

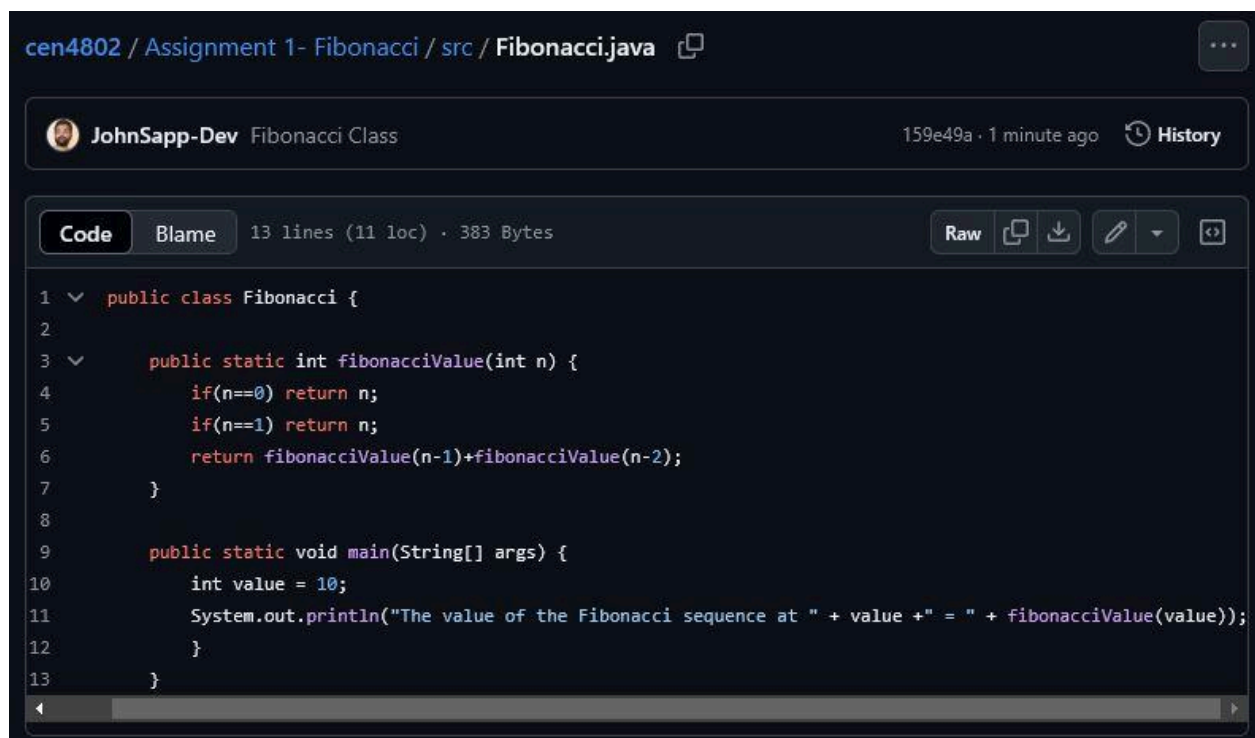
John Sapp  
1/12/25

CEN-4802c  
Assignment 1

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-jav
The value of the Fibonacci sequence at 10 = 55

Process finished with exit code 0
```

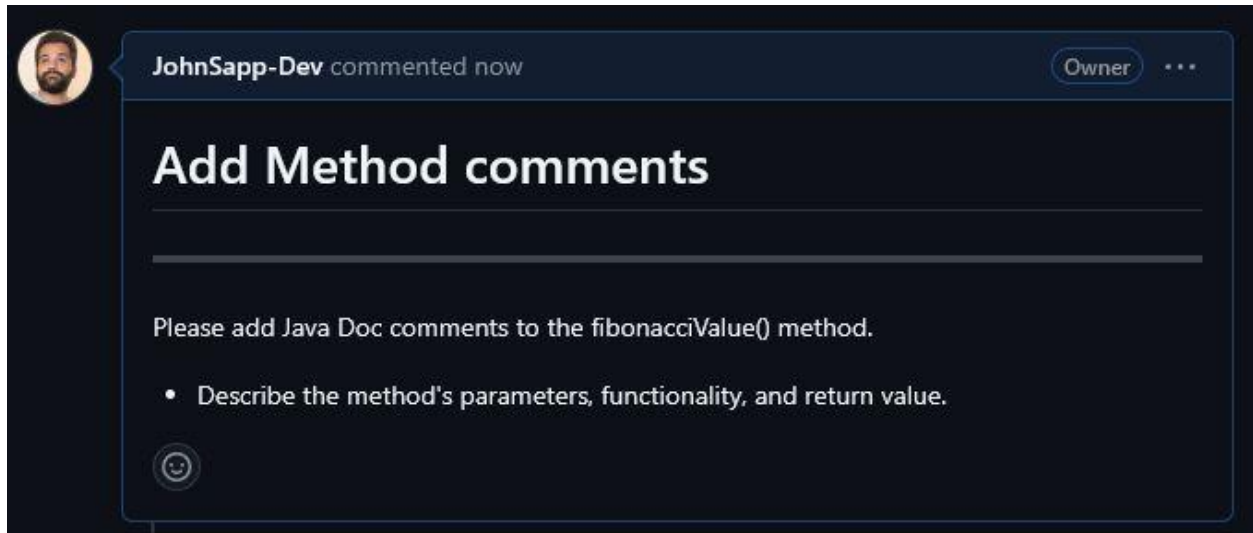
Screenshot one above shows the execution of the recursive Fibonacci sequence.



