Lab04: Skills - Using a Debugger

Getting Ready: Before going any further, you should:

1. Setup your development environment.

2. Download the following files:
   `PhoneDriver.java`
   `PhoneCard.java`
   to an appropriate directory/folder. (In most browsers/OSs, the easiest way to do this is by right-clicking/control-clicking on each of the links above.)

3. If you don't already have one from earlier in the semester, create a project named eclipseskills.

4. Drag the file `PhoneCard.java` and `PhoneDriver.java` into the default package (using the "Copy files" option).

5. Open `PhoneCard.java` and `PhoneDriver.java`.

Part 1. Review: This part of the lab will review a few topics related to object-oriented programming in Java.

1. In the `main()` method in the `PhoneDriver` class, what kind of objects are end, now, and start?

   They are all Date objects.

2. In the `main()` method in the `PhoneDriver` class, what kind of object is card?

   Card is a PhoneCard object.

3. Where is the code for the `PhoneCard` class?

The code for the PhoneCard class is within PhoneCard.java

4. Where is the code for the `Date` class?

The code comes from java.util.Date

5. Read the documentation for the `Date` (https://docs.oracle.com/javase/7/docs/api/java/util/Date.html). Make sure you find the documentation for the `Date` class that is in `java.util`. (There are several `Date` classes in the Java library.)

6. When you construct a `Date` object using the default constructor (i.e., the constructor that has no parameters), what properties will it have?

   "Allocates a `Date` object and initializes it so that it represents the time at which it was allocated, measured to the nearest millisecond."

7. When you construct a `Date` object using the default constructor (i.e., the constructor that has no parameters), what properties will it have?

   Allocates a `Date` object and initializes it so that it represents the time at which it was allocated, measured to the nearest millisecond.

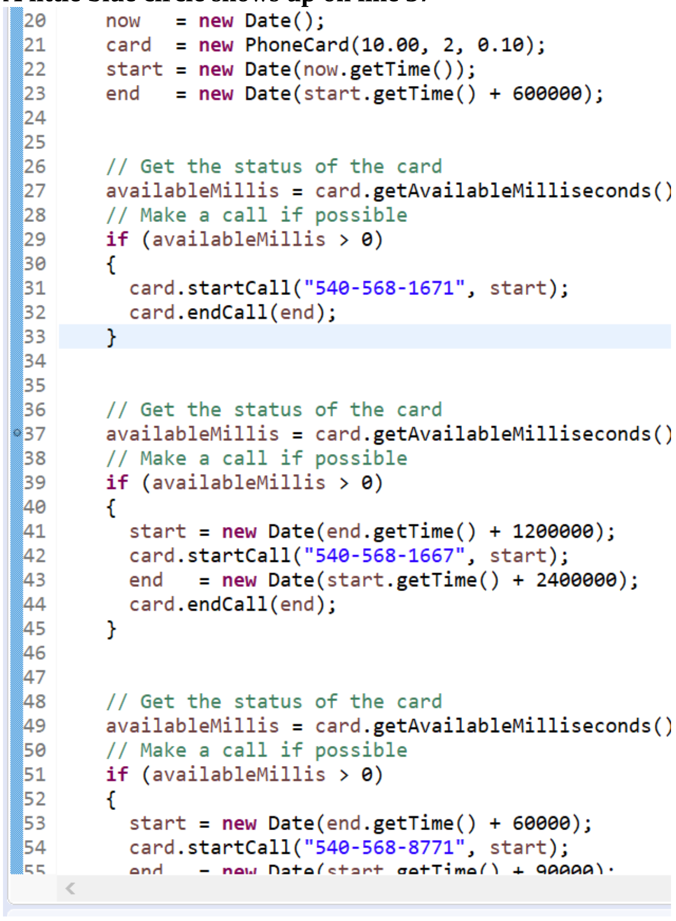
Part 2. Setting a Breakpoint: One of the nice things about running an application in a debugger is that you can stop the execution at one or more pre-defined locations (called *breakpoints*). This part of the lab will teach you how.

1. Click on the tab containing `PhoneDriver.java` to make sure that it has the focus.

2. Right-click in line 33 of `PhoneDriver.java` and pull down to Toggle Breakpoint.

3. What happened?

A little blue circle shows up on line 37



```
20      now   = new Date();
21      card  = new PhoneCard(10.00, 2, 0.10);
22      start = new Date(now.getTime());
23      end   = new Date(start.getTime() + 600000);
24
25
26      // Get the status of the card
27      availableMillis = card.getAvailableMilliseconds()
28      // Make a call if possible
29      if (availableMillis > 0)
30      {
31        card.startCall("540-568-1671", start);
32        card.endCall(end);
33      }
34
35
36      // Get the status of the card
37      availableMillis = card.getAvailableMilliseconds()
38      // Make a call if possible
39      if (availableMillis > 0)
40      {
41        start = new Date(end.getTime() + 1200000);
42        card.startCall("540-568-1667", start);
43        end   = new Date(start.getTime() + 2400000);
44        card.endCall(end);
45      }
46
47
48      // Get the status of the card
49      availableMillis = card.getAvailableMilliseconds()
50      // Make a call if possible
51      if (availableMillis > 0)
52      {
53        start = new Date(end.getTime() + 60000);
54        card.startCall("540-568-8771", start);
55        end   = new Date(start.getTime() + 90000);
```

