```
//
//  ThreadSafeStack.m
//  ThreadSafeStack
//
//  Created by Juncheng Han on 2/12/18.
//  Copyright © 2018 Jason H. All rights reserved.
//

#import "ThreadSafeStack.h"
#import "ListNode.h"

@interface ThreadSafeStack()

@property (nonatomic, strong) ListNode *head;
@property (nonatomic, strong) dispatch_queue_t isolationQueue;

@end

@implementation ThreadSafeStack

- (instancetype)init {
    self = [super init];
    if (self) {
        self.head = NULL;
        NSString *name = [NSString
         stringWithFormat:@"com.ThreadSafeStack.dispatchQueue.%ld", (unsigned
         long)self.hash];
        self.isolationQueue = dispatch_queue_create([name
         cStringUsingEncoding:NSASCIIStringEncoding], DISPATCH_QUEUE_CONCURRENT);
    }
    return self;
}

#pragma mark - public methods

- (id)peek {
    __block id res;
    dispatch_sync(self.isolationQueue, ^{
        res = self.head.value;
    });
    return res;
}

- (void)pop {
    dispatch_barrier_async(self.isolationQueue, ^{
        if (self.head == NULL) {
            abort();
        } else {
            self.head = self.head.next;
        }

#if DEBUG
        NSLog(@"%@", self);
#endif
```

```
    });
}

- (void)push:(id)object {
    dispatch_barrier_sync(self.isolationQueue, ^{
        self.head = [[ListNode alloc] initWithValue:object andNext:self.head];
#if DEBUG
        NSLog(@"%@", self);
#endif
    });
}

#pragma mark - override methods

- (NSString *)description {
    NSMutableString *des = [NSMutableString stringWithFormat:@"%@: ",
     NSStringFromClass([self class])];
    ListNode *temp = self.head;
    while (temp != NULL) {
        [des appendString:[NSString stringWithFormat:@"%@ ", [temp.value
         description]]];
        temp = temp.next;
    }

    return [des copy];
}

#pragma mark - NSCopy and NSMutableCopy

- (id)copyWithZone:(NSZone *)zone {

    ThreadSafeStack *copy = [[[self class] alloc] init];

    copy.head = self.head;

    return copy;
}

- (id)mutableCopyWithZone:(NSZone *)zone {
    return [self copyWithZone:zone];
}

#pragma mark - NSCoding

#define kHeadKey @"headKey"

- (void)encodeWithCoder:(nonnull NSCoder *)aCoder {
    [aCoder encodeObject:_head forKey:kHeadKey];
}

- (nullable instancetype)initWithCoder:(nonnull NSCoder *)aDecoder {
    ListNode *head = [aDecoder decodeObjectForKey:kHeadKey];

    self = [super init];
```

```
    if (self) {
        self.head = head;
        NSString *name = [NSString
         stringWithFormat:@"com.ThreadSafeStack.dispatchQueue.%ld", (unsigned
         long)self.hash];
        self.isolationQueue = dispatch_queue_create([name
         cStringUsingEncoding:NSASCIIStringEncoding], DISPATCH_QUEUE_CONCURRENT);
    }
    return self;
}

/*
 state: Context information that is used in the enumeration
 buffer: A C array of objects over which the sender is to iterate
 len: The maximum number of objects to return in buffer
 */
- (NSUInteger)countByEnumeratingWithState:(nonnull NSFastEnumerationState *)state
                                  objects:(id  _Nullable __unsafe_unretained *
                                   _Nonnull)buffer
                                    count:(NSUInteger)len {
    return 0;
}

@end
```