be declared as Final only. Interfaces can only have static and final variables and they can be implemented by using “implements” only. Liang mentions that interfaces are more flexible than abstract because you can implement an abundant number of interfaces, while abstract classes are only restricted to extending to one class.

For Example of the interface:

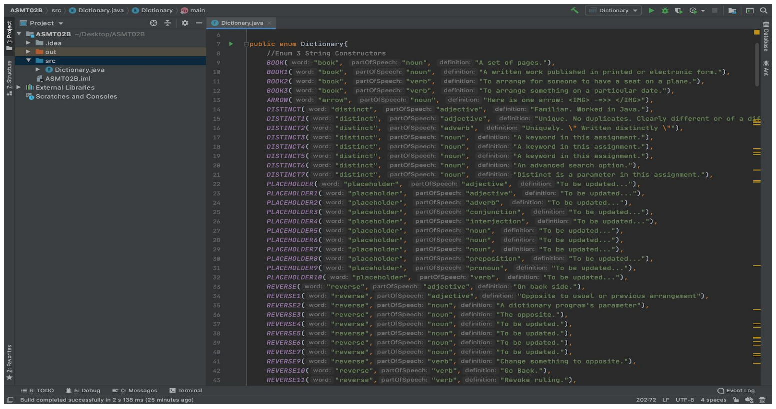
```
public interface linkedBag<T{
public int getCurrentSize();
public boolean isEmpty();
public final class resizeableArrayBag<T> implements linkedBag<T>{
```

What the example I used is the class resizeableArrayBag is implementing all methods from linkedBag. You can see that the Abstract class is much of the opposite of the interface but it has its own ways of big same/different. The Abstract class can have different types of methods like abstract and non-abstract methods as we see but the classes have other type variables including final, non- final, static. For the abstract classes can implementation from interface classes but just got to use the keyword EXTEND!

Part B

After reviewing the complete sample output and making sure line by line is given for the dictionary, the program is to be clean and the best fit for the client. In this way, the client could add a word, part of speech, and definition in order to get the right copy needed from the dictionary. Another way we can say is the client shouldn't go line by line knowing what is needed. Instead what they can do is add a word and get the result they need. The problem we are solving is that we had to take information from the database which was the enum, rather where it came from we had to make sure it was ready and accessible for the client. So we used the enum class, each entries is made up of three strings that correspond to the word's keyword and part of speech, definition.Why did we store this data? Because if multiple entries have the same keyword, those entries can be stored in their own keyword or even their own part of speech. I have also chosen to do my program using a hashmap array list because it's way easier and proficient to utilize. The key and array list for values as well which makes the range of capabilities bigger then other data structures. So also I can make values of entries in the hashmap and the keys for such entries into a multimap and key into the hashmap so it can be together as a data structure.

2. Dictionary program starts with enum class Entry, and it's where all of the keywords and part of speech definitions and words are listed as well. If the client wanted to add a word to the dictionary you would add enum entry with the words keyword, part of speech, and definition.