4. Click on . This will run `PhoneDriver` and stop the execution at the breakpoint (i.e., line 33). Note: If prompted, allow Eclipse to enter the "Debug Perspective".

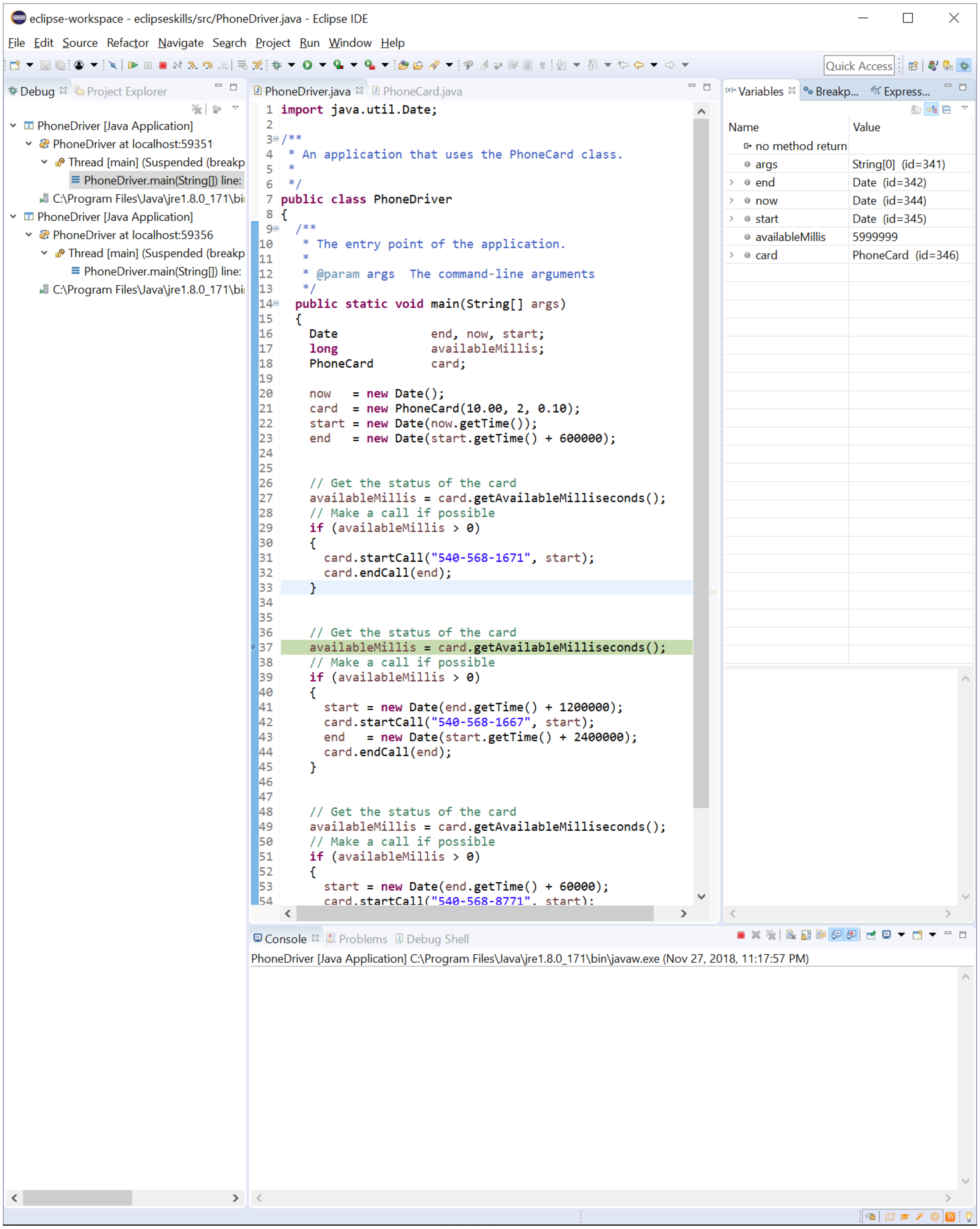
5. What happened?

   The debug perspective opened up and line 37 is now highlighted in green. Everytime I toggle breakpoint in line 33, it toggles line 37 instead.

