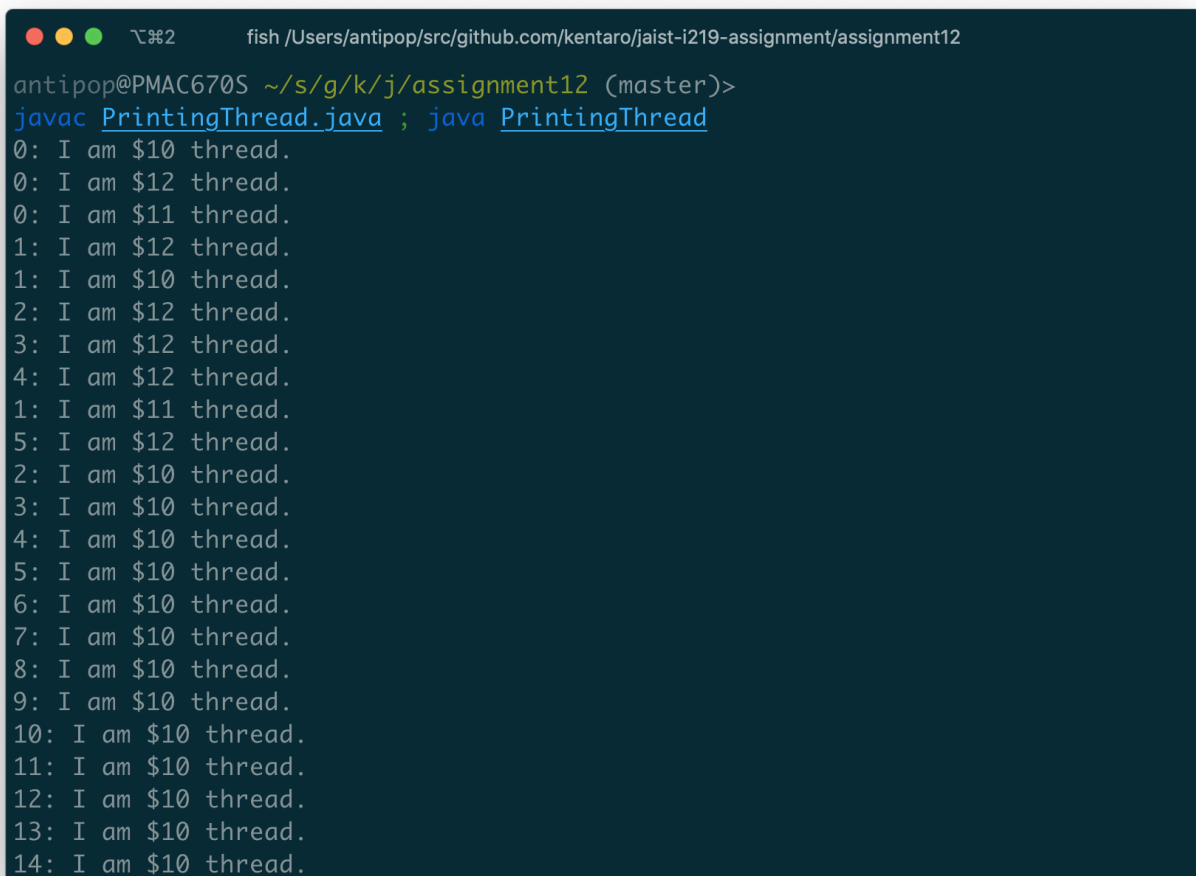


Assignment 12

- 氏名: 栗林健太郎
- 学生番号: 2030006
- 作成日: 2020年11月23日

PrintingThread

3つのスレッドを立ち上げ、それぞれのスレッド内で50回プリント処理を行っている。スレッドは並行処理されるため、以下の実行結果の通り、プリント結果が入り混じって表示される。



```
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PrintingThread.java ; java PrintingThread
0: I am $10 thread.
0: I am $12 thread.
0: I am $11 thread.
1: I am $12 thread.
1: I am $10 thread.
2: I am $12 thread.
3: I am $12 thread.
4: I am $12 thread.
1: I am $11 thread.
5: I am $12 thread.
2: I am $10 thread.
3: I am $10 thread.
4: I am $10 thread.
5: I am $10 thread.
6: I am $10 thread.
7: I am $10 thread.
8: I am $10 thread.
9: I am $10 thread.
10: I am $10 thread.
11: I am $10 thread.
12: I am $10 thread.
13: I am $10 thread.
14: I am $10 thread.
```