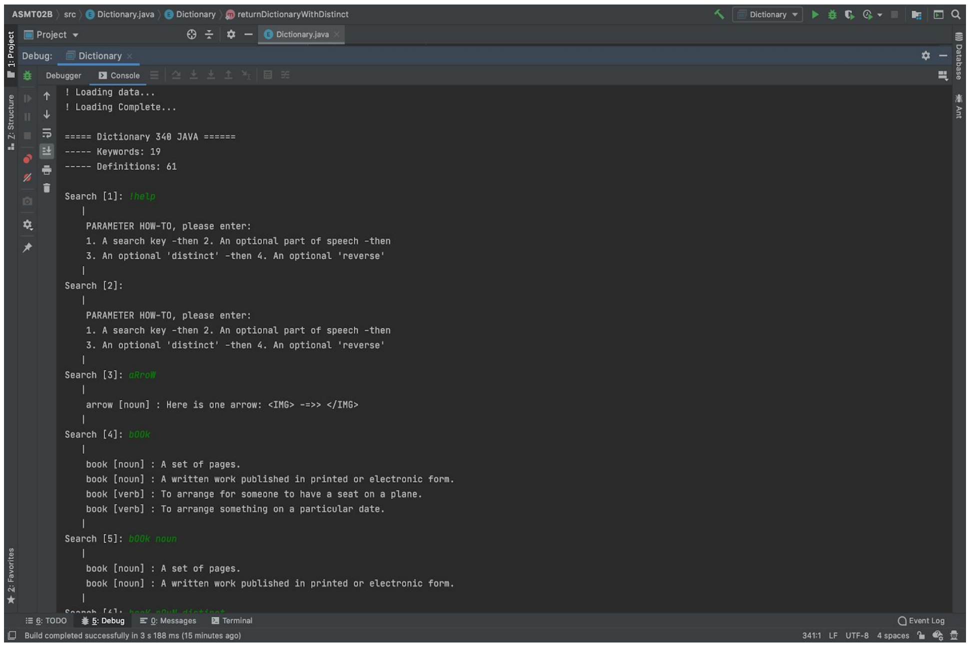
```java

public enum Dictionary{
    //Enum 3 String Constructors
    BOOK( word: "book",  partOfSpeech: "noun",  definition: "A set of pages."),
    BOOK1( word: "book",  partOfSpeech: "noun",  definition: "A written work published in printed or electronic form."),
    BOOK2( word: "book",  partOfSpeech: "verb",  definition: "To arrange for someone to have a seat on a plane."),
    BOOK3( word: "book",  partOfSpeech: "verb",  definition: "To arrange something on a particular date."),
    ARROW( word: "arrow",  partOfSpeech: "noun",  definition: "Here is one arrow: <IMG> -=>> </IMG>"),
    DISTINCT( word: "distinct",  partOfSpeech: "adjective",  definition: "Familiar. Worked in Java."),
    DISTINCT1( word: "distinct",  partOfSpeech: "adjective",  definition: "Unique. No duplicates. Clearly different or of a dif
    DISTINCT2( word: "distinct",  partOfSpeech: "adverb",  definition: "Uniquely. \" Written distinctly \""),
    DISTINCT3( word: "distinct",  partOfSpeech: "noun",  definition: "A keyword in this assignment."),
    DISTINCT4( word: "distinct",  partOfSpeech: "noun",  definition: "A keyword in this assignment."),
    DISTINCT5( word: "distinct",  partOfSpeech: "noun",  definition: "A keyword in this assignment."),
    DISTINCT6( word: "distinct",  partOfSpeech: "noun",  definition: "An advanced search option."),
    DISTINCT7( word: "distinct",  partOfSpeech: "noun",  definition: "Distinct is a parameter in this assignment."),
    PLACEHOLDER( word: "placeholder",  partOfSpeech: "adjective",  definition: "To be updated..."),
    PLACEHOLDER1( word: "placeholder",  partOfSpeech: "adjective",  definition: "To be updated..."),
    PLACEHOLDER2( word: "placeholder",  partOfSpeech: "adverb",  definition: "To be updated..."),
    PLACEHOLDER3( word: "placeholder",  partOfSpeech: "conjunction",  definition: "To be updated..."),
    PLACEHOLDER4( word: "placeholder",  partOfSpeech: "interjection",  definition: "To be updated..."),
    PLACEHOLDER5( word: "placeholder",  partOfSpeech: "noun",  definition: "To be updated..."),
    PLACEHOLDER6( word: "placeholder",  partOfSpeech: "noun",  definition: "To be updated..."),
    PLACEHOLDER7( word: "placeholder",  partOfSpeech: "noun",  definition: "To be updated..."),
    PLACEHOLDER8( word: "placeholder",  partOfSpeech: "preposition",  definition: "To be updated..."),
    PLACEHOLDER9( word: "placeholder",  partOfSpeech: "pronoun",  definition: "To be updated..."),
    PLACEHOLDER10( word: "placeholder",  partOfSpeech: "verb",  definition: "To be updated..."),
    REVERSE( word: "reverse", partOfSpeech: "adjective", definition: "On back side."),
    REVERSE1( word: "reverse", partOfSpeech: "adjective", definition: "Opposite to usual or previous arrangement"),
    REVERSE2( word: "reverse", partOfSpeech: "noun", definition: "A dictionary program's parameter"),
    REVERSE3( word: "reverse", partOfSpeech: "noun", definition: "The opposite."),
    REVERSE4( word: "reverse", partOfSpeech: "noun", definition: "To be updated."),
    REVERSE5( word: "reverse", partOfSpeech: "noun", definition: "To be updated."),
    REVERSE6( word: "reverse", partOfSpeech: "noun", definition: "To be updated."),
    REVERSE7( word: "reverse", partOfSpeech: "noun", definition: "To be updated."),
    REVERSE9( word: "reverse", partOfSpeech: "verb", definition: "Change something to opposite."),
    REVERSE10( word: "reverse", partOfSpeech: "verb", definition: "Go Back."),
    REVERSE11( word: "reverse", partOfSpeech: "verb", definition: "Revoke ruling."),
```

Build completed successfully in 2 s 138 ms (25 minutes ago)

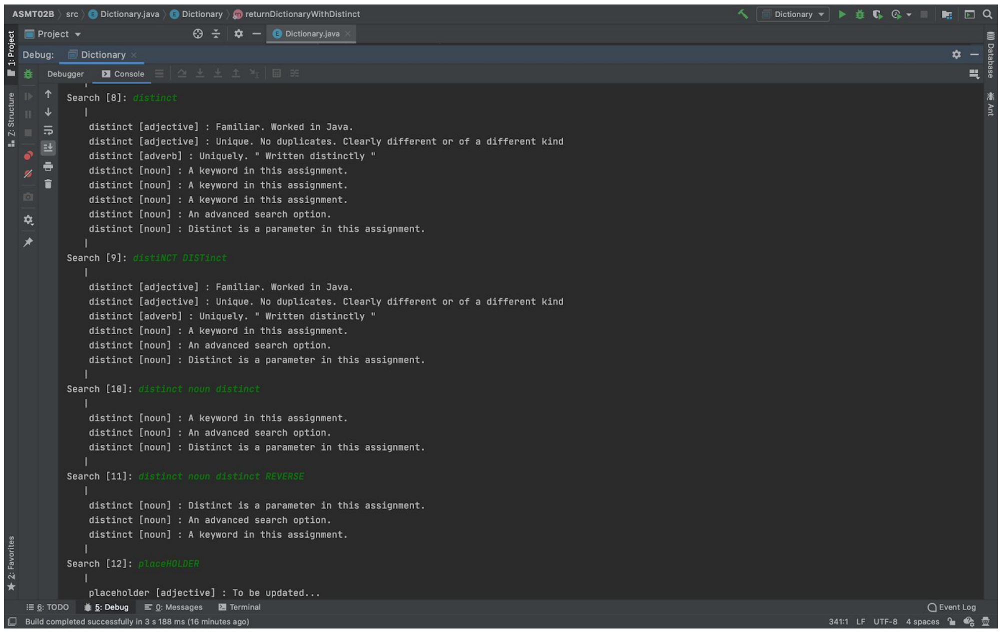
The program puts enum values into HashMap, and uses a loop through each entry as well. My program has also used If statements and for loops to match the user input. Another great way is if the user inputs the key then it will print out all the values that are with the key! Anything else it will correspond to the correct! I have also made comments all around my code. My knowledge in explaining in CS is getting quite better so I have made great statements in my program on what's happening. So all programs have used enum, hashmaps and array list to store all words and definitions as well. Add 14 extra words to make 61 definitions and meet the basic program sample output. Me and my partner had same code but we had some differences which I think was correct including print statement needing hashmap size and Keywords etc, my partner just said print statement with the number of definitions for example, `System.out.println("----- Keywords: 19"` Then I did `System.out.println("----- Keywords: "+ System.out.println("----- Keywords: " + hashMap.size()).`

