Explanation A

The method *tanh(real, img)* calculates a complex number. The assertion in line 4 checks if the method outputs the same value as the expected complex number *Complex(1, 1)*. Internally, the method computes the values of the variables *real2* and *img2* based on the method's *real* and *img* parameters.

However,

1. In V1, line 9, the method checks whether *real* is ≦ 20, which evaluates to *false*.

   Therefore, the code inside this "*if*" block is not executed. The next check, if img is ≧ 1 in line 20, evaluates to *true*.

   The values of *real2* and *img2* are thus calculated by adding *img* to *real2* and *img2* respectively.

2. In V2, due to the change in line 9, the method rather checks whether *real* is ≧ 20, which evaluates to *true*. This leads to *real2* being calculated as half the value of the exponential of *real* and *img2* being calculated as a function of *img*.

In summary, the change in line 9 leads to a different block of code being executed in V1 and V2. In V1, *real2* and *img2* are

directly incremented by the value of *img*.

In V2, *real2* and *img2* are calculated through more complex mathematical operations. The calculations in V2 cause the resulting complex number to differ from the expected *Complex(1, 1)* value in the assertion in line 4.

Explanation B

The method *tanh(real, img)* calculates a complex number. The assertion in line 4 checks if the method outputs the same value as the expected complex number *Complex(1, 1)*. Internally, the method computes the values of the variables *real2* and *img2* based on the method's *real* and *img* parameters.

However,

1. In V1, line 9, the method checks whether *real* is ≦ 20, which evaluates to *false*.

   Therefore, the code inside this "*if*" block is not executed. The next check, if img is ≧ 1 in line 20, evaluates to *true*.

   The values of *real2* and *img2* are thus calculated by adding *img* to *real2* and *img2* respectively.

2. In V2, due to the change in line 9, the method rather checks whether *real* is ≧ 20, which evaluates to *true*. This leads to *real2* being calculated as half the value of the exponential of *real*.
   In lines 11-18, the value of *k* is calculated as a multiplication of *img* and 0.6366197725814; the value of *a* – as a multiplication of *-k* and 1.570796251296997; the values of *remA* – as a manipulation over the values of *img*, *a*, and *k*; the value of *remB* – as a manipulation *remA*. Ultimately, *img2* is assigned the value of *remB**5 in line 18.

In summary, the change in line 9 leads to a different block of code being executed in V1 and V2. In V1, *real2* and *img2* are

directly incremented by the value of *img*.

In V2, *real2* and *img2* are calculated through more complex mathematical operations. The calculations in V2 cause the resulting complex number to differ from the expected *Complex(1, 1)* value in the assertion in line 4.

Explanation C

The method *tanh(real, img)* calculates a complex number. The assertion in line 4 checks if the method outputs the same value as the expected complex number *Complex(1, 1)*. Internally, the method computes the value of the variables *real2* and *img2* based on its *real* and *img* parameters,

which are initialized to *positive infinity* and 1, in lines 2 and 3, respectively.
The values of *real2* and *img2* are first initialized to 0 in lines 7 and 8, respectively.

However,

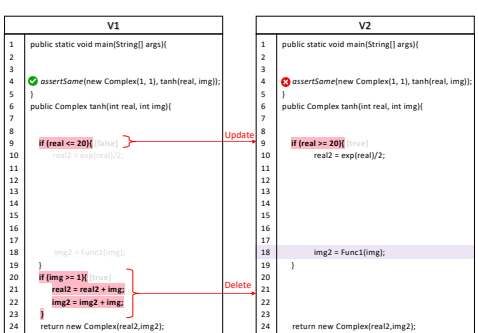
1. In V1, line 9, the method checks whether *real* is ≦ 20, which evaluates to *false*
   as *real* is *positive infinity*.
   Therefore, the code inside this "*if*" block is not executed. The next check, if img is ≧ 1 in line 20, evaluates to *true*
   as img is 1.
   The values of *real2* and *img2* are thus calculated by adding *img* to *real2* and *img2* respectively,
   resulting in the values of *real2* and *img2* being 1 each.
2. In V2, due to the change in line 9, the method rather checks whether *real* is ≧ 20, which evaluates to *true*. This leads to *real2* being calculated as half the value of the exponential of *real* and *img2* being calculated as a function of *img*.

In summary, the change in line 9 leads to a different block of code being executed in V1 and V2. In V1, *real2* and *img2* are
initialized as 0 and then
directly incremented by the value of *img*,
which is 1.
In V2, *real2* and *img2* are calculated through more complex mathematical operations. The calculations in V2 cause the resulting complex number to differ from the expected *Complex(1, 1)* value in the assertion in line 4.

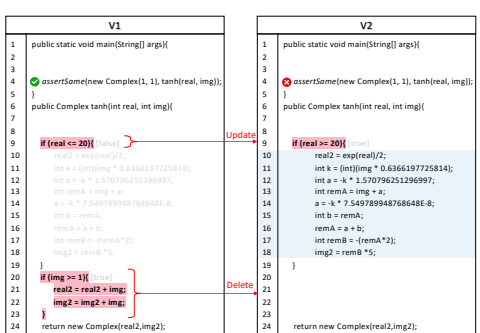
Notations: Colored backgrounds highlight the differences between the views.

FYI: Views are given below again, for your reference.

View A



View B



View C