UnsafeInc, NonatomicCounter

3つのスレッドを立ち上げ、それぞれのスレッド内で100万回、**NonatomicCounter**クラスで実装されたカウンターをインクリメントする処理を行っている。

各スレッドからのカウンターへの更新処理を同期していないため、以下の実行結果の通り、実行するたびに結果が異なっている（レースコンディションが存在する）。

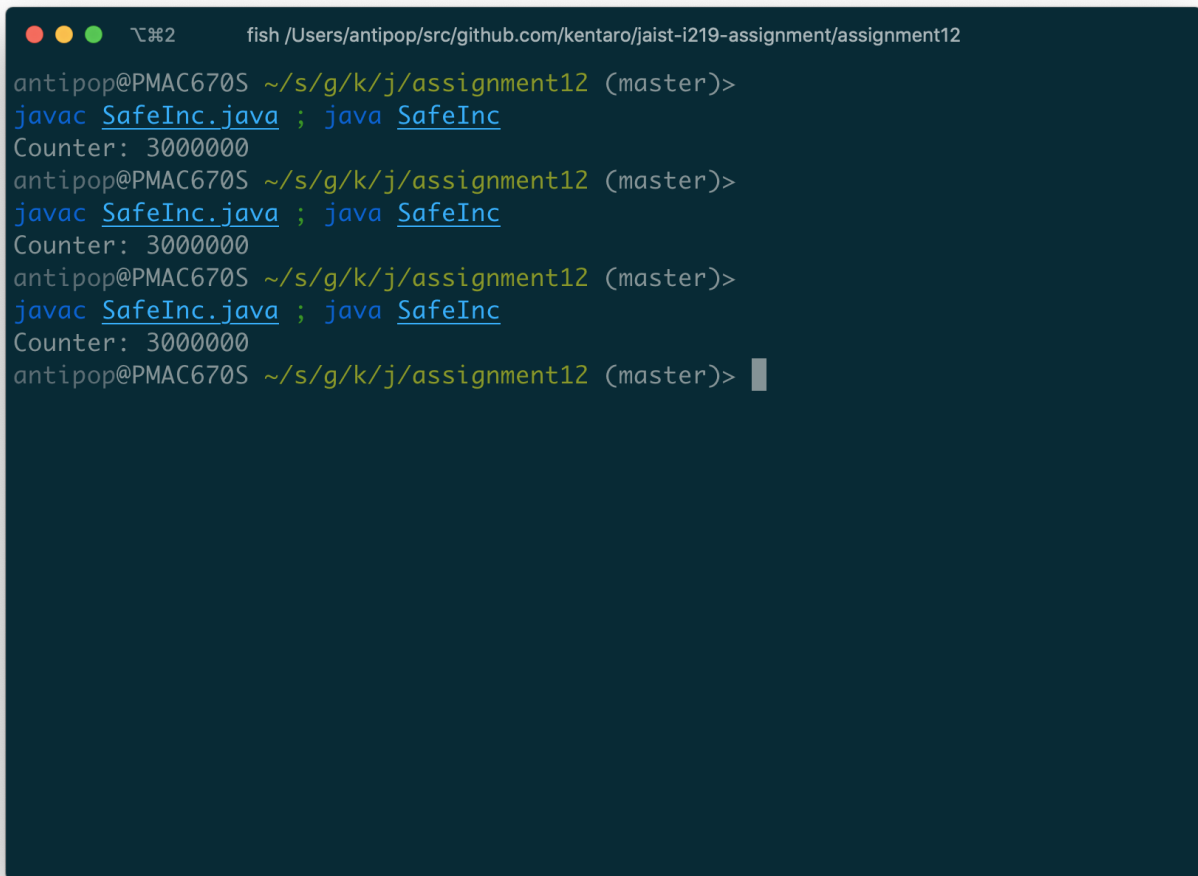
```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12

antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac UnsafeInc.java ; java UnsafeInc
Counter: 1024612
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac UnsafeInc.java ; java UnsafeInc
Counter: 1876060
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac UnsafeInc.java ; java UnsafeInc
Counter: 1009885
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac UnsafeInc.java ; java UnsafeInc
Counter: 1021688
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)> █
```