I tried and continue to work hard as much as I can and I got it done on the last day. This project was hard for me. I really enjoyed as well was lot's of problem solving as well. The part I struggled the most was for distinct and reverse. Yes my program works properly I wrote as efficiently I can and I believe there was no hard coding involved at all.. Once I figured out everything else was smoove as possible. I used many different approaches as well first off I used MultiMap and then I decided to refuse that and use HashMap. As now everything works and if I had more time I would re-read my code to make more spaces!

<u>Screenshots of my program down below!</u>

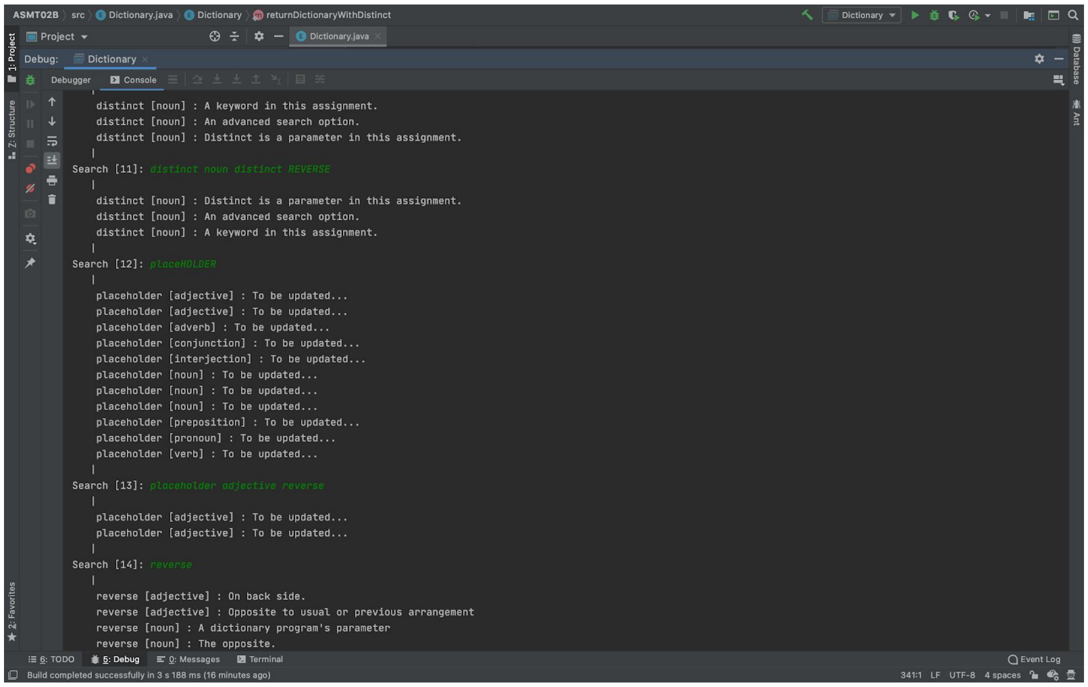

ASMT02B src Dictionary.java Dictionary returnDictionaryWithDistinct
Project
Dictionary.java
Debug: Dictionary
Debugger Console
6: TODO 5: Debug 0: Messages Terminal
Event Log
Build completed successfully in 3 s 188 ms (15 minutes ago)
341:1 LF UTF-8 4 spaces

```
! Loading data...
! Loading Complete...

===== Dictionary 340 JAVA ======
----- Keywords: 19
----- Definitions: 61

Search [1]: !help
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [2]:
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [3]: aRroW
   |
    arrow [noun] : Here is one arrow: <IMG> -=>> </IMG>
   |
Search [4]: bOOk
   |
    book [noun] : A set of pages.
    book [noun] : A written work published in printed or electronic form.
    book [verb] : To arrange for someone to have a seat on a plane.
    book [verb] : To arrange something on a particular date.
   |
Search [5]: bOOk noun
   |
    book [noun] : A set of pages.
    book [noun] : A written work published in printed or electronic form.
   |
```



