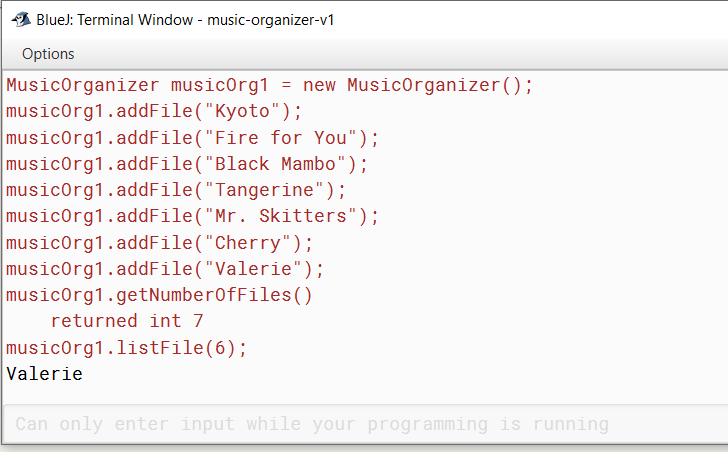
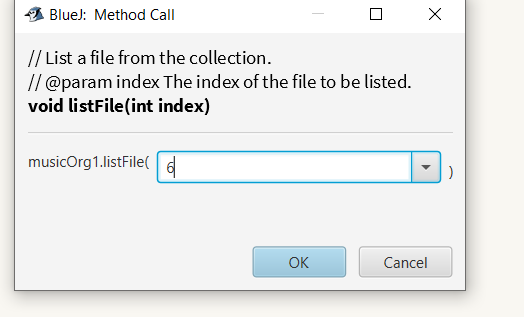
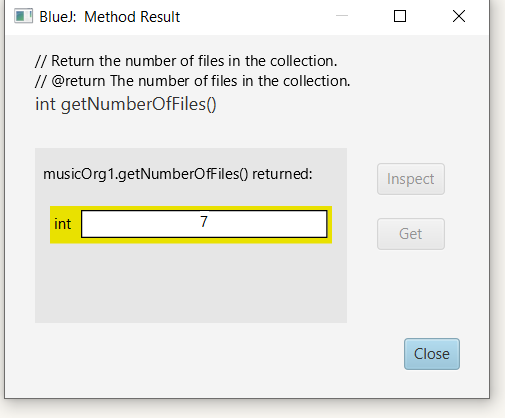
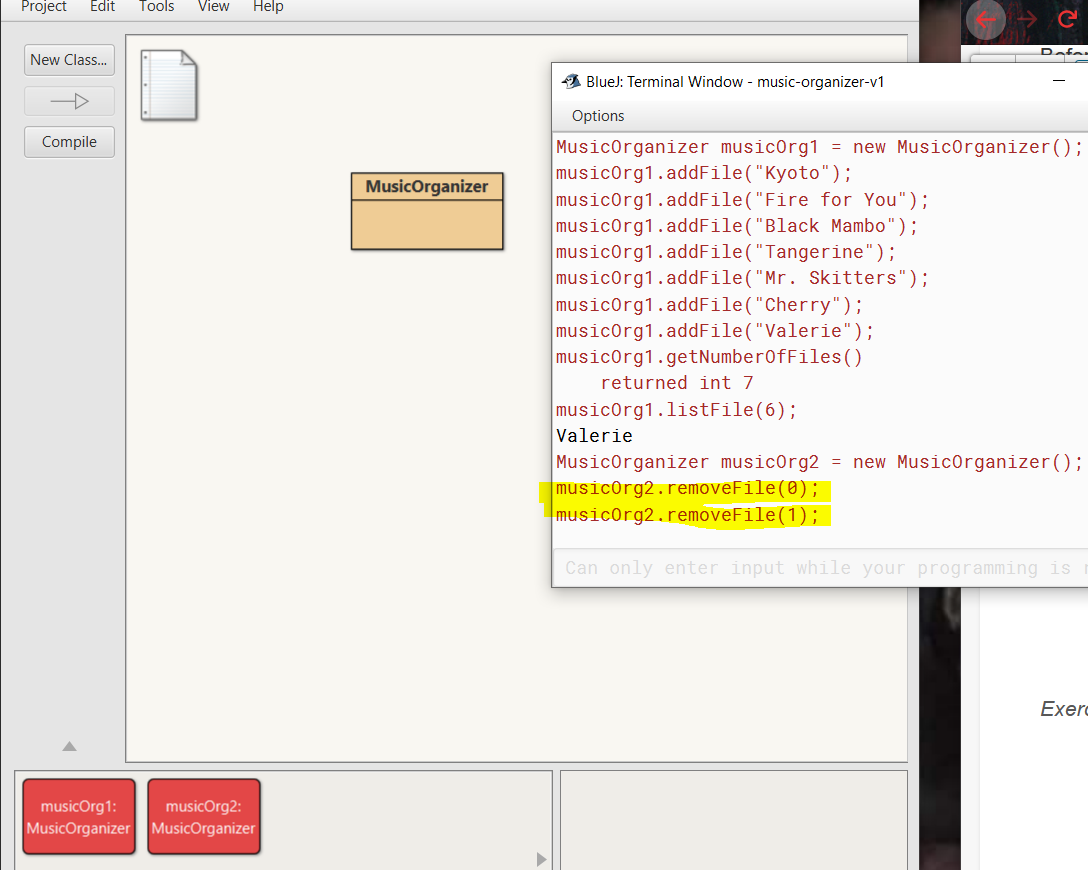
4.1

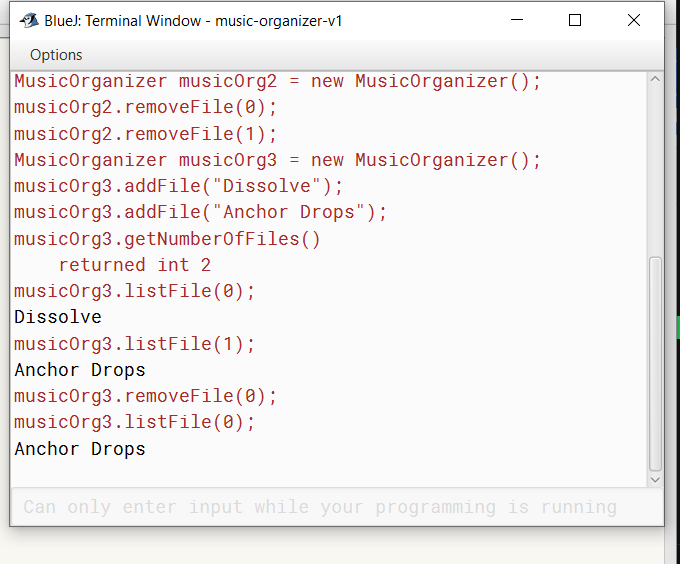
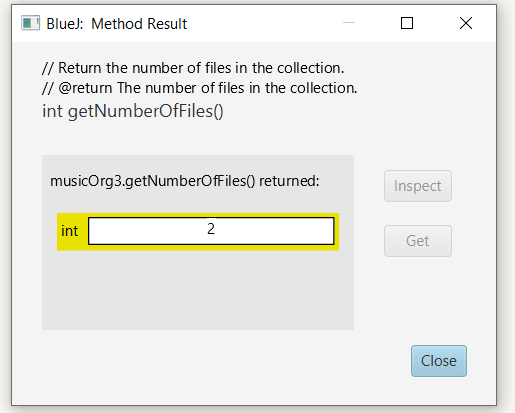