SafeInc, AtomicCounter

3つのスレッドを立ち上げ、それぞれのスレッド内で100万回、**AtomicCounter**クラスで実装されたカウンターをインクリメントする処理を行っている。

各スレッドからのカウンターへの更新処理を同期しているため、以下の実行結果の通り、何度実行しても同じ結果が表示される（レースコンディションが存在しない）。



```
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac SafeInc.java ; java SafeInc
Counter: 3000000
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac SafeInc.java ; java SafeInc
Counter: 3000000
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac SafeInc.java ; java SafeInc
Counter: 3000000
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
```

SafeInc2, NonatomicCounter

3つのスレッドを立ち上げ、それぞれのスレッド内で100万回、**NonatomicCounter**クラスで実装されたカウンターをインクリメントする処理を行っている。

各スレッドからは、**NonatomicCounter**クラスのオブジェクトへのロックを獲得しているため、**counter.inc()**への呼び出しが同期され、以下の実行結果の通り、何度実行しても同じ結果が表示される（レースコンディションが存在しない）。

```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12

antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac SafeInc2.java ; java SafeInc2
Counter: 3000000
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac SafeInc2.java ; java SafeInc2
Counter: 3000000
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac SafeInc2.java ; java SafeInc2
Counter: 3000000
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)> █
```

PseudoSafeInc, PseudoAtomicCounter

3つのスレッドを立ち上げ、それぞれのスレッド内で100万回、**PseudoAtomicCounter**クラスで実装されたカウンターをインクリメントする処理を行っている。また、その際、各スレッドからはそれぞれインスタンス化した**PseudoAtomicCounter**オブジェクトを参照している。

PseudotomicCounterクラスのメソッドは**synchronized**でメソッドレベルの同期が宣言されているため一見問題ないように見えるが、**count**属性はスタティック変数として宣言されているためそれへの操作が同期されず、以下の実行結果の通り、以下の実行結果の通り、実行するたびに結果が異なっている（レースコンディションが存在する）。

```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12
antipop@PMAC670S ~/s/g/k/j/assignment12 (master) [1]>
javac PseudoSafeInc.java ; java PseudoSafeInc
Counter: 2189381
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc.java ; java PseudoSafeInc
Counter: 2166465
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc.java ; java PseudoSafeInc
Counter: 2180735
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc.java ; java PseudoSafeInc
Counter: 2171314
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)> █
```

PseudoSafeInc2, NonatomicCounter

3つのスレッドを立ち上げ、それぞれのスレッド内で100万回、**NonatomicCounter**クラスで実装されたカウンターをインクリメントする処理を行っている。

PseudoSafeInc2クラスの**run**メソッドはカウントアップ処理する直前で**synchronized**で自らに対してロックを取っているため一見問題ないように見えるが、**NonatomicCounter**の**inc()**メソッドの処理に対しては操作が同期されておらず、以下の実行結果の通り、以下の実行結果の通り、実行するたびに結果が異なっている（レースコンディションが存在する）。