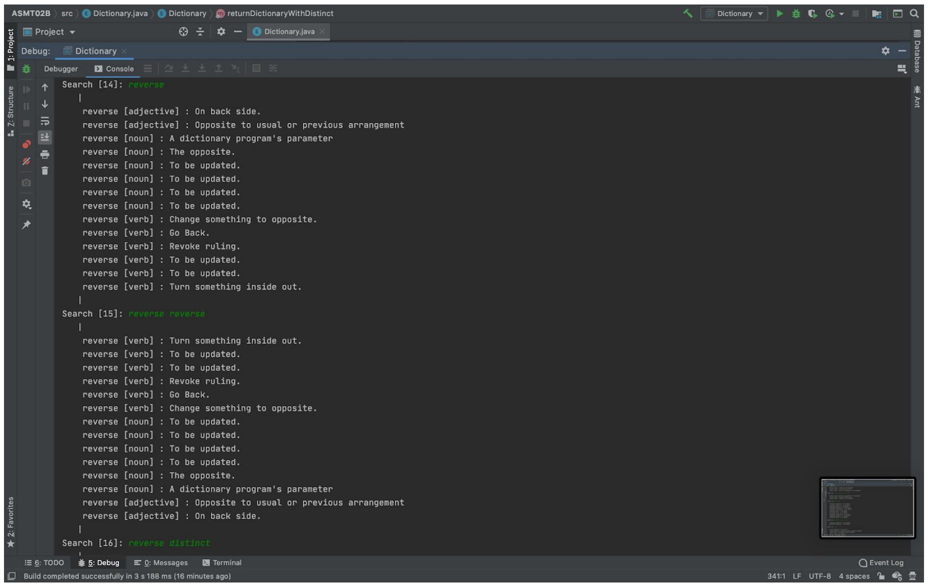
```
Search [8]: distinct
  |
   distinct [adjective] : Familiar. Worked in Java.
   distinct [adjective] : Unique. No duplicates. Clearly different or of a different kind
   distinct [adverb] : Uniquely. " Written distinctly "
   distinct [noun] : A keyword in this assignment.
   distinct [noun] : A keyword in this assignment.
   distinct [noun] : A keyword in this assignment.
   distinct [noun] : An advanced search option.
   distinct [noun] : Distinct is a parameter in this assignment.
  |
Search [9]: distiNCT DISTinct
  |
   distinct [adjective] : Familiar. Worked in Java.
   distinct [adjective] : Unique. No duplicates. Clearly different or of a different kind
   distinct [adverb] : Uniquely. " Written distinctly "
   distinct [noun] : A keyword in this assignment.
   distinct [noun] : An advanced search option.
   distinct [noun] : Distinct is a parameter in this assignment.
  |
Search [10]: distinct noun distinct
  |
   distinct [noun] : A keyword in this assignment.
   distinct [noun] : An advanced search option.
   distinct [noun] : Distinct is a parameter in this assignment.
  |
Search [11]: distinct noun distinct REVERSE
  |
   distinct [noun] : Distinct is a parameter in this assignment.
   distinct [noun] : An advanced search option.
   distinct [noun] : A keyword in this assignment.
  |
Search [12]: placeHOLDER
  |
   placeholder [adjective] : To be updated...
```

Build completed successfully in 3 s 188 ms (16 minutes ago)

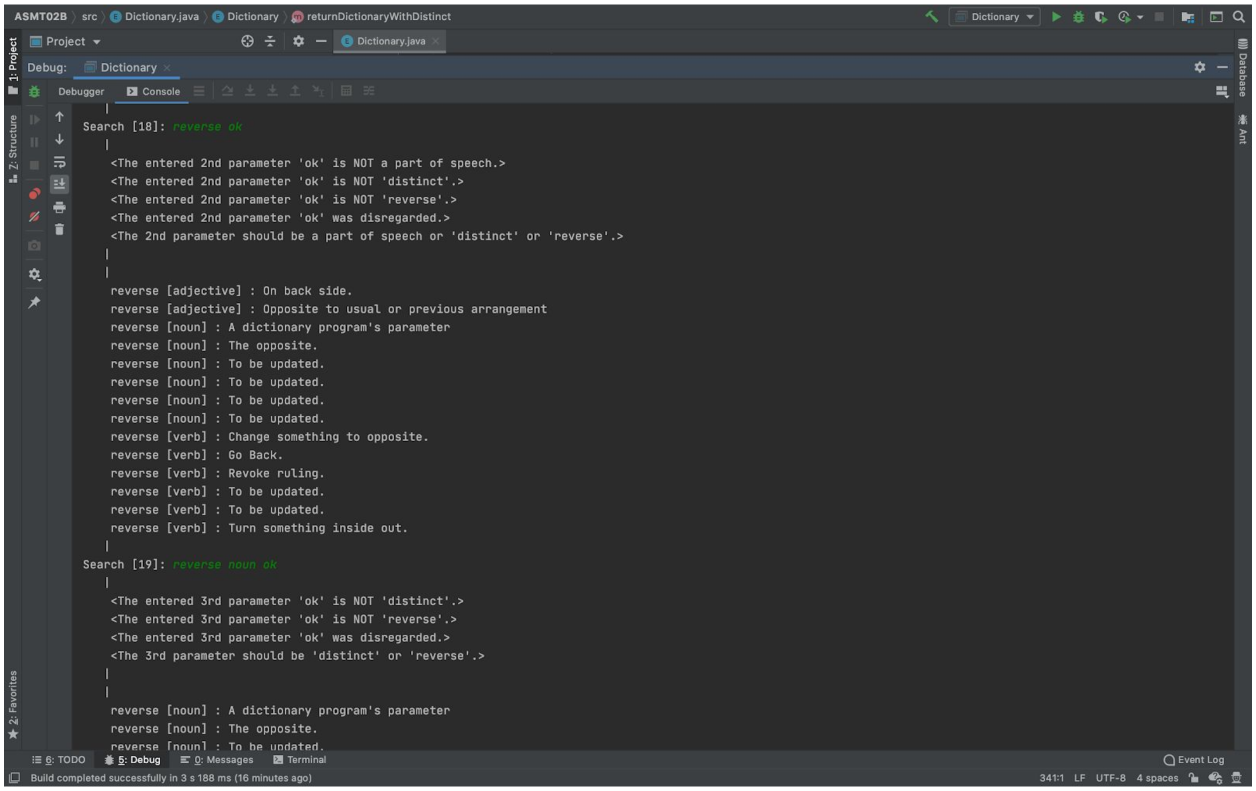


```
   distinct [noun] : A keyword in this assignment.
   distinct [noun] : An advanced search option.
   distinct [noun] : Distinct is a parameter in this assignment.
  |
Search [11]: distinct noun distinct REVERSE
  |
   distinct [noun] : Distinct is a parameter in this assignment.
   distinct [noun] : An advanced search option.
   distinct [noun] : A keyword in this assignment.
  |
Search [12]: placeHOLDER
  |
   placeholder [adjective] : To be updated...
   placeholder [adjective] : To be updated...
   placeholder [adverb] : To be updated...
   placeholder [conjunction] : To be updated...
   placeholder [interjection] : To be updated...
   placeholder [noun] : To be updated...
   placeholder [noun] : To be updated...
   placeholder [noun] : To be updated...
   placeholder [preposition] : To be updated...
   placeholder [pronoun] : To be updated...
   placeholder [verb] : To be updated...
  |
Search [13]: placeholder adjective reverse
  |
   placeholder [adjective] : To be updated...
   placeholder [adjective] : To be updated...
  |
Search [14]: reverse
  |
   reverse [adjective] : On back side.
   reverse [adjective] : Opposite to usual or previous arrangement
   reverse [noun] : A dictionary program's parameter
   reverse [noun] : The opposite.
```