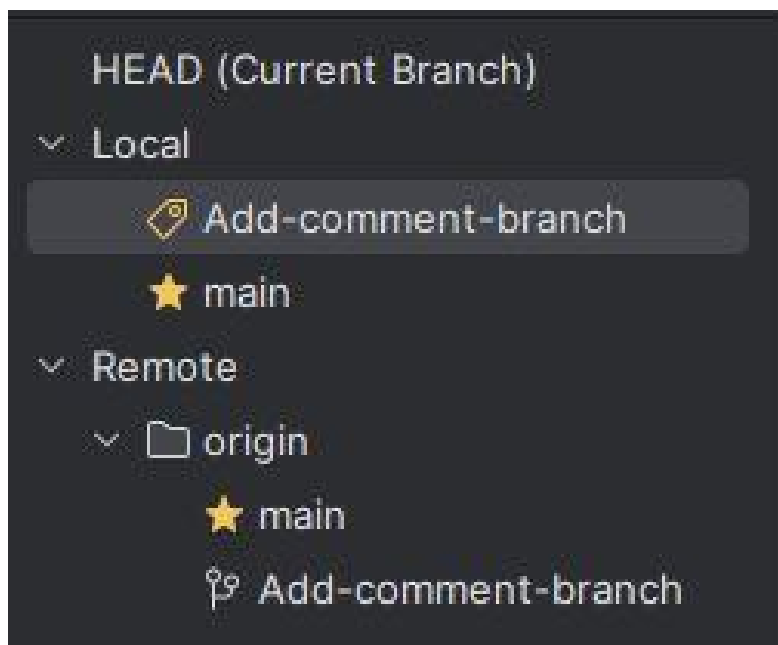
The screenshot shows a GitHub repository interface for a project named "cen4802 / Assignment 1- Fibonacci". The file "src / Fibonacci.java" is selected. The repository is owned by "JohnSapp-Dev" and was pushed 1 minute ago. The file statistics show 13 lines (11 loc) and 383 Bytes. The code is displayed in a dark-themed editor with line numbers 1 through 13. The code defines a public class "Fibonacci" with a static method "fibonacciValue" and a static method "main". The "main" method sets a value of 10 and prints the Fibonacci sequence value for that number.