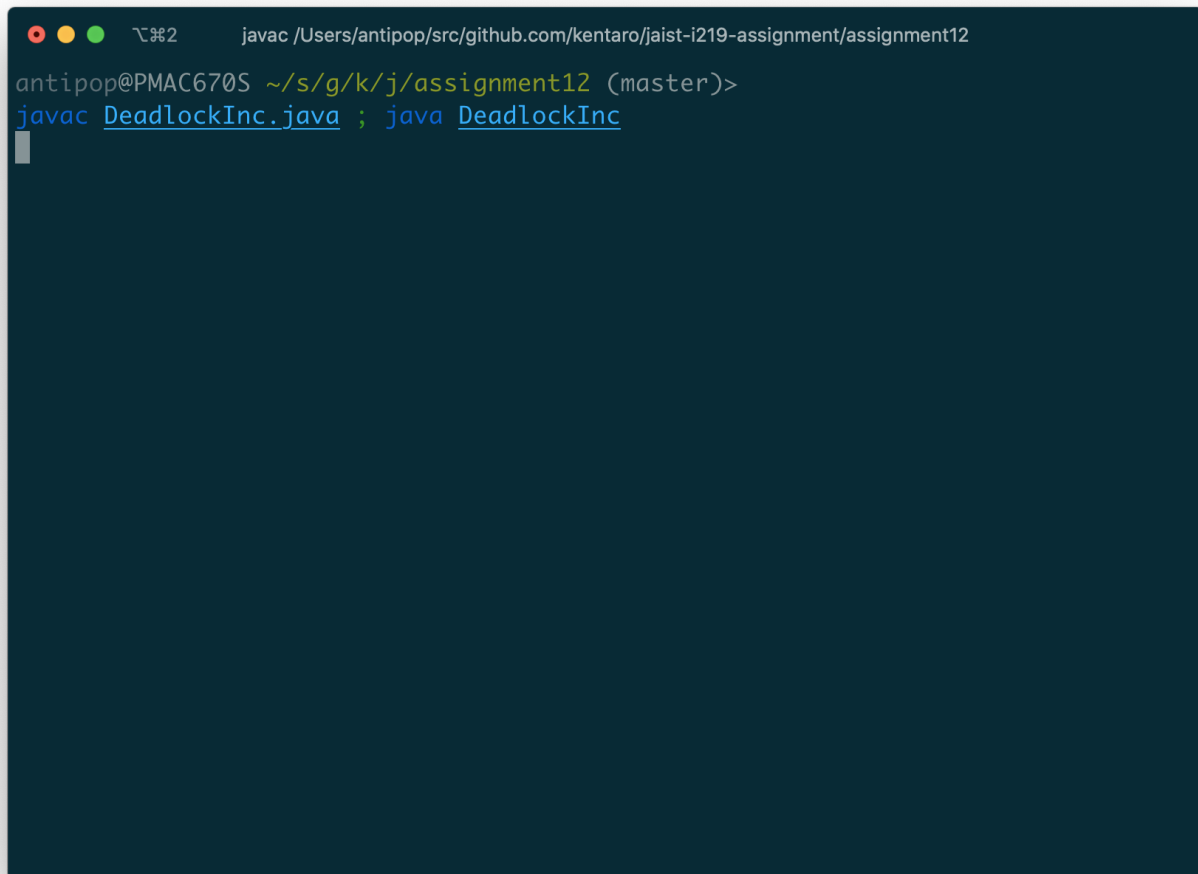
```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12

antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc2.java ; java PseudoSafeInc2
Counter: 1295850
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc2.java ; java PseudoSafeInc2
Counter: 1424901
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc2.java ; java PseudoSafeInc2
Counter: 1281171
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac PseudoSafeInc2.java ; java PseudoSafeInc2
Counter: 1389109
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)> █
```

DeadlockInc, NonatomicCounter

2つのスレッドを立ち上げ、それぞれのスレッド内で5,000回、**NonatomicCounter**クラスで実装されたカウンターをインクリメントする処理を行おうとしている。

DeadlockIncクラスの`run`メソッドはカウントアップ処理する直前で、`counter1`と`counter2`とをその順番で同期する処理を行っている。そのため、2つのスレッドがカウンターの順番を逆に参照してしまうとそれぞれのスレッドが他方のスレッドのロックが開放されるのを待ち続けてしまうため、以下の実行結果の通り、プログラムが終了しない（デッドロックが発生する）。