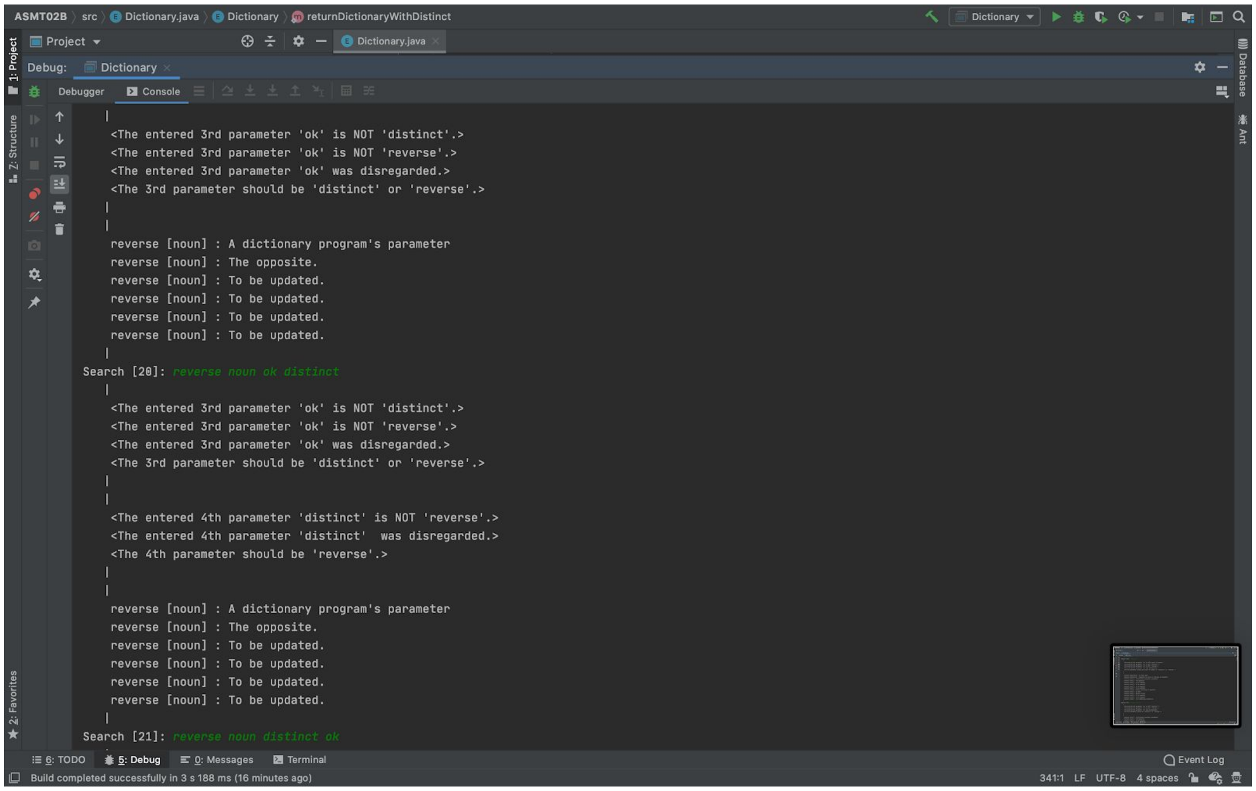
Build completed successfully in 3 s 188 ms (16 minutes ago)



```
Search [14]: reverse
    |
     reverse [adjective] : On back side.
     reverse [adjective] : Opposite to usual or previous arrangement
     reverse [noun] : A dictionary program's parameter
     reverse [noun] : The opposite.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [verb] : Change something to opposite.
     reverse [verb] : Go Back.
     reverse [verb] : Revoke ruling.
     reverse [verb] : To be updated.
     reverse [verb] : To be updated.
     reverse [verb] : Turn something inside out.
    |
Search [15]: reverse reverse
    |
     reverse [verb] : Turn something inside out.
     reverse [verb] : To be updated.
     reverse [verb] : To be updated.
     reverse [verb] : Revoke ruling.
     reverse [verb] : Go Back.
     reverse [verb] : Change something to opposite.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : The opposite.
     reverse [noun] : A dictionary program's parameter
     reverse [adjective] : Opposite to usual or previous arrangement
     reverse [adjective] : On back side.
    |
Search [16]: reverse distinct
```