Part 3. Checking State Information: Another nice thing about running an application in a debugger is that, once you stop the execution at a breakpoint, you can check state information (e.g., the value of attributes and variables). This part of the lab will teach you how.

1. Click on the "Variables" tab on the left side of the debug window.
2. Click on the "tree icon" next to "Locals" to expand it.

   I don’t see Locals


eclipse-workspace - eclipseskills/src/PhoneDriver.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
Project Explorer
PhoneDriver.java
PhoneCard.java
Variables
Breakp...
Express...
Console
Problems
Debug Shell

```
PhoneDriver [Java Application]
  PhoneDriver at localhost:59351
    Thread [main] (Suspended (breakp
      PhoneDriver.main(String[]) line:
  C:\Program Files\Java\jre1.8.0_171\bi
PhoneDriver [Java Application]
  PhoneDriver at localhost:59356
    Thread [main] (Suspended (breakp
      PhoneDriver.main(String[]) line:
  C:\Program Files\Java\jre1.8.0_171\bi
```

```java
import java.util.Date;

/**
 * An application that uses the PhoneCard class.
 *
 */
public class PhoneDriver
{
  /**
   * The entry point of the application.
   *
   * @param args  The command-line arguments
   */
  public static void main(String[] args)
  {
    Date             end, now, start;
    long             availableMillis;
    PhoneCard        card;

    now   = new Date();
    card  = new PhoneCard(10.00, 2, 0.10);
    start = new Date(now.getTime());
    end   = new Date(start.getTime() + 600000);


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      card.startCall("540-568-1671", start);
      card.endCall(end);
    }


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      start = new Date(end.getTime() + 1200000);
      card.startCall("540-568-1667", start);
      end   = new Date(start.getTime() + 2400000);
      card.endCall(end);
    }


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      start = new Date(end.getTime() + 60000);
      card.startCall("540-568-8771", start);
```

Name	Value
no method return	
args	String[0] (id=341)
end	Date (id=342)
now	Date (id=344)
start	Date (id=345)
availableMillis	5999999
card	PhoneCard (id=346)

```
PhoneDriver [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:17:57 PM)
```

3. What is the current value of `availableMillis`?

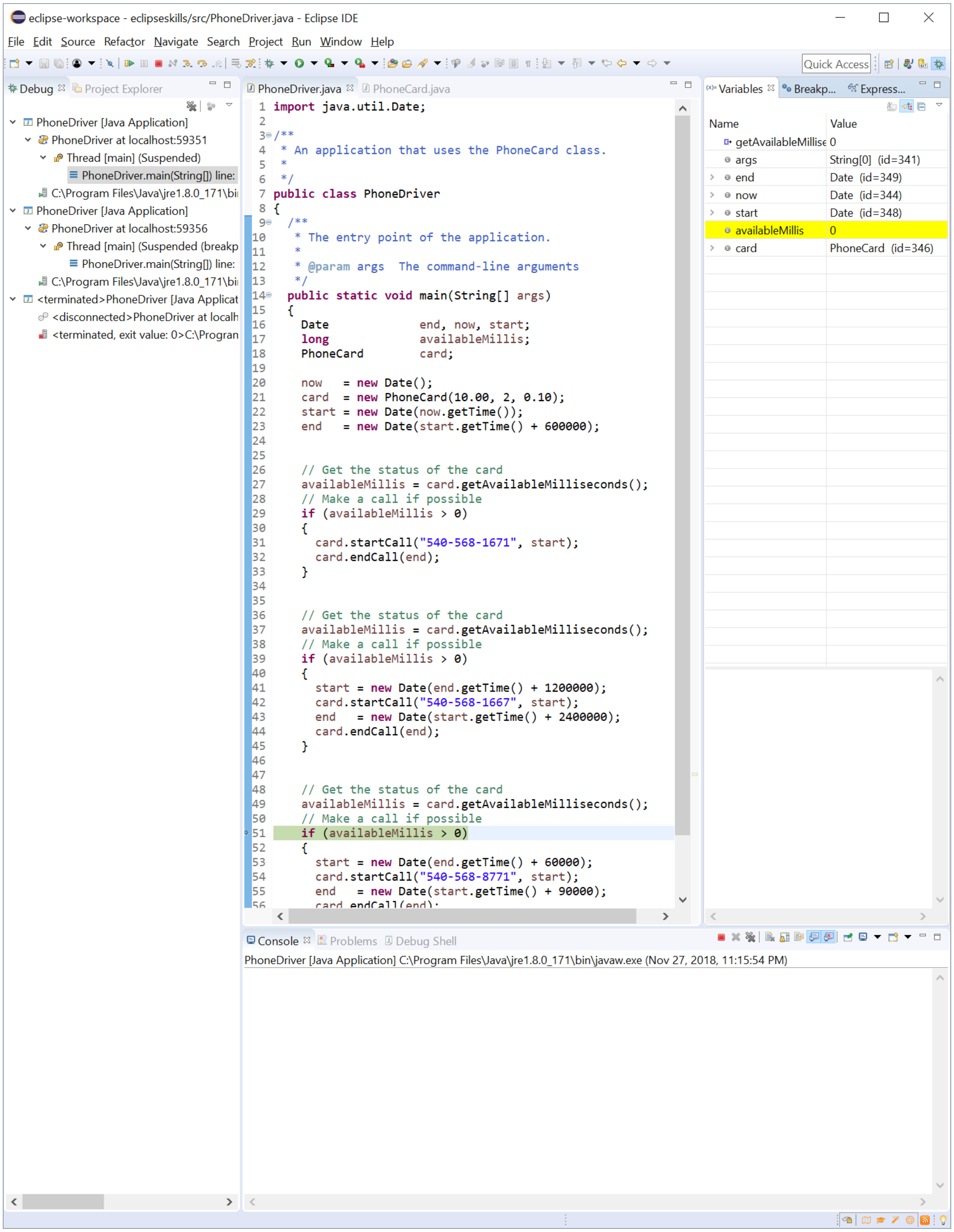
The current value of availableMillis is 5999999