A terminal window with a dark blue background and light green text. The window title bar shows three colored circles (red, yellow, green) and the text 'javac /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12'. The terminal content shows the user 'antipop@PMAC670S' in the directory '~/s/g/k/j/assignment12 (master)' running the command 'javac DeadlockInc.java ; java DeadlockInc'. A cursor is visible on the line following the command.

```
javac /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12  
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>  
javac DeadlockInc.java ; java DeadlockInc
```

NoDeadlockInc, NonatomicCounter

2つのスレッドを立ち上げ、それぞれのスレッド内で5,000回、**NonatomicCounter**クラスで実装されたカウンターをインクリメントする処理を行おうとしている。

DeadlockInc、と違って**counter1**と**counter2**とを同じオブジェクトにひもづく順番で参照しているため、以下の実行結果の通り、プログラムが終了する（デッドロックが発生しない）。

```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12

antipop@PMAC670S ~/s/g/k/j/assignment12 (master) [SIGINT]>
javac NoDeadlockInc.java ; java NoDeadlockInc
Counter1: 9998, Counter2: 9998
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac NoDeadlockInc.java ; java NoDeadlockInc
Counter1: 9998, Counter2: 9998
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac NoDeadlockInc.java ; java NoDeadlockInc
Counter1: 9998, Counter2: 9998
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac NoDeadlockInc.java ; java NoDeadlockInc
Counter1: 9998, Counter2: 9998
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)> █
```

Bounded Buffer Problem

Bounded Buffer Problemの実装について検討する

FBBProb1

まず、バッファへの操作がなにも同期されない実装で動作させてみると、レースコンディションが存在するため、以下の実行結果の通り、Senderが送信したメッセージとReceiverが受け取るメッセージとは一致しない。


```
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null, null, null, null, null, null, null, null, null, null, null, null,  
    null, null, null, null, null, null, null, null, null, null, null, null, null, n  
ull, null, null, null, null, null, null, null, null, null, null, null, null, nul  
l, null, null]
```

Failure!

```
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>  
javac FBBProb1.java ; java FBBProb1 # ← これを実行した
```