4.2



As you can see, I tried to removeFile(0) and (1). Although there are no files in either of these locations I do not get an error message. I kind of would have expected to get one for sure since there is nothing located at these indices. Maybe it could say “No file located at List index (x)”, or something like that. However, nothing happens really. Which in this case isn't really a big deal since it was going to remove what was there anyway. Although, I do feel like it would have been helpful to know I was trying to remove something that wasn’t there. Maybe I was referencing the wrong list or something and actually meant to remove something real from another list.

4.3



What happens is that when the file at (0) is removed it opened the first index (0) spot. So, then the file name that was in (1) is now actually in spot (0). This is because it is the next first one in the list now that the file that was in (0) is gone. This is what I would expect since the index is just a way to determine where in the list an item is but, it is not an ID number.

4.4

Private ArrayList<Book> library;

4.5

ArrayList<student> cs101;

4.6

Private ArrayList<MusicTrack> tracks;

4.7

library = new ArrayList<Book>;

cs101 = new ArrayList<student>;

tracks = new ArrayList<MusicTrack>;

library = new ArrayList<>;

cs101 = new ArrayList<>;

tracks = new ArrayList<>;

4.8

It would have a size of 10.

4.9

items.get(4);

4.10

The last object would be at index 14.

4.11

files.add(favoriteTrack);

4.12

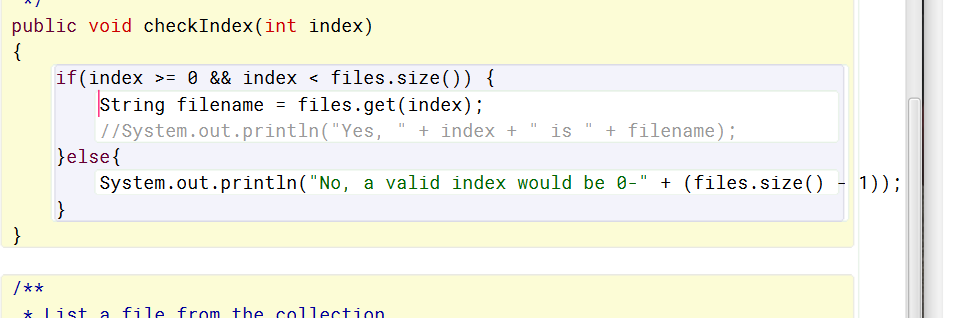
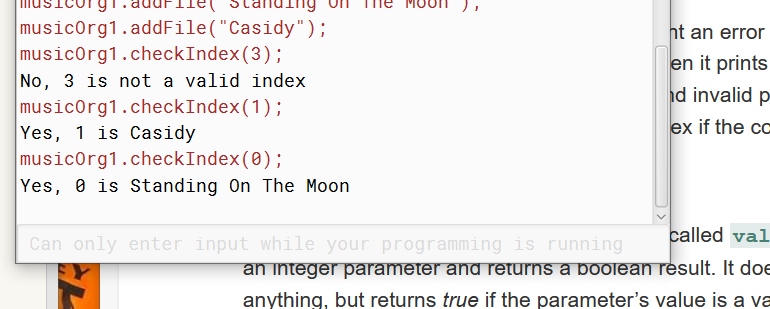
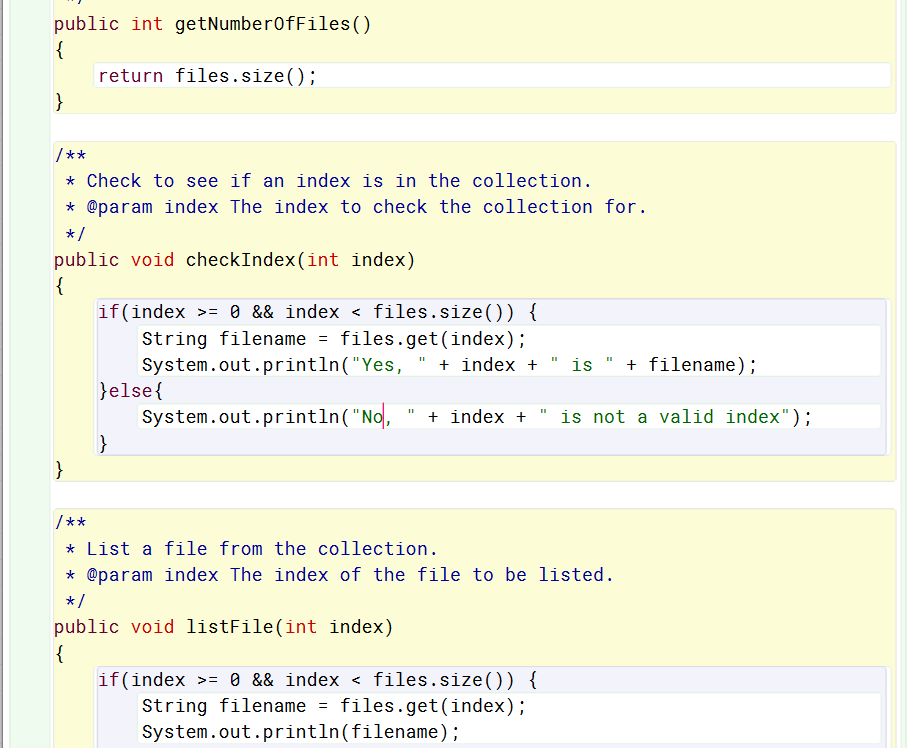
dates.remove(2);

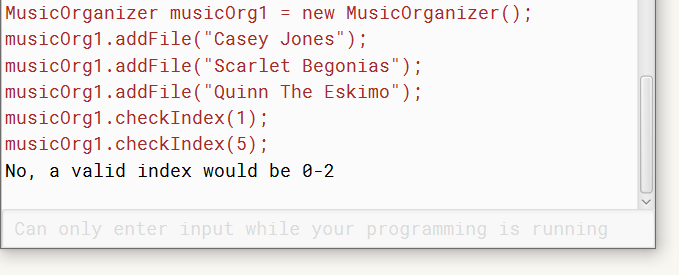
4.13

It will be stored in index 5 after removal of index 0 and 9;