Part 4. Stepping Over Lines: When running an application in a debugger, once you stop the execution at a breakpoint, you can continue the execution one "step" at a time. This part of the lab will teach you how.

1. Click on 🐞. This will run `PhoneDriver` again and stop the execution at the breakpoint (i.e., line 33).
2. Click on the ↷ button.
3. What happened?

availableMillis turned yellow and is now showing 5399999

4. Click on the ↷ button until the next if statement is highlighted.


eclipse-workspace - eclipseskills/src/PhoneDriver.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
Project Explorer
PhoneDriver.java
PhoneCard.java
Variables
Breakp...
Express...
Console
Problems
Debug Shell

```
PhoneDriver [Java Application]
  PhoneDriver at localhost:59351
    Thread [main] (Suspended)
      PhoneDriver.main(String[]) line:
  C:\Program Files\Java\jre1.8.0_171\bi
PhoneDriver [Java Application]
  PhoneDriver at localhost:59356
    Thread [main] (Suspended (breakp
      PhoneDriver.main(String[]) line:
  C:\Program Files\Java\jre1.8.0_171\bi
<terminated>PhoneDriver [Java Applicat
  <disconnected>PhoneDriver at localh
  <terminated, exit value: 0>C:\Progran
```

```java
import java.util.Date;

/**
 * An application that uses the PhoneCard class.
 *
 */
public class PhoneDriver
{
  /**
   * The entry point of the application.
   *
   * @param args  The command-line arguments
   */
  public static void main(String[] args)
  {
    Date            end, now, start;
    long            availableMillis;
    PhoneCard       card;

    now   = new Date();
    card  = new PhoneCard(10.00, 2, 0.10);
    start = new Date(now.getTime());
    end   = new Date(start.getTime() + 600000);


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      card.startCall("540-568-1671", start);
      card.endCall(end);
    }


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      start = new Date(end.getTime() + 1200000);
      card.startCall("540-568-1667", start);
      end   = new Date(start.getTime() + 2400000);
      card.endCall(end);
    }


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      start = new Date(end.getTime() + 60000);
      card.startCall("540-568-8771", start);
      end   = new Date(start.getTime() + 90000);
      card.endCall(end);
```

Name	Value
getAvailableMillise	0
args	String[0] (id=341)
end	Date (id=349)
now	Date (id=344)
start	Date (id=348)
availableMillis	0
card	PhoneCard (id=346)

```
PhoneDriver [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:15:54 PM)
```

5. What is the current value of `availableMillis`? (Hint: Look in the "Variables" tab. You may beed to scroll.)

   The current value of availableMillis is 0.

6. Click on the button to run to the end of the application.

Part 5: Stepping Into Lines: So far, all of the "stepping" you have done has been in one method in one class. This is called "stepping over". You can also "step into" a line of code to see what happens there. This part of the lab will teach you how.

1. Click on . This will run `PhoneDriver` and stop the execution at the breakpoint (i.e., line 33).
2. Click on the button.
3. What happened?

   It jumped to line 70.

4. Click on the button again.
5. What happened?

It jumped to the if statement in line 71

6. Look at the call stack in the "Debug" tab. It tells you what class and method you are in and where this method was called from.