FBBProb2

次に、バッファへのputおよびgetメソッドをsynchronizedで同期する実装で動作させてみる。

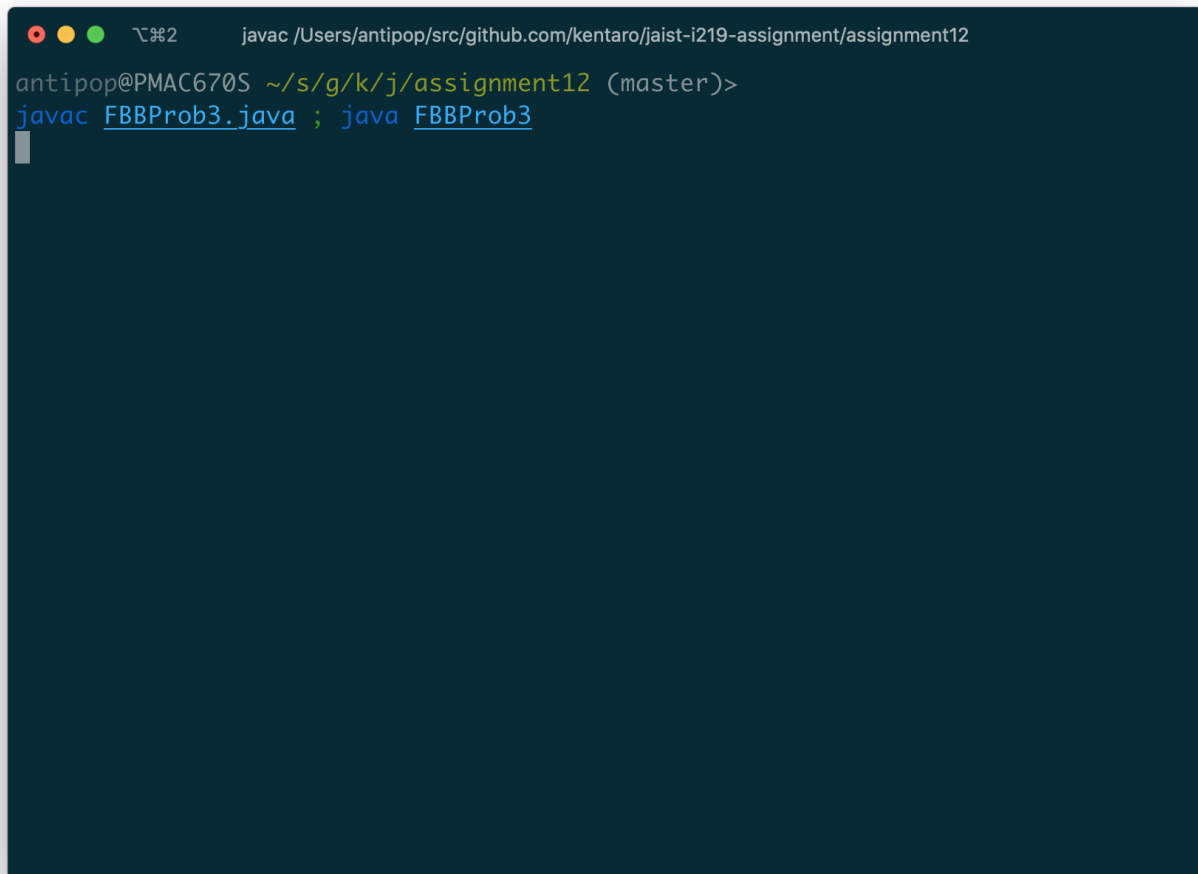
しかし、`put`では`capacity`で設定されている数のメッセージは捨てられてしまい、かつ、`get`ではキューにメッセージが入っていないと`null`を返すため、以下の実行結果の通り、Senderが送信したメッセージとReceiverが受け取るメッセージとは一致しない。

```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12
, 9770, null, null, null, null, null, 9771, null, null, null, null, null, null,
null, null, null, null, null, 9772, null, 9773, null, null, 9774, null, null, nu
ll, 9775, 9776, null, null, null, null, null, null, null, null, null, null, 9777
, null, null, null, 9778, null, null, 9779, null, 9780, null, 9781, null, null,
null, null, null, null, 9782, null, null, 9783, 9784, null, null, null, null, nu
ll, null, null, null, null, null, null, null, null, 9785, null, null, null, 9786, null
, 9787, null, 9788, null, null, null, null, null, null, 9789, null, 9790, null,
9791, null, null, 9792, null, 9793, null, null, 9794, null, 9795, null, null, 97
96, null, null, 9797, null, null, null, null, null, null, null, null, null, null
, null, null, null, null, 9798, null, 9799, null, null, 9800, null, 9801, 9802,
9803, null, 9804, null, 9805, null, 9806, 9807, null, 9808, 9809, null, null, 98
10, null, null, null, null, null, null, null, null, null, null, 9811, 9812, null, null
, null, null, null, null, null, null, null, null, 9813, 9814, 9815, null, 9816, null,
null, null, null, null, null, null, null, null, null, null, null, null, null, nu
ll, 9817, 9818, null, null, null, null, null, null, null, null, null, null, 9819, 9820, null
, null, 9821, null, 9822, null, null, null, null, null, null, null, null, null,
null, null, null, null, null, null, null, null, 9823, null, 9824, null, 9825, 98
26, null, null, 9827, null, null, 9828, null, 9829, null, null, null, null, null
, null, null, 9830, null, null, null, null, null, null, null, 9831, null, null, null,
null, null, null, null, null, null, null, 9832, 9833, 9834, 9835, null, null, 9836, 98
37, null, null, null, null, null, null, null, null, null, null, null, null, null, null
, 9838, null, null, null, null, null, null, null, null, null, null, null]
Failure!
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac FBBProb2.java ; java FBBProb2 # ← これを実行した
```

FBBProb3

そこで、バッファへの`put`時にキューにメッセージが入れられるようになるまで待ち、かつ、`get`時にキューから`null`でないメッセージが取得できるようになるまで待つ実装で動作させてみる。