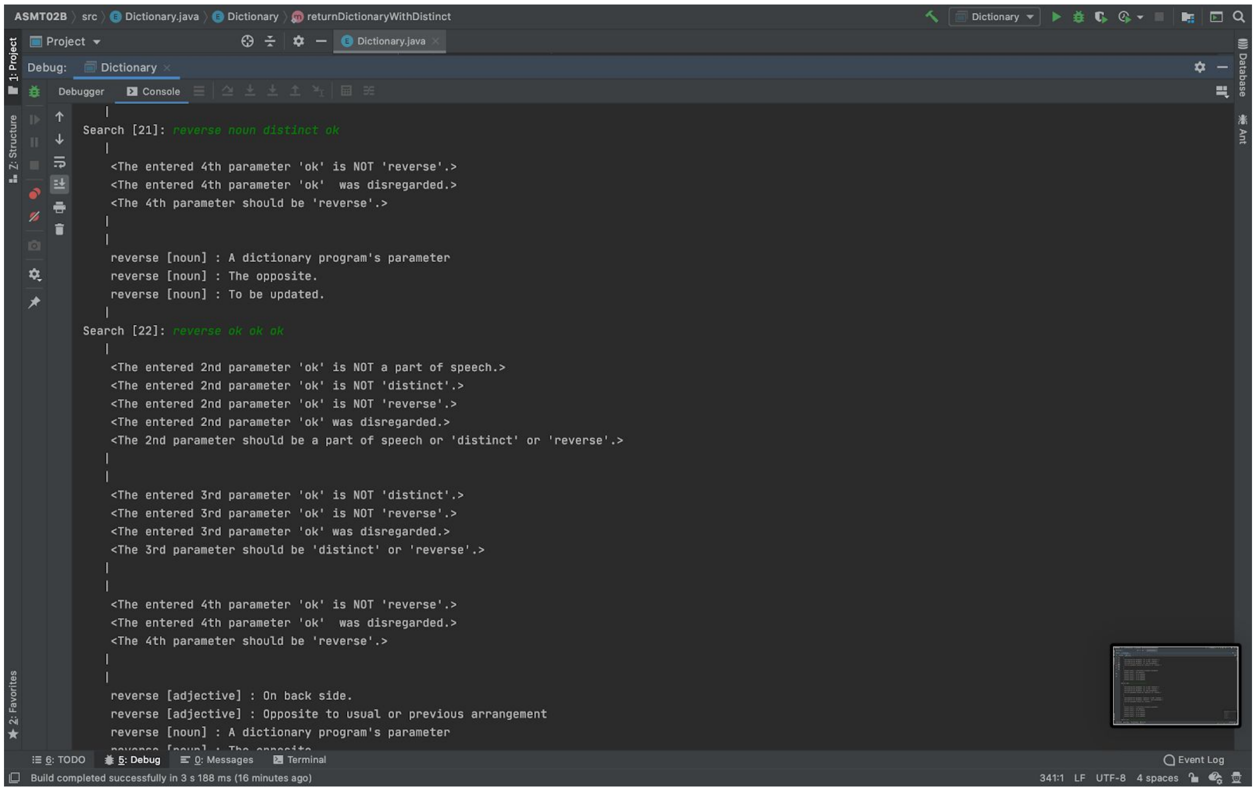


```
    |
Search [18]: reverse ok
    |
     <The entered 2nd parameter 'ok' is NOT a part of speech.>
     <The entered 2nd parameter 'ok' is NOT 'distinct'.>
     <The entered 2nd parameter 'ok' is NOT 'reverse'.>
     <The entered 2nd parameter 'ok' was disregarded.>
     <The 2nd parameter should be a part of speech or 'distinct' or 'reverse'.>
    |
    |
     reverse [adjective] : On back side.
     reverse [adjective] : Opposite to usual or previous arrangement
     reverse [noun] : A dictionary program's parameter
     reverse [noun] : The opposite.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [verb] : Change something to opposite.
     reverse [verb] : Go Back.
     reverse [verb] : Revoke ruling.
     reverse [verb] : To be updated.
     reverse [verb] : To be updated.
     reverse [verb] : Turn something inside out.
    |
Search [19]: reverse noun ok
    |
     <The entered 3rd parameter 'ok' is NOT 'distinct'.>
     <The entered 3rd parameter 'ok' is NOT 'reverse'.>
     <The entered 3rd parameter 'ok' was disregarded.>
     <The 3rd parameter should be 'distinct' or 'reverse'.>
    |
    |
     reverse [noun] : A dictionary program's parameter
     reverse [noun] : The opposite.
```