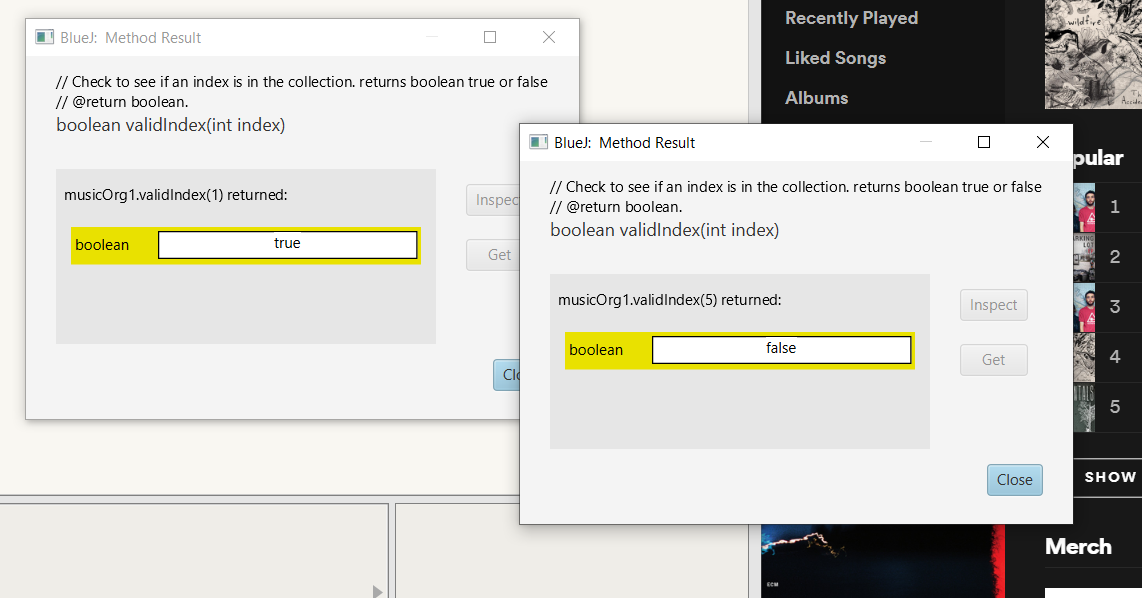
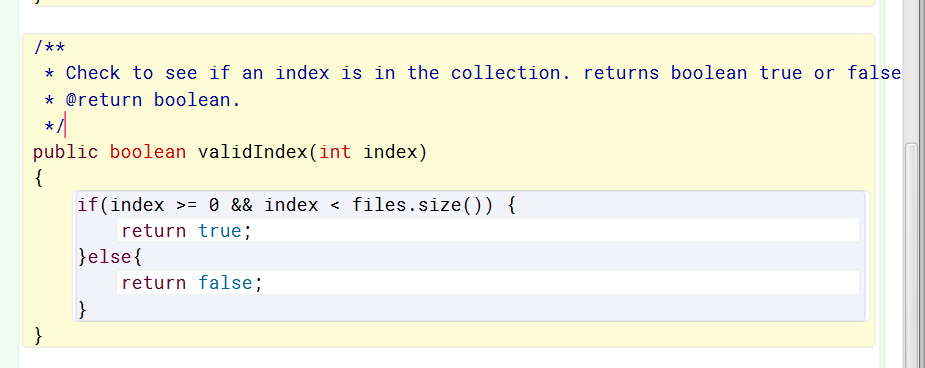
4.14

The first two pictures are what I did first. The second two pictures follow the directions more closely.