```
1  public class Fibonacci {
2
3  public static int fibonacciValue(int n) {
4      if(n==0) return n;
5      if(n==1) return n;
6      return fibonacciValue(n-1)+fibonacciValue(n-2);
7  }
8
9  public static void main(String[] args) {
10     int value = 10;
11     System.out.println("The value of the Fibonacci sequence at " + value + " = " + fibonacciValue(value));
12 }
13 }
```

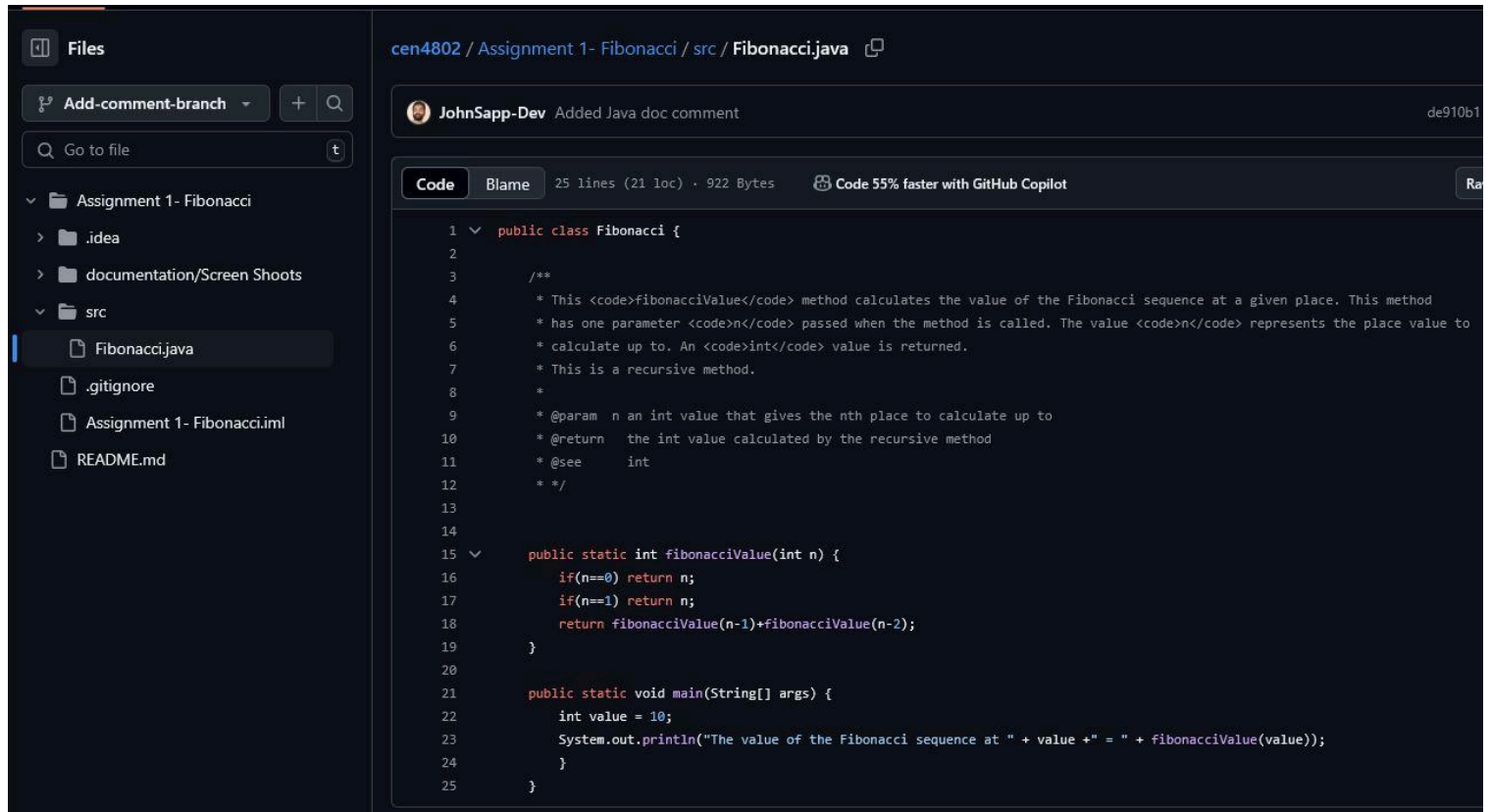
Screenshot two above shows the Fibonacci class pushed to the GitHub repo.