7. What method is currently being executed (and what class is it in)?

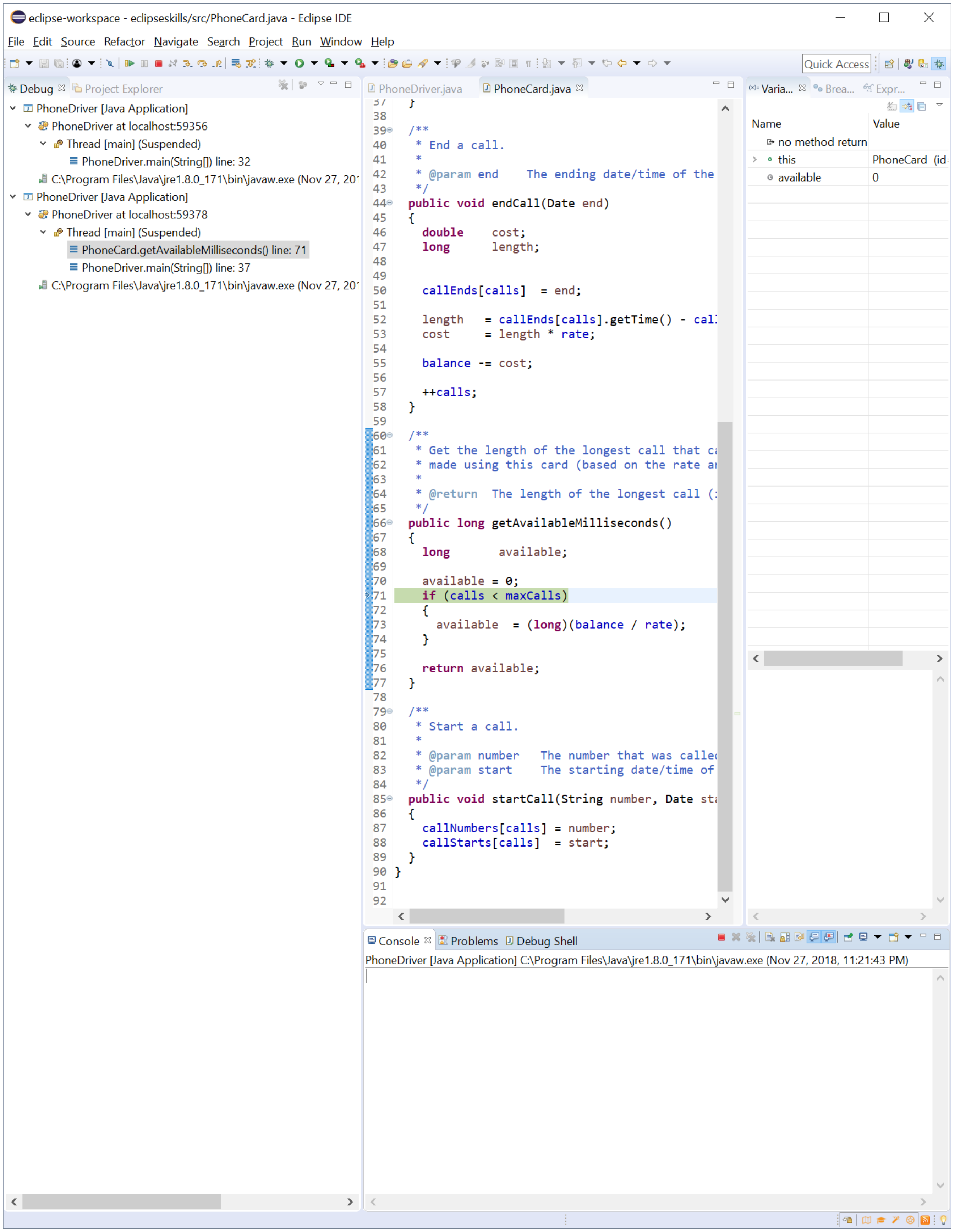
   getAvailableMilliseconds is being executed from Class PhoneCard

8. What line is currently being executed?

   Line 71 is currently being executed

9. Where was this method called from?

   This method was called from main


eclipse-workspace - eclipseskills/src/PhoneCard.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
Project Explorer
PhoneDriver.java
PhoneCard.java
Varia...
Brea...
Expr...
Console
Problems
Debug Shell