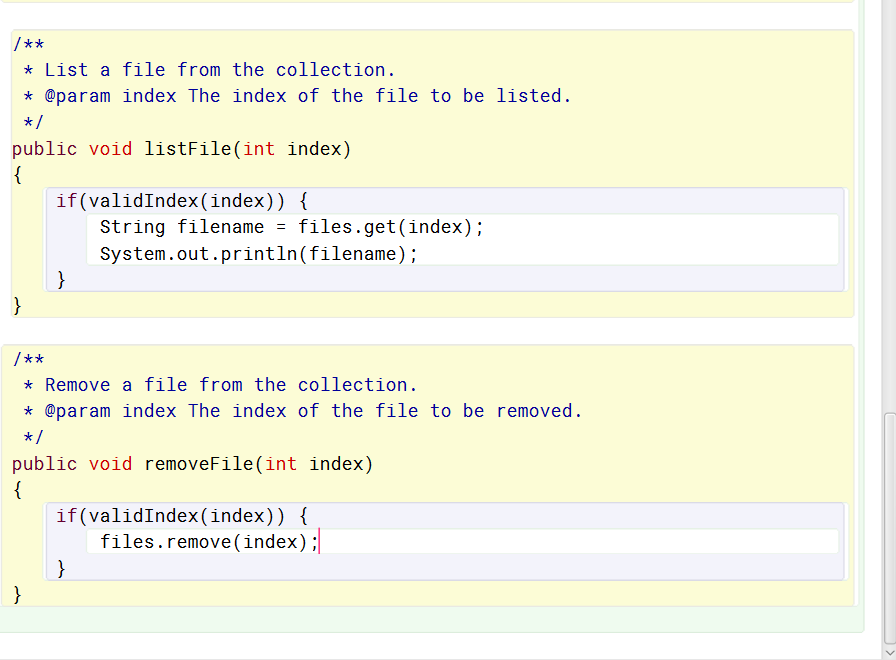




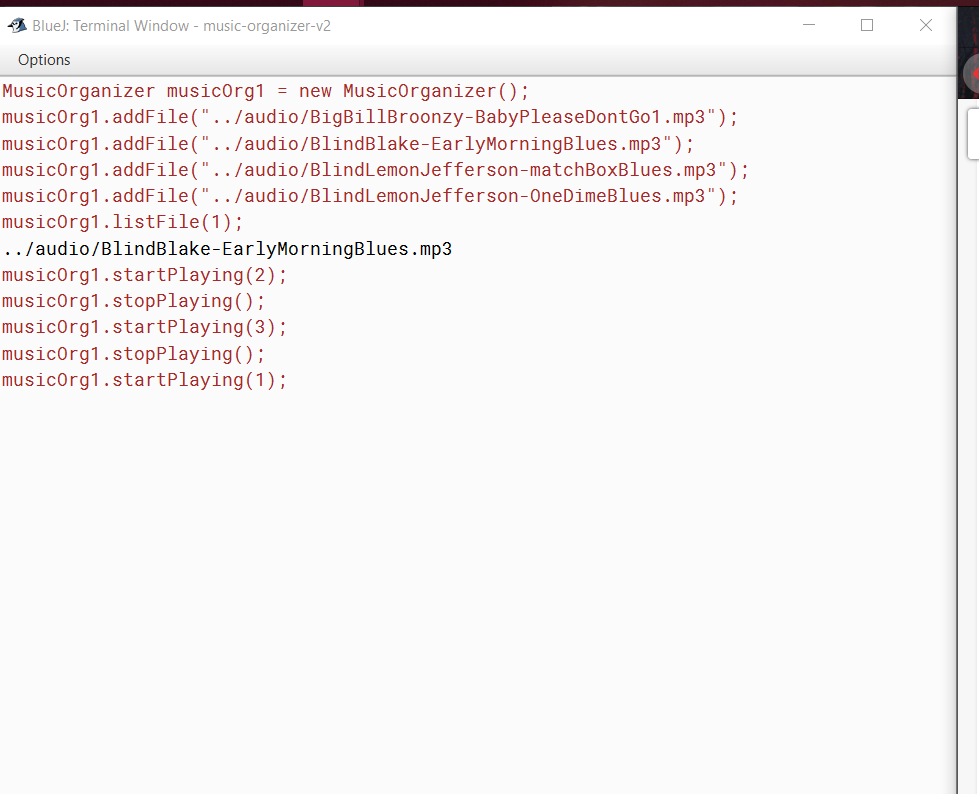
4.15



4.16



4.17



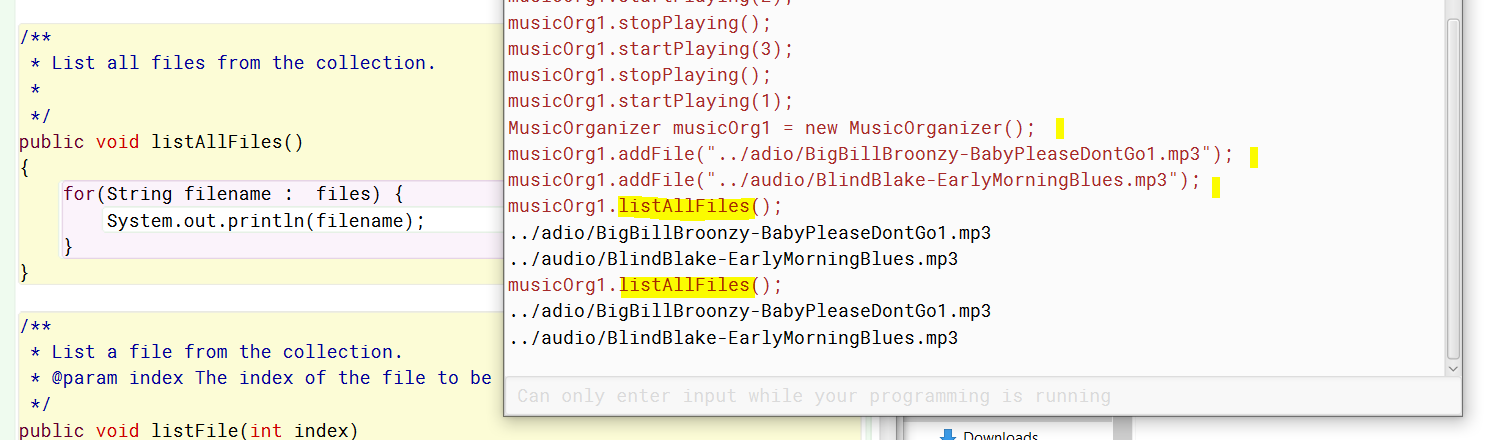
4.18

A header for a method, listAllFiles, would look like this, public void listAllFiles(). I do not think that listAllFiles will need any parameters since the files are kept in an ArrayList that is set as one of the fields of MusicOrganizerClass and accessable by our listAllFiles() method. I also do not think we need to return anything to the caller here. We will just need to print the files to the termanal no need to return anything.

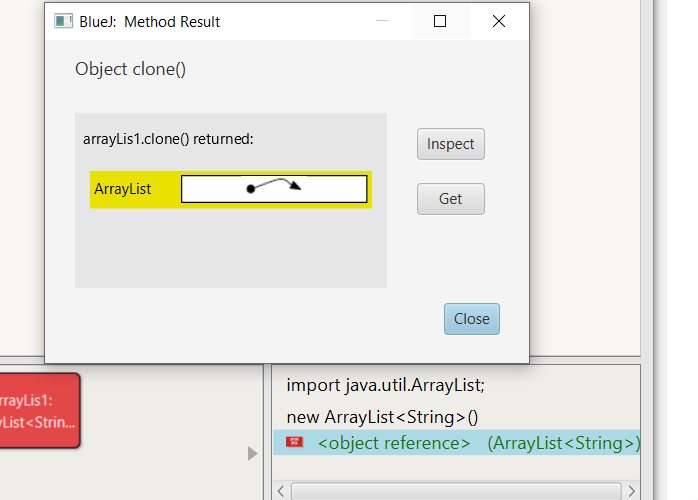
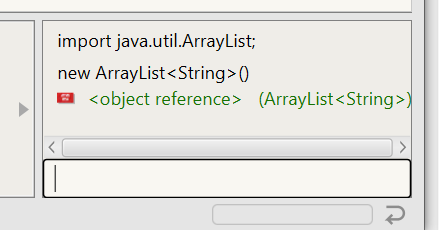
4.19

As many lines as there are files in your organizer. Just as many print statements. This is not an expandable way to do this for sure there must be a better way.