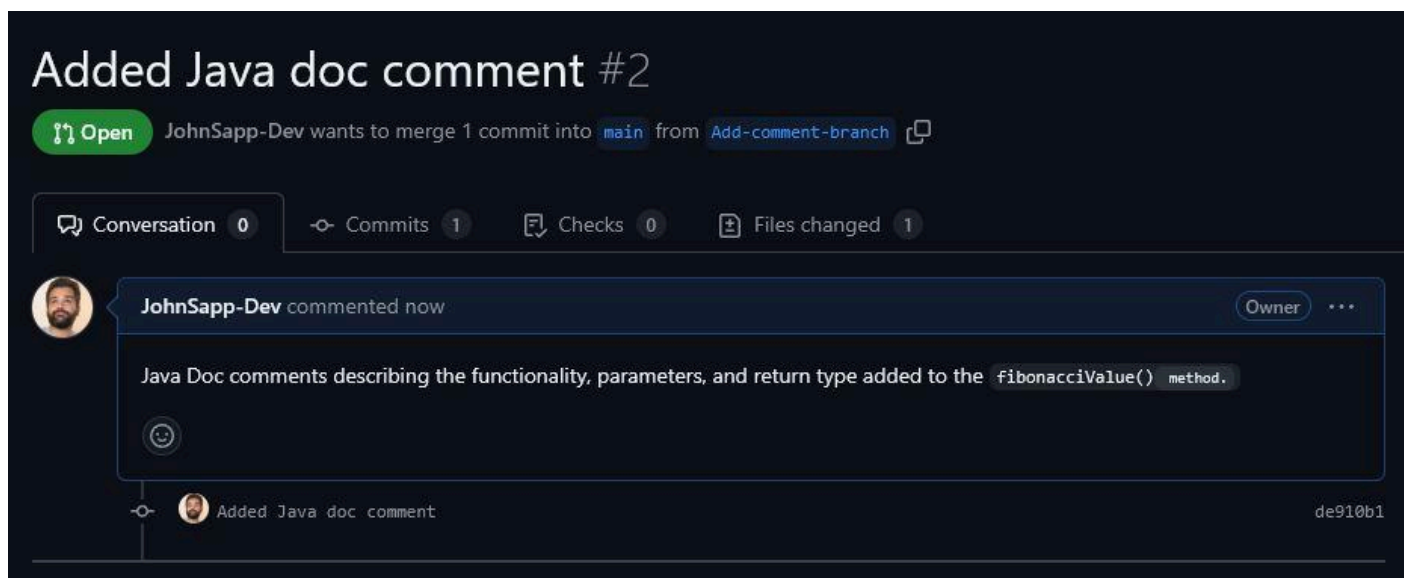
Screenshot three above shows the creation of an “issue” to add Java doc comments to the fibonacciValue() method.



Screenshot four above shows the creation of a new branch “Add-comment-branch”



Screenshot five above shows the comment has been added to a new branch named “Add-comment-branch” and has been pushed to GitHub.

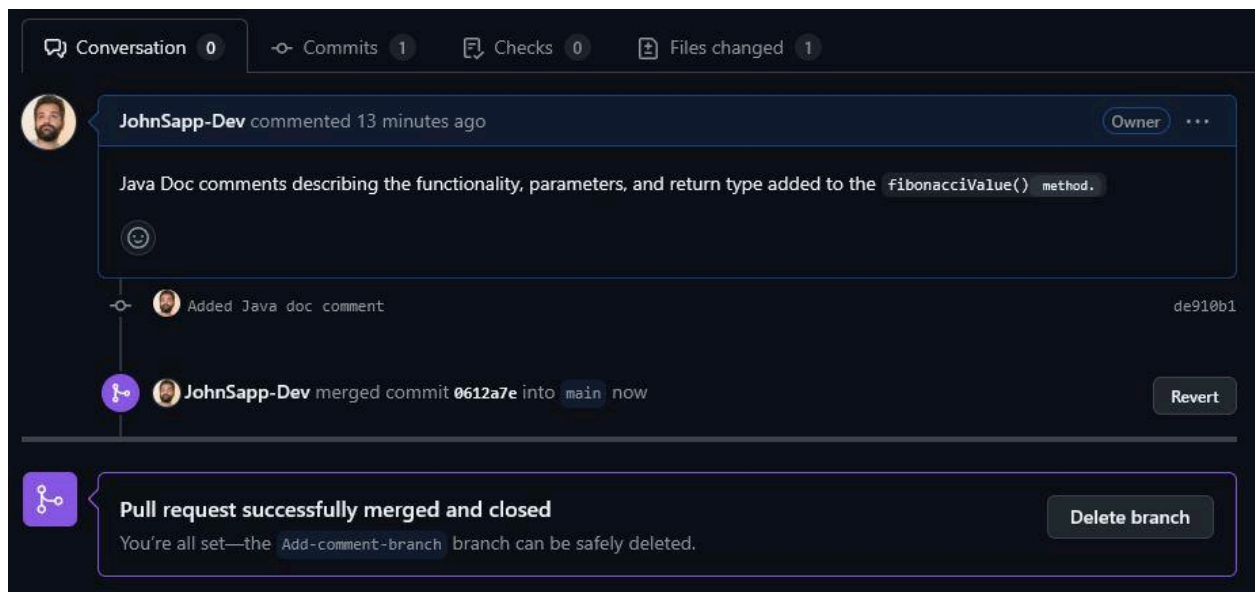


Screenshot six above shows a pull request requesting to add the commit from the “Add-comment-branch” branch to the “main” branch.