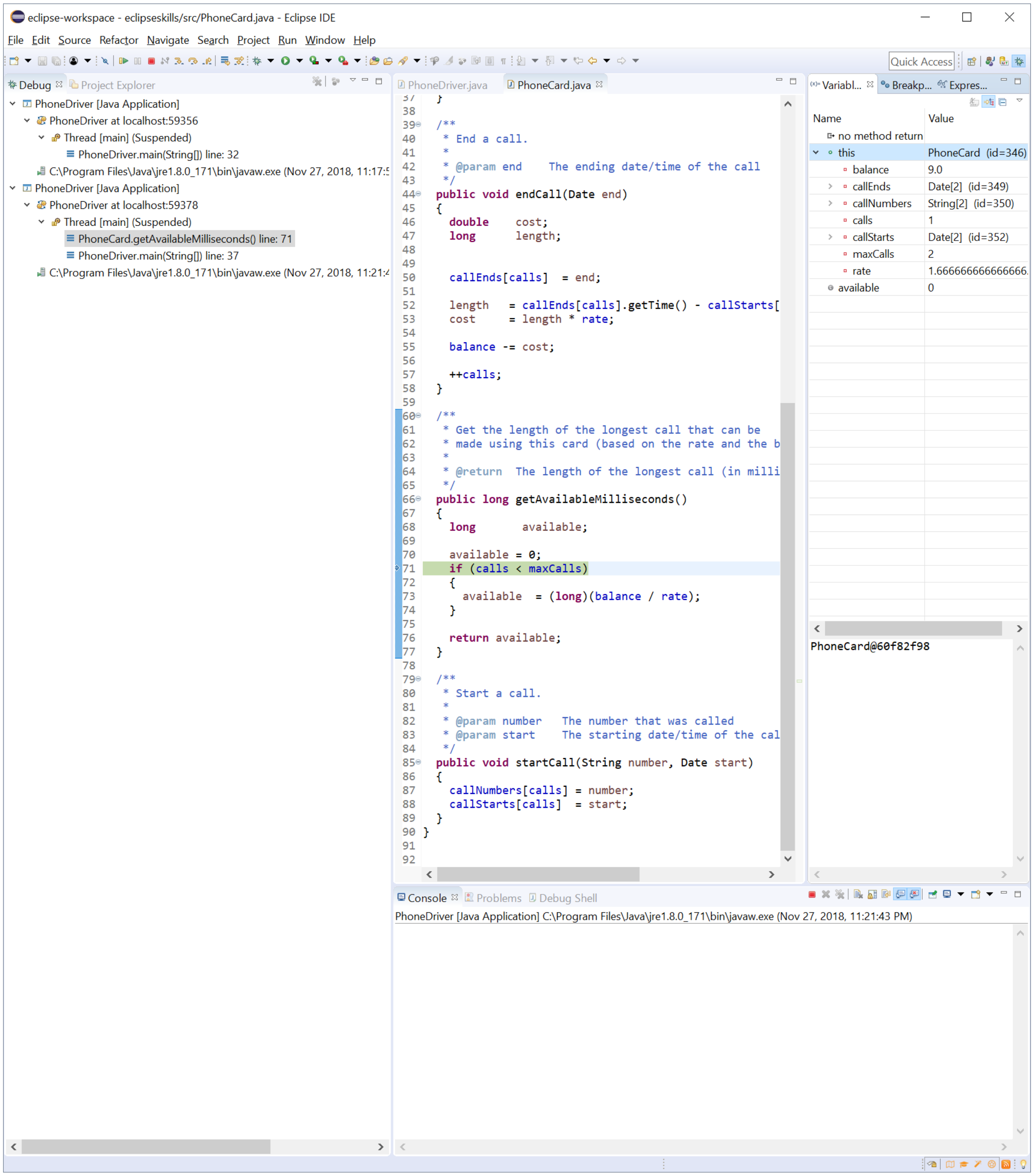
```
PhoneDriver [Java Application]
  PhoneDriver at localhost:59356
    Thread [main] (Suspended)
      PhoneDriver.main(String[]) line: 32
  C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 201
PhoneDriver [Java Application]
  PhoneDriver at localhost:59378
    Thread [main] (Suspended)
      PhoneCard.getAvailableMilliseconds() line: 71
      PhoneDriver.main(String[]) line: 37
  C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 201
```

```
37      }
38
39      /**
40       * End a call.
41       *
42       * @param end     The ending date/time of the
43       */
44      public void endCall(Date end)
45      {
46        double     cost;
47        long       length;
48
49
50        callEnds[calls]  = end;
51
52        length    = callEnds[calls].getTime() - cal
53        cost      = length * rate;
54
55        balance -= cost;
56
57        ++calls;
58      }
59
60      /**
61       * Get the length of the longest call that ca
62       * made using this card (based on the rate ar
63       *
64       * @return  The length of the longest call (
65       */
66      public long getAvailableMilliseconds()
67      {
68        long        available;
69
70        available = 0;
71        if (calls < maxCalls)
72        {
73          available  = (long)(balance / rate);
74        }
75
76        return available;
77      }
78
79      /**
80       * Start a call.
81       *
82       * @param number   The number that was callec
83       * @param start    The starting date/time of
84       */
85      public void startCall(String number, Date sta
86      {
87        callNumbers[calls] = number;
88        callStarts[calls]  = start;
89      }
90    }
91
92
```

Name	Value
no method return	
this	PhoneCard (id:
available	0

```
PhoneDriver [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:21:43 PM)
```