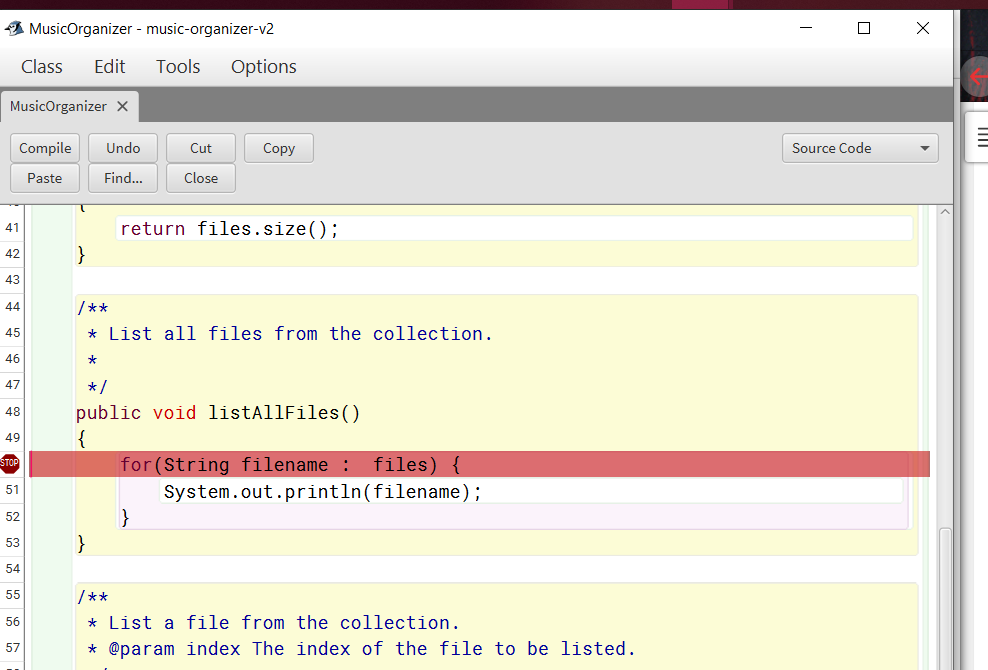
4.20/4.21

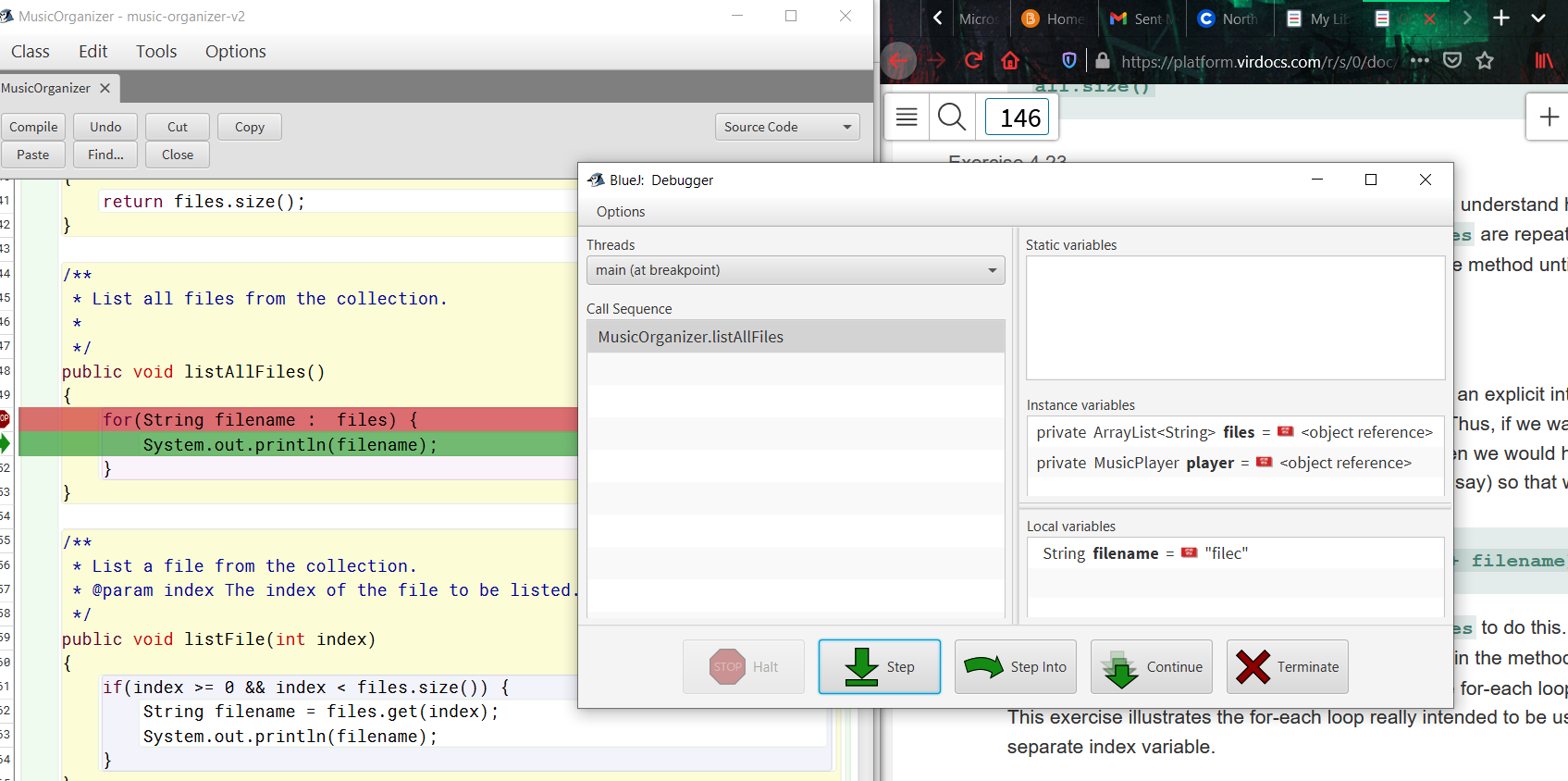


4.22

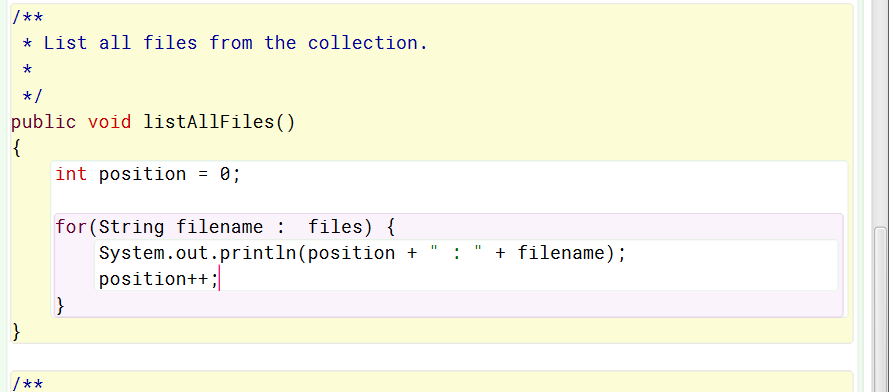


4.23

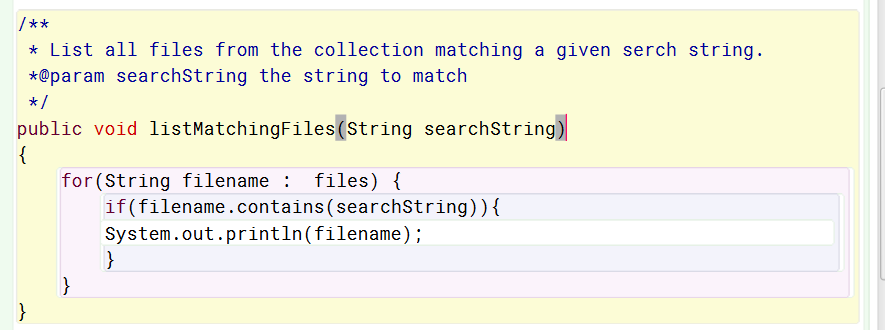


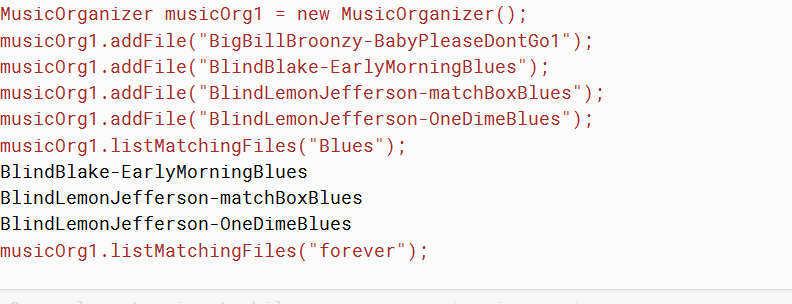


4.24



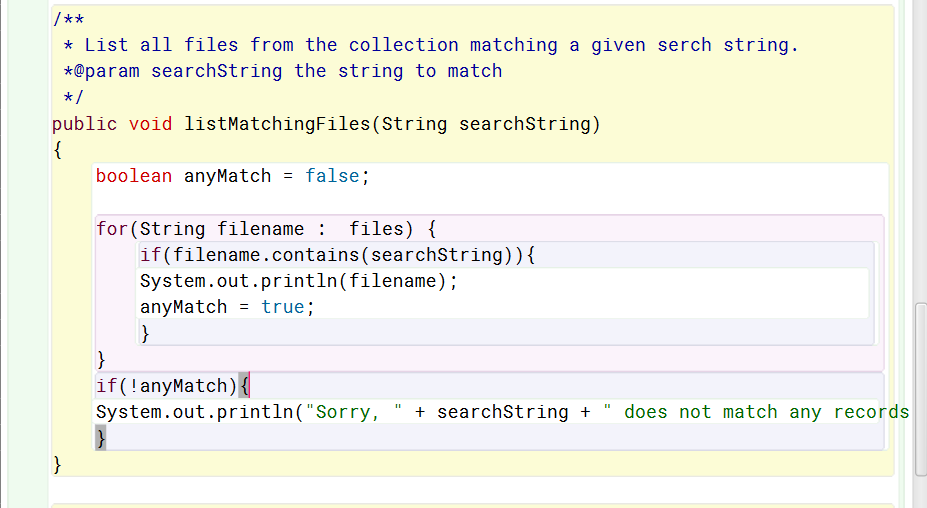
4.25

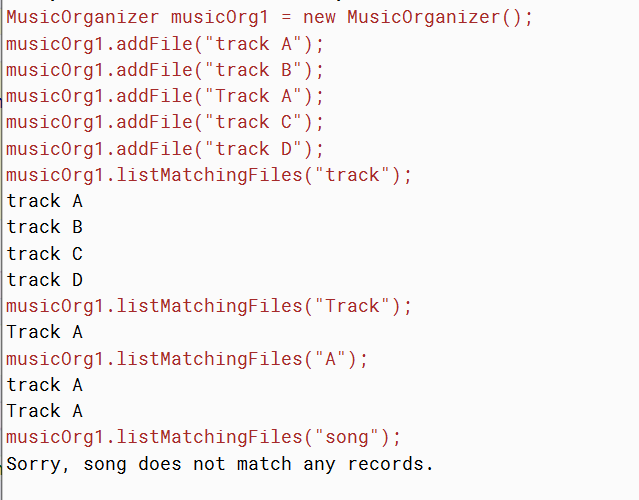




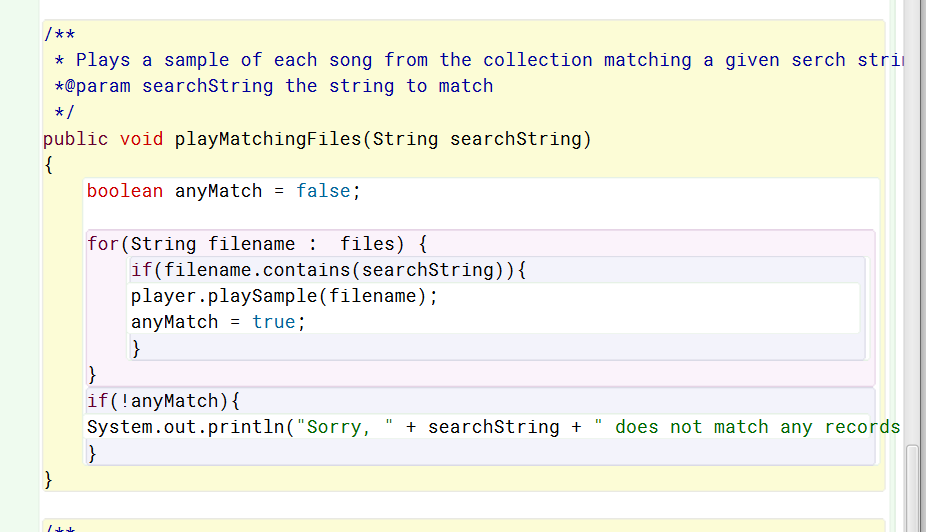
No, not if there are no matches. In that case nothing is printed.

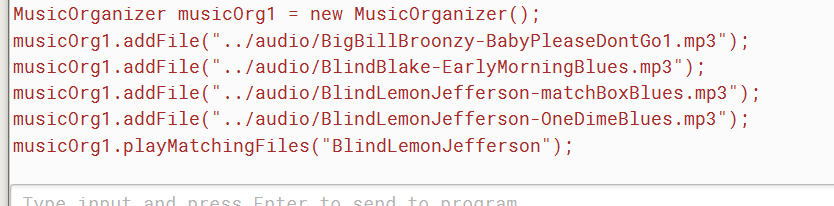
4.26





4.27





It worked!!

4.28

for(String track : tracks)

4.29

Boolean found = false

While(! found) {

if (the keys are in the next place){

found = true;

}

}

4.30

int index = 5;

while(index < 100){

System.out.Println(index);

Index = index + 5;

}

4.31

int index = 0;

Int total = 0;

while(index < 10){

total = total + index.

index++

}

System.out.Println(total);

4.32

Public int sum(int a, int b)

{

int sum = 0

while(a < b){

sum = sum + a;

a++

}

return sum;

}

4.33

public boolean isPrime(int n)

{

int index = 2;

while(index < n){

if((n % index) == 0){

return false;

}

index++

}

}

4.34

No, this value does not change. What does change is the index.