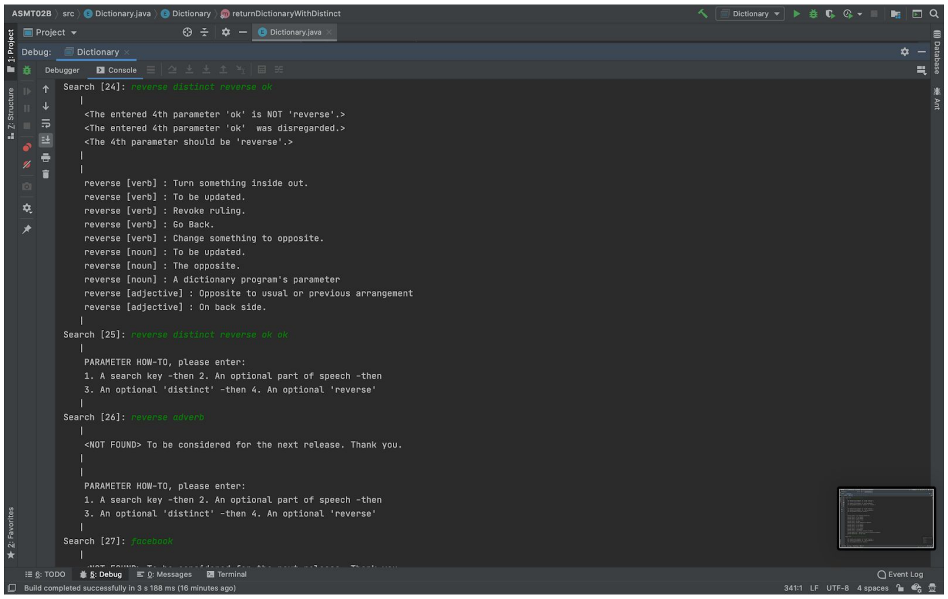
```
   |
    <The entered 3rd parameter 'ok' is NOT 'distinct'.>
    <The entered 3rd parameter 'ok' is NOT 'reverse'.>
    <The entered 3rd parameter 'ok' was disregarded.>
    <The 3rd parameter should be 'distinct' or 'reverse'.>
   |
   |
    reverse [noun] : A dictionary program's parameter
    reverse [noun] : The opposite.
    reverse [noun] : To be updated.
    reverse [noun] : To be updated.
    reverse [noun] : To be updated.
    reverse [noun] : To be updated.
   |
Search [20]: reverse noun ok distinct
   |
    <The entered 3rd parameter 'ok' is NOT 'distinct'.>
    <The entered 3rd parameter 'ok' is NOT 'reverse'.>
    <The entered 3rd parameter 'ok' was disregarded.>
    <The 3rd parameter should be 'distinct' or 'reverse'.>
   |
   |
    <The entered 4th parameter 'distinct' is NOT 'reverse'.>
    <The entered 4th parameter 'distinct'  was disregarded.>
    <The 4th parameter should be 'reverse'.>
   |
   |
    reverse [noun] : A dictionary program's parameter
    reverse [noun] : The opposite.
    reverse [noun] : To be updated.
    reverse [noun] : To be updated.
    reverse [noun] : To be updated.
    reverse [noun] : To be updated.
   |
Search [21]: reverse noun distinct ok
```

Build completed successfully in 3 s 188 ms (16 minutes ago)

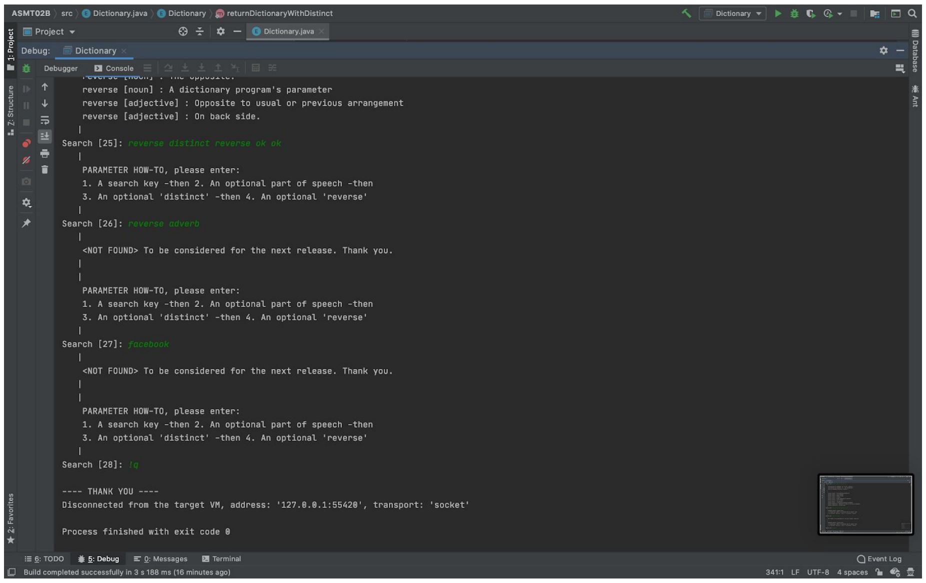


```
   |
Search [21]: reverse noun distinct ok
   |
    <The entered 4th parameter 'ok' is NOT 'reverse'.>
    <The entered 4th parameter 'ok'  was disregarded.>
    <The 4th parameter should be 'reverse'.>
   |
   |
    reverse [noun] : A dictionary program's parameter
    reverse [noun] : The opposite.
    reverse [noun] : To be updated.
   |
Search [22]: reverse ok ok ok
   |
    <The entered 2nd parameter 'ok' is NOT a part of speech.>
    <The entered 2nd parameter 'ok' is NOT 'distinct'.>
    <The entered 2nd parameter 'ok' is NOT 'reverse'.>
    <The entered 2nd parameter 'ok' was disregarded.>
    <The 2nd parameter should be a part of speech or 'distinct' or 'reverse'.>
   |
   |
    <The entered 3rd parameter 'ok' is NOT 'distinct'.>
    <The entered 3rd parameter 'ok' is NOT 'reverse'.>
    <The entered 3rd parameter 'ok' was disregarded.>
    <The 3rd parameter should be 'distinct' or 'reverse'.>
   |
   |
    <The entered 4th parameter 'ok' is NOT 'reverse'.>
    <The entered 4th parameter 'ok'  was disregarded.>
    <The 4th parameter should be 'reverse'.>
   |
   |
    reverse [adjective] : On back side.
    reverse [adjective] : Opposite to usual or previous arrangement
    reverse [noun] : A dictionary program's parameter
```