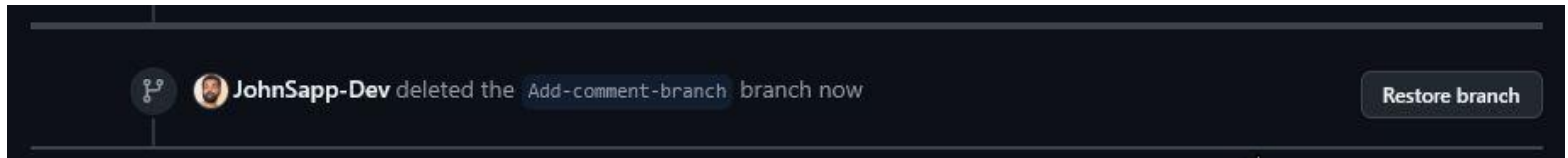
```
1 public class Fibonacci {
2
3     public static int fibonacciValue(int n) {
4         if(n==0) return n;
5         if(n==1) return n;
6         return fibonacciValue(n-1)+fibonacciValue(n-2);
7     }
8
9     public static void main(String[] args) {
10         int value = 10;
11         System.out.println("The value of the Fibonacci sequence at " + value + " = " +
12             fibonacciValue(value));
13     }
14 }
```

```
1 public class Fibonacci {
2
3     /**
4      * This <code>fibonacciValue</code> method calculates the value of the Fibonacci sequence at
5      * a given place. This method
6      * has one parameter <code>n</code> passed when the method is called. The value
7      * <code>n</code> represents the place value to
8      * calculate up to. An <code>int</code> value is returned.
9      * This is a recursive method.
10     *
11     * @param n an int value that gives the nth place to calculate up to
12     * @return the int value calculated by the recursive method
13     * @see int
14     */
15     public static int fibonacciValue(int n) {
16         if(n==0) return n;
17         if(n==1) return n;
18         return fibonacciValue(n-1)+fibonacciValue(n-2);
19     }
20
21     public static void main(String[] args) {
22         int value = 10;
23         System.out.println("The value of the Fibonacci sequence at " + value + " = " +
24             fibonacciValue(value));
25     }
26 }
```

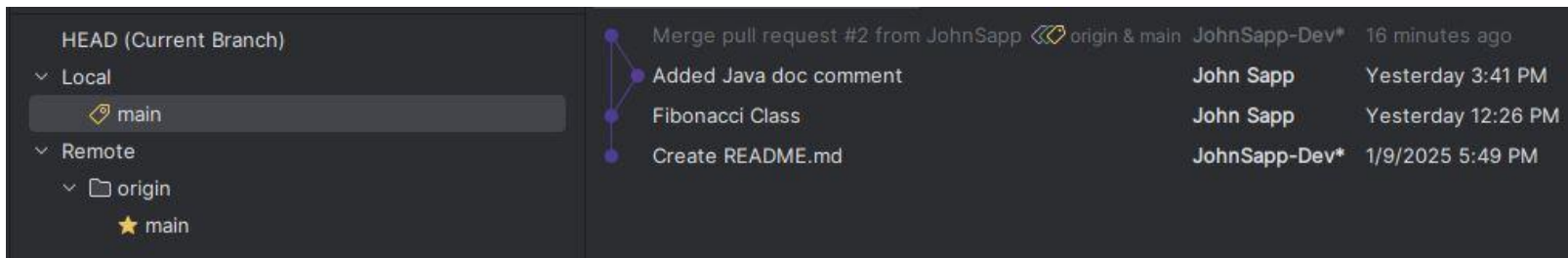
Screenshot seven above shows the comparison between the “main” branch Fibonacci.java file on the left and the “Add-comment-branch” branch Fibonacci.java file on the right. The file on the right has new comments to be added.



Screenshot eight above shows the pull request has been successfully merged and closed.



Screenshot nine above shows that the “Add-comment-branch” branch has been deleted.



Screenshot ten above shows the local repo matches the remote repo.

```
/**  
 * This fibonacciValue method calculates the value of the Fibonacci sequence  
at a given place. This method  
 * has one parameter n passed when the method is called. The value  
n represents the place value to  
 * calculate up to. An int value is returned.  
 * This is a recursive method.  
 *  
 * @param n an int value that gives the nth place to calculate up to
```

```
* @return the int value calculated by the recursive method
* @see    int
**/
```