

Problem 1115: Print FooBar Alternately

Problem Information

Difficulty: Medium

Acceptance Rate: 71.29%

Paid Only: No

Tags: Concurrency

Problem Description

Suppose you are given the following code:

```
class FooBar { public void foo() { for (int i = 0; i < n; i++) { print("foo"); } } public void bar() { for (int i = 0; i < n; i++) { print("bar"); } } }
```

The same instance of `FooBar` will be passed to two different threads:

* thread `A` will call `foo()`, while * thread `B` will call `bar()`.

Modify the given program to output "foobar" `n` times.

Example 1:

Input: n = 1 **Output:** "foobar" **Explanation:** There are two threads being fired asynchronously. One of them calls foo(), while the other calls bar(). "foobar" is being output 1 time.

Example 2:

Input: n = 2 **Output:** "foobarfoobar" **Explanation:** "foobar" is being output 2 times.

Constraints:

* `1 <= n <= 1000`

Code Snippets

C++:

```
class FooBar {  
private:  
    int n;  
  
public:  
    FooBar(int n) {  
        this->n = n;  
    }  
  
    void foo(function<void()> printFoo) {  
  
        for (int i = 0; i < n; i++) {  
  
            // printFoo() outputs "foo". Do not change or remove this line.  
            printFoo();  
        }  
    }  
  
    void bar(function<void()> printBar) {  
  
        for (int i = 0; i < n; i++) {  
  
            // printBar() outputs "bar". Do not change or remove this line.  
            printBar();  
        }  
    };
```

Java:

```
class FooBar {  
private int n;  
  
    public FooBar(int n) {  
        this.n = n;  
    }
```

```
public void foo(Runnable printFoo) throws InterruptedException {

    for (int i = 0; i < n; i++) {

        // printFoo.run() outputs "foo". Do not change or remove this line.
        printFoo.run();
    }
}

public void bar(Runnable printBar) throws InterruptedException {

    for (int i = 0; i < n; i++) {

        // printBar.run() outputs "bar". Do not change or remove this line.
        printBar.run();
    }
}
```

Python3:

```
class FooBar:

    def __init__(self, n):
        self.n = n


    def foo(self, printFoo: 'Callable[[], None]') -> None:

        for i in range(self.n):

            # printFoo() outputs "foo". Do not change or remove this line.
            printFoo()


    def bar(self, printBar: 'Callable[[], None]') -> None:

        for i in range(self.n):

            # printBar() outputs "bar". Do not change or remove this line.
            printBar()
```