Build completed successfully in 3 s 188 ms (16 minutes ago)

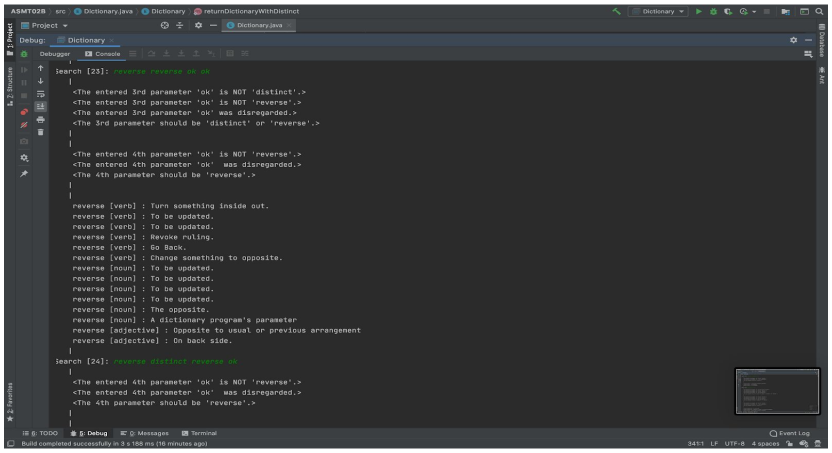


```
Search [24]: reverse distinct reverse ok
   |
    <The entered 4th parameter 'ok' is NOT 'reverse'.>
    <The entered 4th parameter 'ok'  was disregarded.>
    <The 4th parameter should be 'reverse'.>
   |
   |
    reverse [verb] : Turn something inside out.
    reverse [verb] : To be updated.
    reverse [verb] : Revoke ruling.
    reverse [verb] : Go Back.
    reverse [verb] : Change something to opposite.
    reverse [noun] : To be updated.
    reverse [noun] : The opposite.
    reverse [noun] : A dictionary program's parameter
    reverse [adjective] : Opposite to usual or previous arrangement
    reverse [adjective] : On back side.
   |
Search [25]: reverse distinct reverse ok ok
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [26]: reverse adverb
   |
    <NOT FOUND> To be considered for the next release. Thank you.
   |
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [27]: facebook
   |
```



```
    reverse [noun] : A dictionary program's parameter
    reverse [adjective] : Opposite to usual or previous arrangement
    reverse [adjective] : On back side.
   |
Search [25]: reverse distinct reverse ok ok
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [26]: reverse adverb
   |
    <NOT FOUND> To be considered for the next release. Thank you.
   |
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [27]: facebook
   |
    <NOT FOUND> To be considered for the next release. Thank you.
   |
   |
    PARAMETER HOW-TO, please enter:
    1. A search key -then 2. An optional part of speech -then
    3. An optional 'distinct' -then 4. An optional 'reverse'
   |
Search [28]: !q

---- THANK YOU ----
Disconnected from the target VM, address: '127.0.0.1:55420', transport: 'socket'

Process finished with exit code 0
```

```
Build completed successfully in 3 s 188 ms (16 minutes ago)
```



ASMT02B 〉 src 〉 Dictionary.java 〉 Dictionary 〉 returnDictionaryWithDistinct
Project
Dictionary.java
Debug: Dictionary
Debugger
Console

```
Search [23]: reverse reverse ok ok
    |
     <The entered 3rd parameter 'ok' is NOT 'distinct'.>
     <The entered 3rd parameter 'ok' is NOT 'reverse'.>
     <The entered 3rd parameter 'ok' was disregarded.>
     <The 3rd parameter should be 'distinct' or 'reverse'.>
    |
    |
     <The entered 4th parameter 'ok' is NOT 'reverse'.>
     <The entered 4th parameter 'ok'  was disregarded.>
     <The 4th parameter should be 'reverse'.>
    |
    |
     reverse [verb] : Turn something inside out.
     reverse [verb] : To be updated.
     reverse [verb] : To be updated.
     reverse [verb] : Revoke ruling.
     reverse [verb] : Go Back.
     reverse [verb] : Change something to opposite.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : To be updated.
     reverse [noun] : The opposite.
     reverse [noun] : A dictionary program's parameter
     reverse [adjective] : Opposite to usual or previous arrangement
     reverse [adjective] : On back side.
    |
Search [24]: reverse distinct reverse ok
    |
     <The entered 4th parameter 'ok' is NOT 'reverse'.>
     <The entered 4th parameter 'ok'  was disregarded.>
     <The 4th parameter should be 'reverse'.>
    |
    |
```

Build completed successfully in 3 s 188 ms (16 minutes ago)