今度は、上記で待つ実装をした箇所でロックを持ったままループに入るために2つのスレッドの間でデッドロックが発生し、以下の実行結果の通り、プログラムが終了しない。

A terminal window with a dark blue background. The title bar shows three colored circles (red, yellow, green) and the text "javac /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12". The terminal content shows the user "antipop@PMAC670S" in the directory "~/s/g/k/j/assignment12" on the "(master)" branch. They have entered the command "javac FBBProb3.java ; java FBBProb3" and a cursor is visible on the line below.

```
javac /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12  
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>  
javac FBBProb3.java ; java FBBProb3
```

BBProb

最後に、前述で待つ処理を実装するループ内で`wait()`を実行し、キューへの操作・参照が終わったら`notifyAll()`を実行する実装に変更してどうさせてみる。

今度は、`wait()`でロックを開放したあとに他のスレッドが`notifyAll()`を実行するまで待つという動作になるため、`put`はキューにメッセージを取りこぼしなく入れることができ、また`get`はキューからすべてのメッセージを取得することができ、以下の実行結果の通り、Senderが送信したメッセージとReceiverが受け取るメッセージとが一致する。

```
fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/assignment12

9716, 9717, 9718, 9719, 9720, 9721, 9722, 9723, 9724, 9725, 9726, 9727, 9728, 9
729, 9730, 9731, 9732, 9733, 9734, 9735, 9736, 9737, 9738, 9739, 9740, 9741, 974
2, 9743, 9744, 9745, 9746, 9747, 9748, 9749, 9750, 9751, 9752, 9753, 9754, 9755,
9756, 9757, 9758, 9759, 9760, 9761, 9762, 9763, 9764, 9765, 9766, 9767, 9768, 9
769, 9770, 9771, 9772, 9773, 9774, 9775, 9776, 9777, 9778, 9779, 9780, 9781, 978
2, 9783, 9784, 9785, 9786, 9787, 9788, 9789, 9790, 9791, 9792, 9793, 9794, 9795,
9796, 9797, 9798, 9799, 9800, 9801, 9802, 9803, 9804, 9805, 9806, 9807, 9808, 9
809, 9810, 9811, 9812, 9813, 9814, 9815, 9816, 9817, 9818, 9819, 9820, 9821, 982
2, 9823, 9824, 9825, 9826, 9827, 9828, 9829, 9830, 9831, 9832, 9833, 9834, 9835,
9836, 9837, 9838, 9839, 9840, 9841, 9842, 9843, 9844, 9845, 9846, 9847, 9848, 9
849, 9850, 9851, 9852, 9853, 9854, 9855, 9856, 9857, 9858, 9859, 9860, 9861, 986
2, 9863, 9864, 9865, 9866, 9867, 9868, 9869, 9870, 9871, 9872, 9873, 9874, 9875,
9876, 9877, 9878, 9879, 9880, 9881, 9882, 9883, 9884, 9885, 9886, 9887, 9888, 9
889, 9890, 9891, 9892, 9893, 9894, 9895, 9896, 9897, 9898, 9899, 9900, 9901, 990
2, 9903, 9904, 9905, 9906, 9907, 9908, 9909, 9910, 9911, 9912, 9913, 9914, 9915,
9916, 9917, 9918, 9919, 9920, 9921, 9922, 9923, 9924, 9925, 9926, 9927, 9928, 9
929, 9930, 9931, 9932, 9933, 9934, 9935, 9936, 9937, 9938, 9939, 9940, 9941, 994
2, 9943, 9944, 9945, 9946, 9947, 9948, 9949, 9950, 9951, 9952, 9953, 9954, 9955,
9956, 9957, 9958, 9959, 9960, 9961, 9962, 9963, 9964, 9965, 9966, 9967, 9968, 9
969, 9970, 9971, 9972, 9973, 9974, 9975, 9976, 9977, 9978, 9979, 9980, 9981, 998
2, 9983, 9984, 9985, 9986, 9987, 9988, 9989, 9990, 9991, 9992, 9993, 9994, 9995,
9996, 9997, 9998, 9999]
Success!
antipop@PMAC670S ~/s/g/k/j/assignment12 (master)>
javac BBProb.java ; java BBProb # ← これを実行した
```