Necklace Designer

basic

an app for iPhone using objective-C

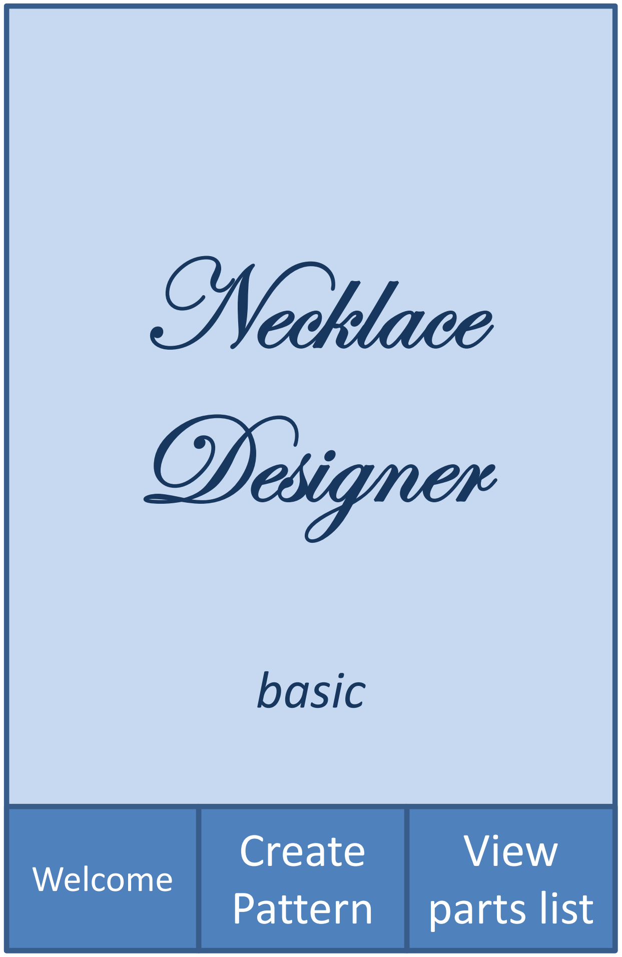
created by

MaryLou Washko

for CIS238 at MCCC

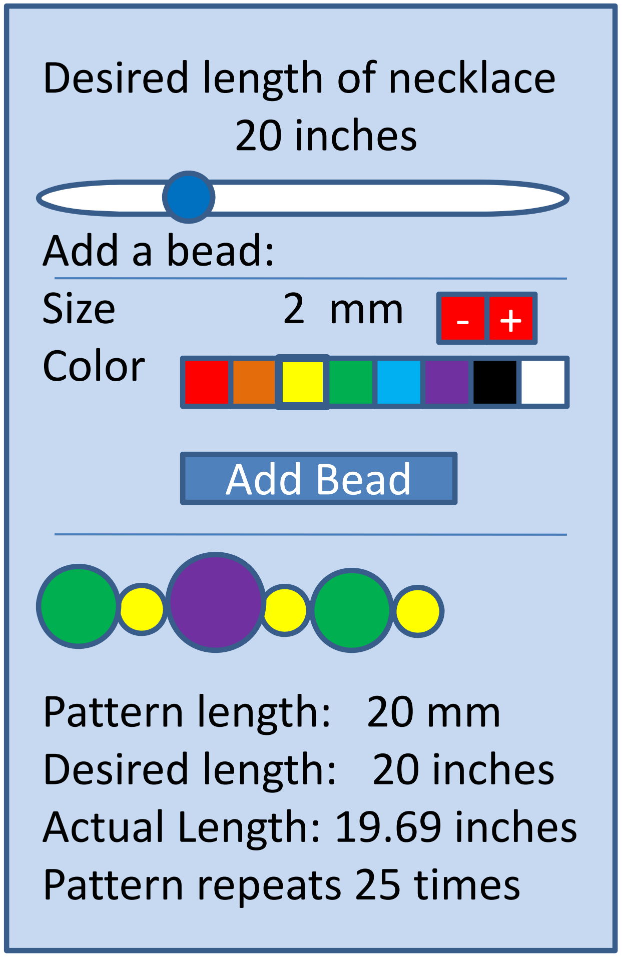
May 9, 2013

Welcome to the preview for Necklace Designer



In this screen beads are added to create a pattern for a basic necklace.

First the necklace length is set and then the size and color of each bead as they are added to the pattern.