Public int findFirst(String searchString)

{

int fileSize = files.size();

int index = 0;

boolean searching = true;

While(searching && index < fileSize){

String fileName = files.get(index);

If(filename.contains(searchSring)){

Searching = false

}else{

Index++

}

}

If(searching){

return –1;

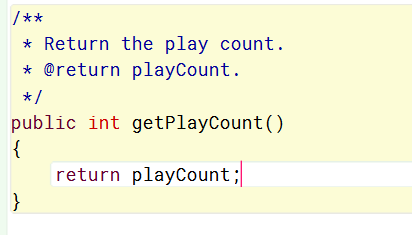
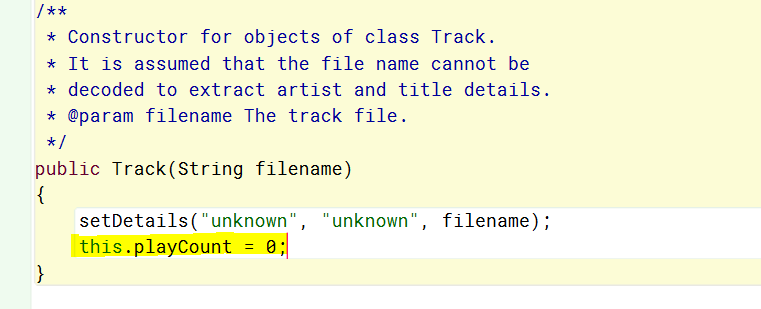
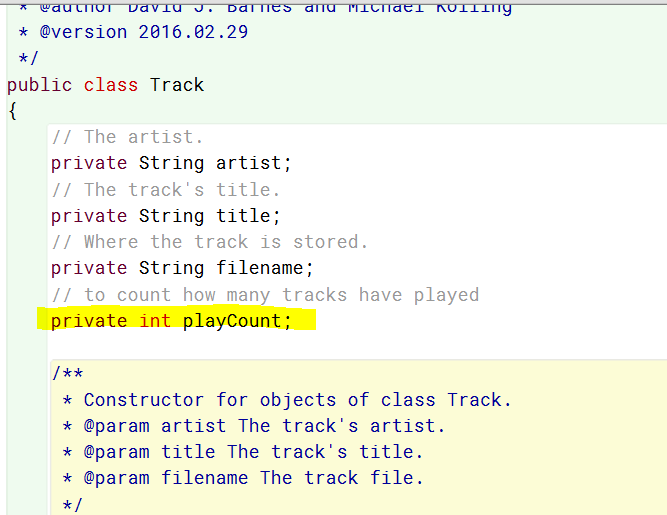
}else{

Return index;