10. Click on the "triangle icon" next to this to expand it.



eclipse-workspace - eclipseskills/src/PhoneCard.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug | Project Explorer

```
PhoneDriver [Java Application]
  PhoneDriver at localhost:59356
    Thread [main] (Suspended)
      PhoneDriver.main(String[]) line: 32
  C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:17:5
PhoneDriver [Java Application]
  PhoneDriver at localhost:59378
    Thread [main] (Suspended)
      PhoneCard.getAvailableMilliseconds() line: 71
      PhoneDriver.main(String[]) line: 37
  C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:21:4
```

PhoneDriver.java | PhoneCard.java

```java
37    }
38
39    /**
40     * End a call.
41     *
42     * @param end    The ending date/time of the call
43     */
44    public void endCall(Date end)
45    {
46      double    cost;
47      long      length;
48
49
50      callEnds[calls]  = end;
51
52      length   = callEnds[calls].getTime() - callStarts[
53      cost     = length * rate;
54
55      balance -= cost;
56
57      ++calls;
58    }
59
60    /**
61     * Get the length of the longest call that can be
62     * made using this card (based on the rate and the b
63     *
64     * @return  The length of the longest call (in milli
65     */
66    public long getAvailableMilliseconds()
67    {
68      long        available;
69
70      available = 0;
71      if (calls < maxCalls)
72      {
73        available  = (long)(balance / rate);
74      }
75
76      return available;
77    }
78
79    /**
80     * Start a call.
81     *
82     * @param number   The number that was called
83     * @param start    The starting date/time of the cal
84     */
85    public void startCall(String number, Date start)
86    {
87      callNumbers[calls] = number;
88      callStarts[calls]  = start;
89    }
90  }
91
92
```

Variabl... | Breakp... | Expres...

Name	Value
no method return	
this	PhoneCard (id=346)
balance	9.0
callEnds	Date[2] (id=349)
callNumbers	String[2] (id=350)
calls	1
callStarts	Date[2] (id=352)
maxCalls	2
rate	1.666666666666666
available	0

```
PhoneCard@60f82f98
```

Console | Problems | Debug Shell

```
PhoneDriver [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:21:43 PM)
```

11. What is the current value of `balance`?

The current value of balance is 9.0

12. Click on the button.
13. Click on the "triangle icon" next to callNumbers to expand it.
14. What is the current value of `callNumbers[0]`?

The current value of callNumbers[0] is 540-568-1671

15. Click on the "triangle icon" next to callStarts to expand it.
16. What is the current value of `callStarts[0]`?

The current value of callStarts[0] is "Date"

17. Why does it have that value?

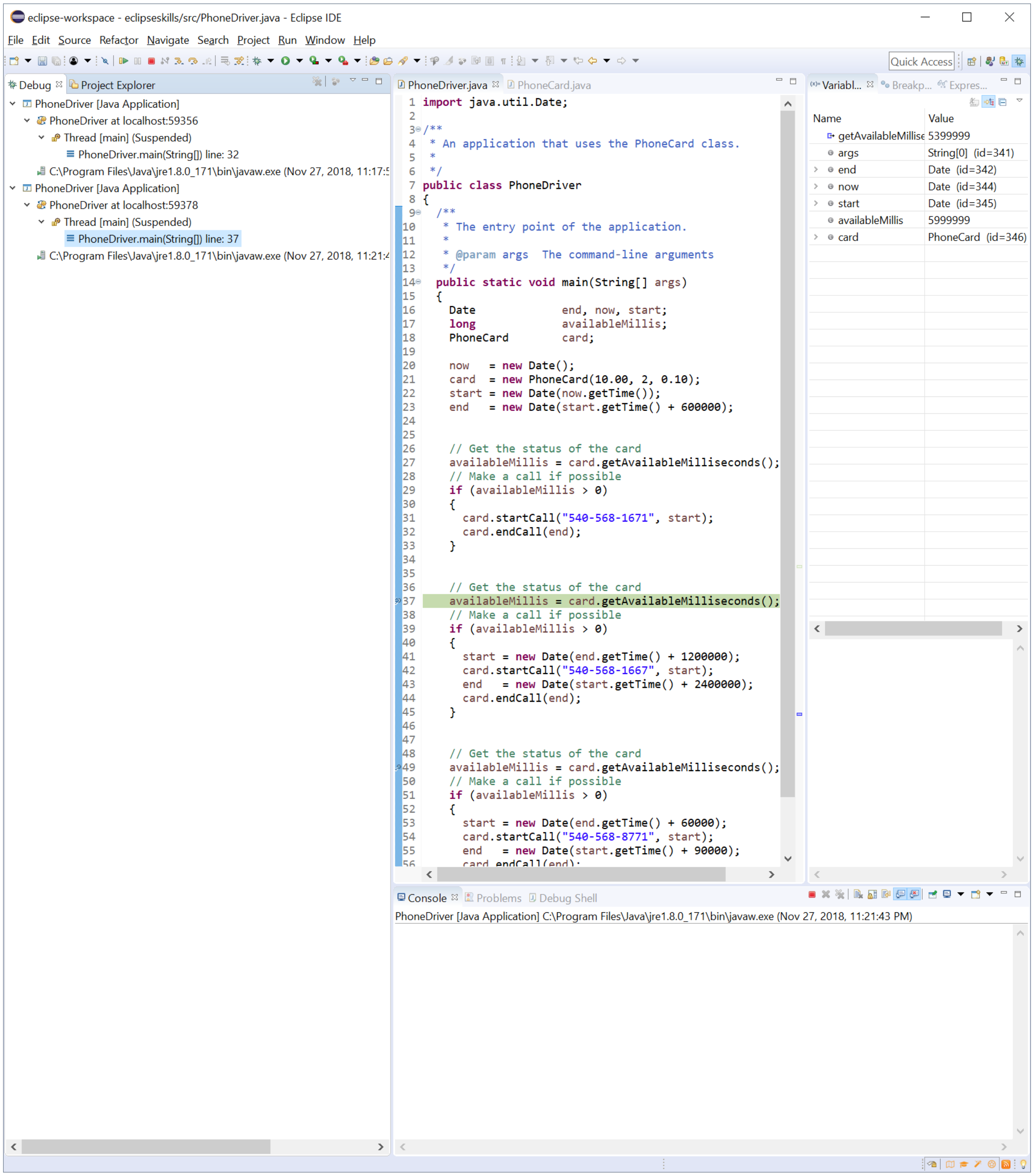
I'm assuming it has that value because the call should start at the current time?

18. Click on the button twice.
19. What happened?

Line 37 is now highlighted green

20. Add a breakpoint at line 46 in `PhoneDriver.java` (i.e., the line that constructs a `Date`).
21. Click on the button. This will run the application to the next breakpoint (i.e., line 46).

I tried to add a breakpoint to line 46 and it added it to line 49 instead. I think Eclipse hates me.


eclipse-workspace - eclipseskills/src/PhoneDriver.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Debug | Project Explorer

```
PhoneDriver [Java Application]
  PhoneDriver at localhost:59356
    Thread [main] (Suspended)
      PhoneDriver.main(String[]) line: 32
  C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:17:5
PhoneDriver [Java Application]
  PhoneDriver at localhost:59378
    Thread [main] (Suspended)
      PhoneDriver.main(String[]) line: 37
  C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:21:4
```

PhoneDriver.java | PhoneCard.java

```java
import java.util.Date;

/**
 * An application that uses the PhoneCard class.
 *
 */
public class PhoneDriver
{
  /**
   * The entry point of the application.
   *
   * @param args  The command-line arguments
   */
  public static void main(String[] args)
  {
    Date              end, now, start;
    long              availableMillis;
    PhoneCard         card;

    now   = new Date();
    card  = new PhoneCard(10.00, 2, 0.10);
    start = new Date(now.getTime());
    end   = new Date(start.getTime() + 600000);


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      card.startCall("540-568-1671", start);
      card.endCall(end);
    }


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      start = new Date(end.getTime() + 1200000);
      card.startCall("540-568-1667", start);
      end   = new Date(start.getTime() + 2400000);
      card.endCall(end);
    }


    // Get the status of the card
    availableMillis = card.getAvailableMilliseconds();
    // Make a call if possible
    if (availableMillis > 0)
    {
      start = new Date(end.getTime() + 60000);
      card.startCall("540-568-8771", start);
      end   = new Date(start.getTime() + 90000);
      card.endCall(end);
```

Variabl... | Breakp... | Expres...

Name	Value
getAvailableMillise	5399999
args	String[0] (id=341)
end	Date (id=342)
now	Date (id=344)
start	Date (id=345)
availableMillis	5999999
card	PhoneCard (id=346)

Console | Problems | Debug Shell

```
PhoneDriver [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (Nov 27, 2018, 11:21:43 PM)
```

22. Click on the button.

23. Why didn't the debugger step into the `Date` constructor?

    The debugger went to line 39. I don’t know why it didn’t step into the Date constructor.

24. Click on the button to run to the end.

I don’t think it ran to the end because it went to line 49 and the debugger says “already running.”

25. Click on Window+Perspective+Close Perspective to close the "Debug Perspective".

Part 6: Advanced Topics: This part of the lab will help you use the debugger more efficiently.

1. How can you display all of the breakpoints?

    By opening the Breakpoints tab inside of the debug perspective.

2. What is a conditional breakpoint?

    It will stop at a line only if the Boolean value changes on it.

3. How can you see variable references while debugging?

    By opening the variables tab inside of the debug perspective.