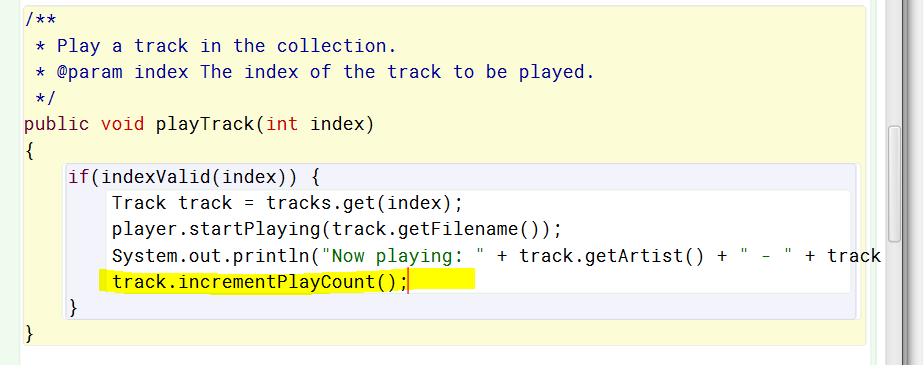
}

}

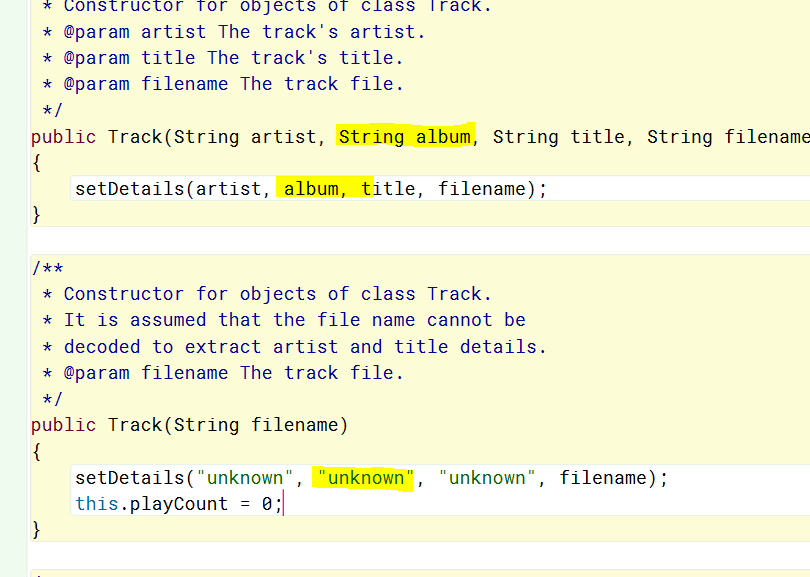
4.35

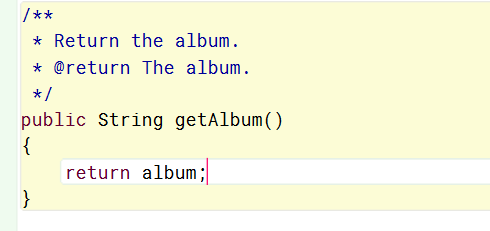


4.36

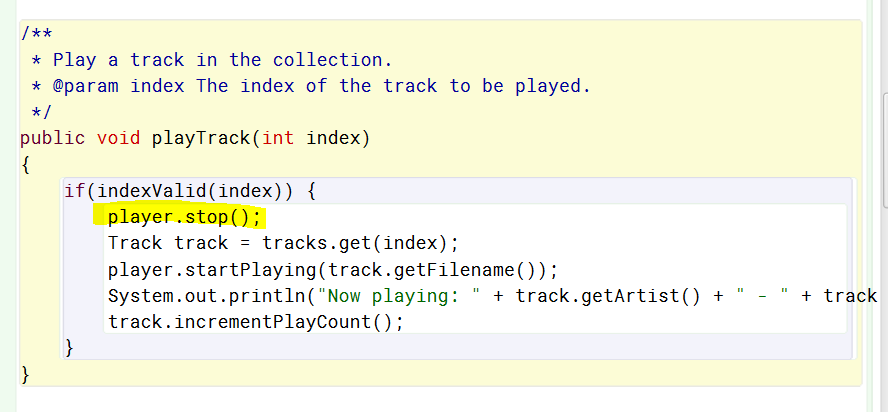


4.37

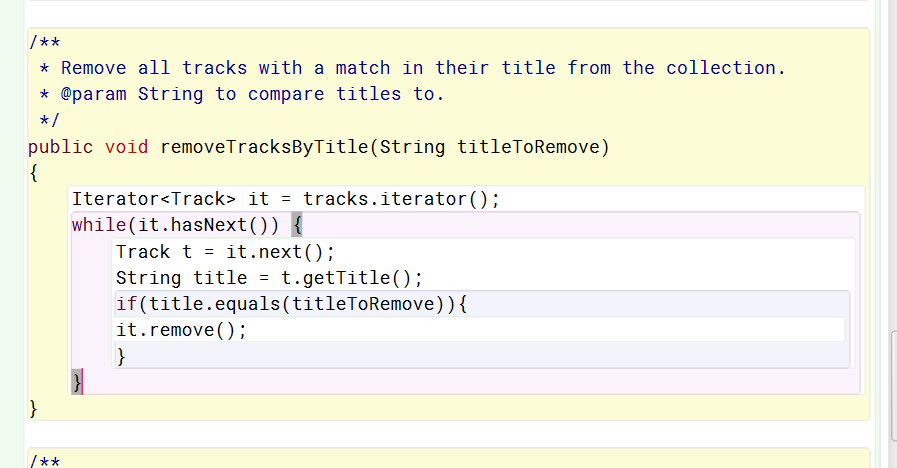




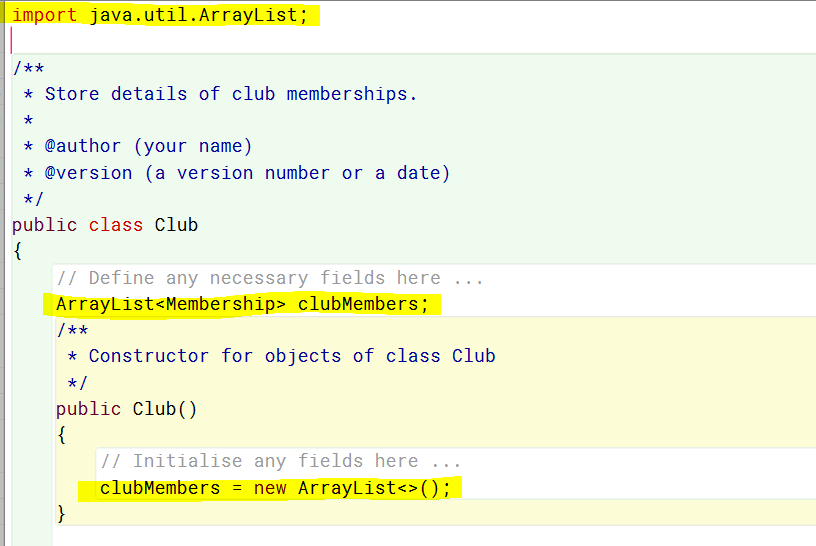
4.38



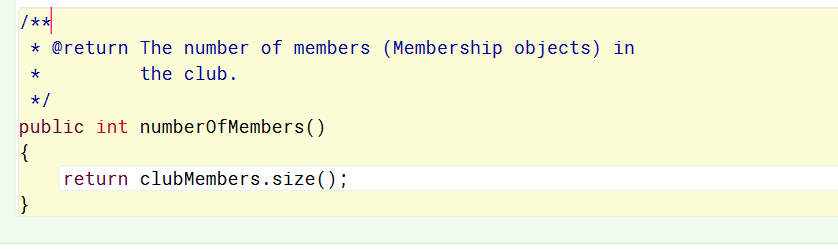
4.39



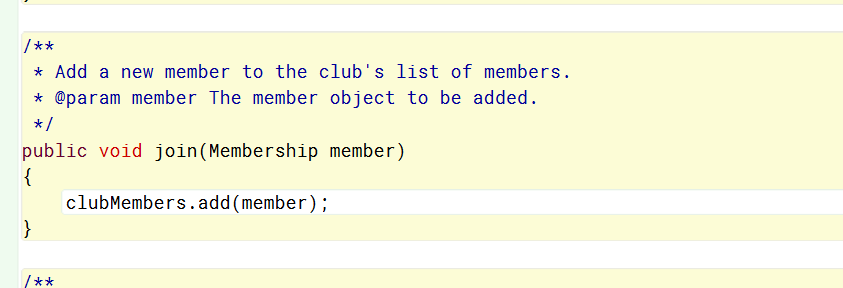
4.40



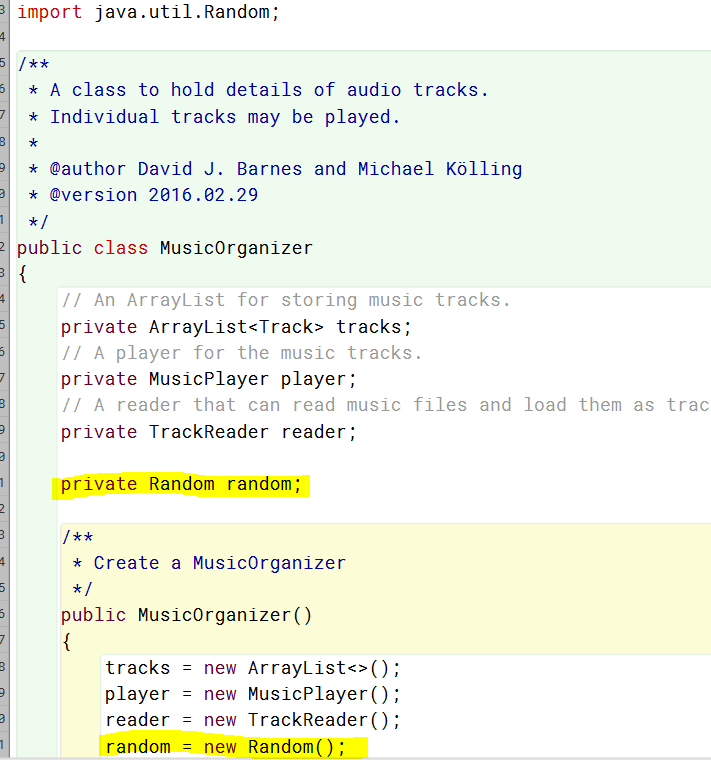
4.41

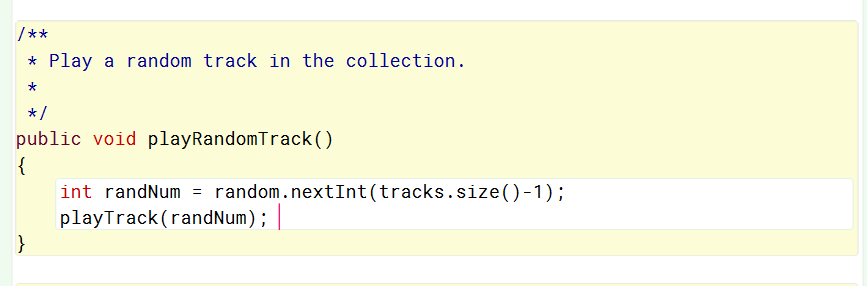


4.42



4.43





4.44

I feel like if I wanted to have my music player play a random set of tracks, I think I would like it to play through all the tracks equally. I think it would make the most sense to have a playlist of my favorite tracks or of different genres. But whichever playlist I am in, I would like it to play all the tracks equally. You could set up a shuffle method to favor certain songs if you wanted though. You could do this by having the songs you like more in the list not taken out after they play from the random list. This way they would most likely play again before the list has been emptied.

4.45