Desired length of necklace
20 inches

Add a bead:

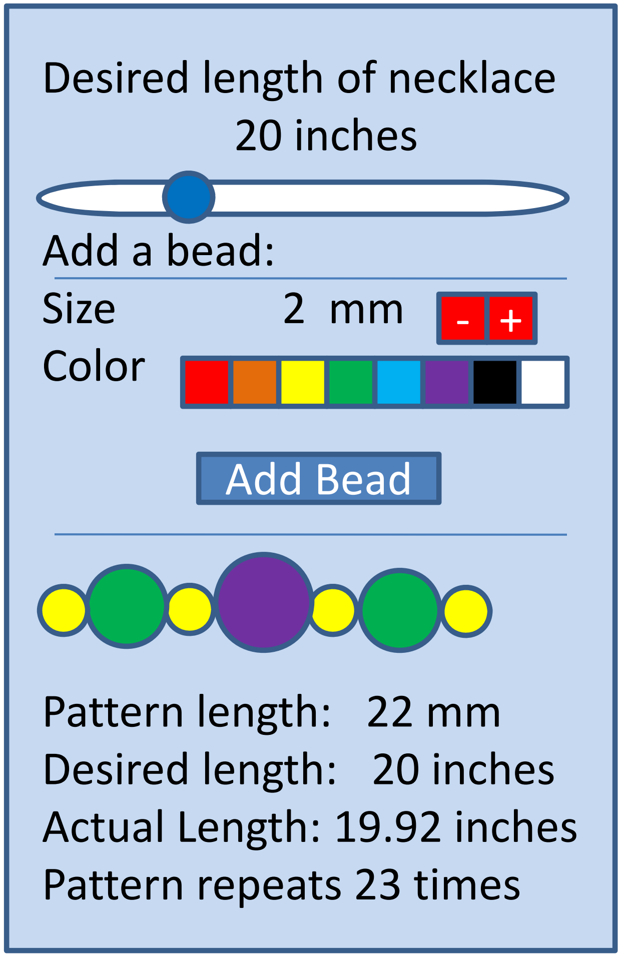
Size 2 mm

Color 🟥🟧🟨🟩🟦🟪⬛⬜

Pattern length: 20 mm
Desired length: 20 inches
Actual Length: 19.69 inches
Pattern repeats 25 times

When the yellow bead is added to the left side of the pattern, the dimensions are updated.

By clicking on the pattern a new screen opens to preview the necklace.



Desired length of necklace
20 inches

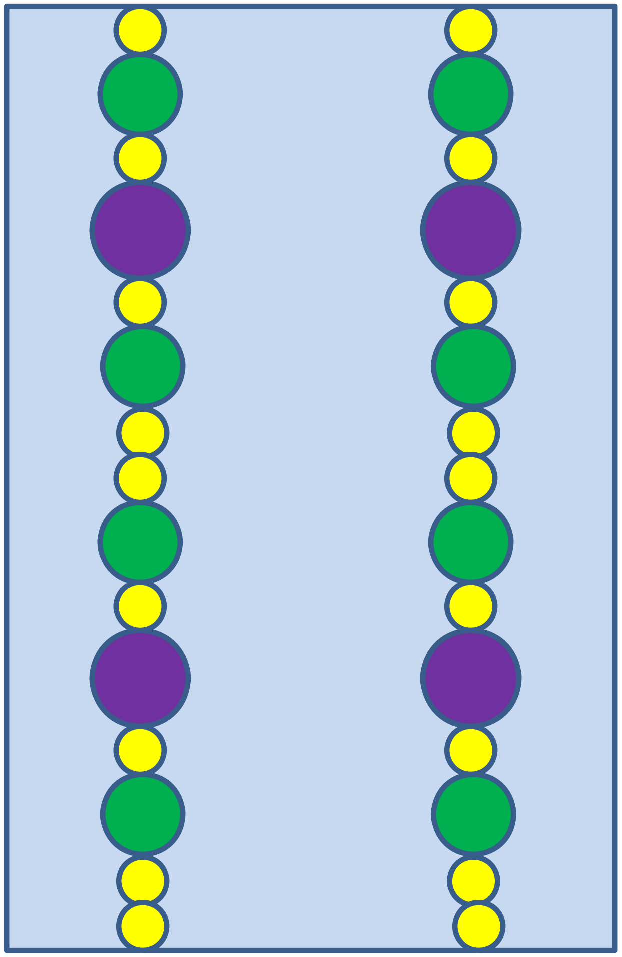
Add a bead:

Size 2 mm

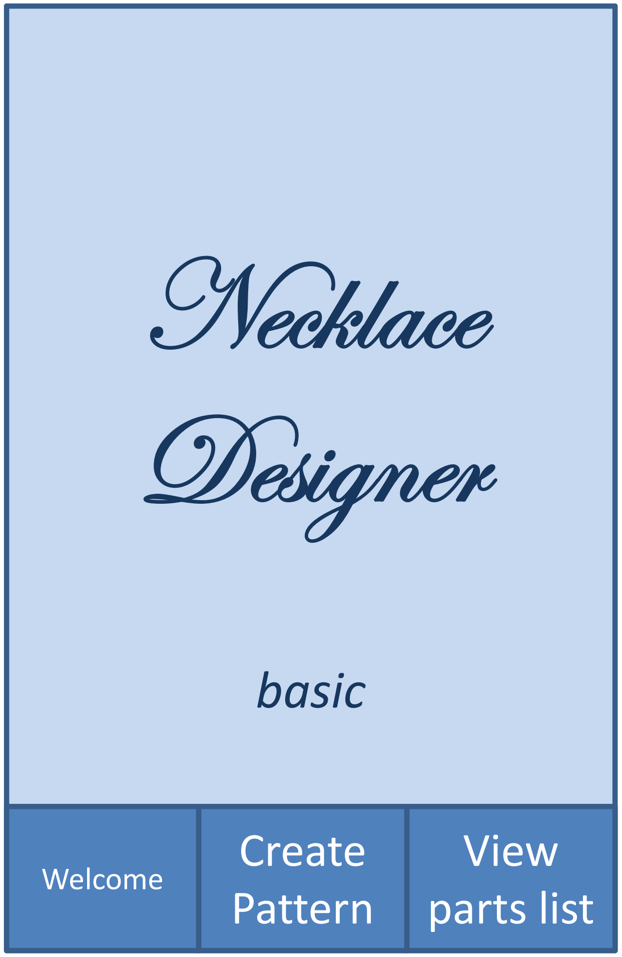
Color 🟥🟧🟨🟩🟦🟪⬛⬜

Pattern length: 22 mm
Desired length: 20 inches
Actual Length: 19.92 inches
Pattern repeats 23 times

The pattern is repeated to resemble a full necklace



The parts list is created from the bead pattern entered.

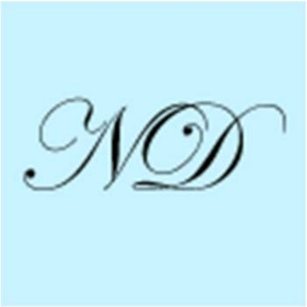


The application calculates the number of beads required to make the desired length of the necklace and displays the parts list to take the guess work out of purchasing beads.

Parts List

Qty	Description
92	2mm yellow beads
46	4mm green beads
23	6mm purple beads

The App Icon



The Text for the App Store

Take the guess work out of how many beads to buy when designing a necklace project!

Necklace Designer *basic* allows you to create up to a 10 bead pattern to get the basic idea of how it will look in a necklace. Necklace Designer allows you to select a necklace length and select up to 10 beads ranging from 2mm to 10mm in 8 basic colors (red, orange, yellow, green, blue, purple, black, and white). From the information entered, the Necklace Designer will calculate the length of the pattern in mm, the actual length of the necklace using this pattern of beads, and the number of times the pattern repeats for a desired necklace length. Finally it produces a parts list of all beads used and how many of each bead will be required to complete the necklace project. While designing the pattern the necklace length can be adjusted to fit the pattern chosen to reveal the actual length and related parts list for ultimate results.

A List of Frameworks Used

- UIKit
- Foundation
- Core Graphics

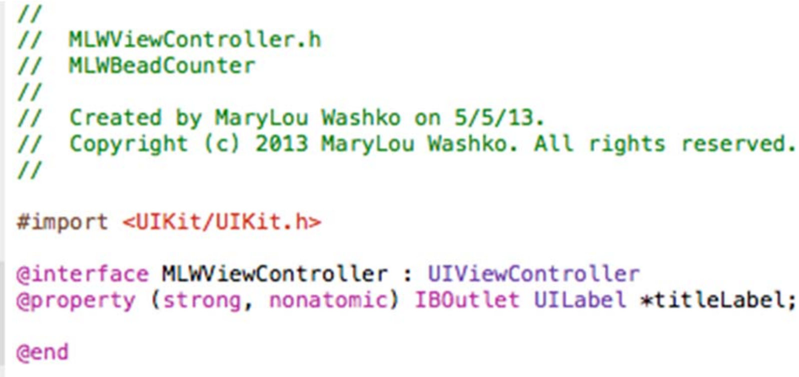
A List of Classes Included

- **TabViewController** - title screen, necklace creator and parts list
- **ViewController** – Shows title page
- **BeadViewController** - to enter all the required information: necklace length, bead pattern
- **NecklaceViewController** - produces a color graphic representation of a necklace repeating the bead pattern given
- **PartsListViewController** - displays the list of beads with quantity required

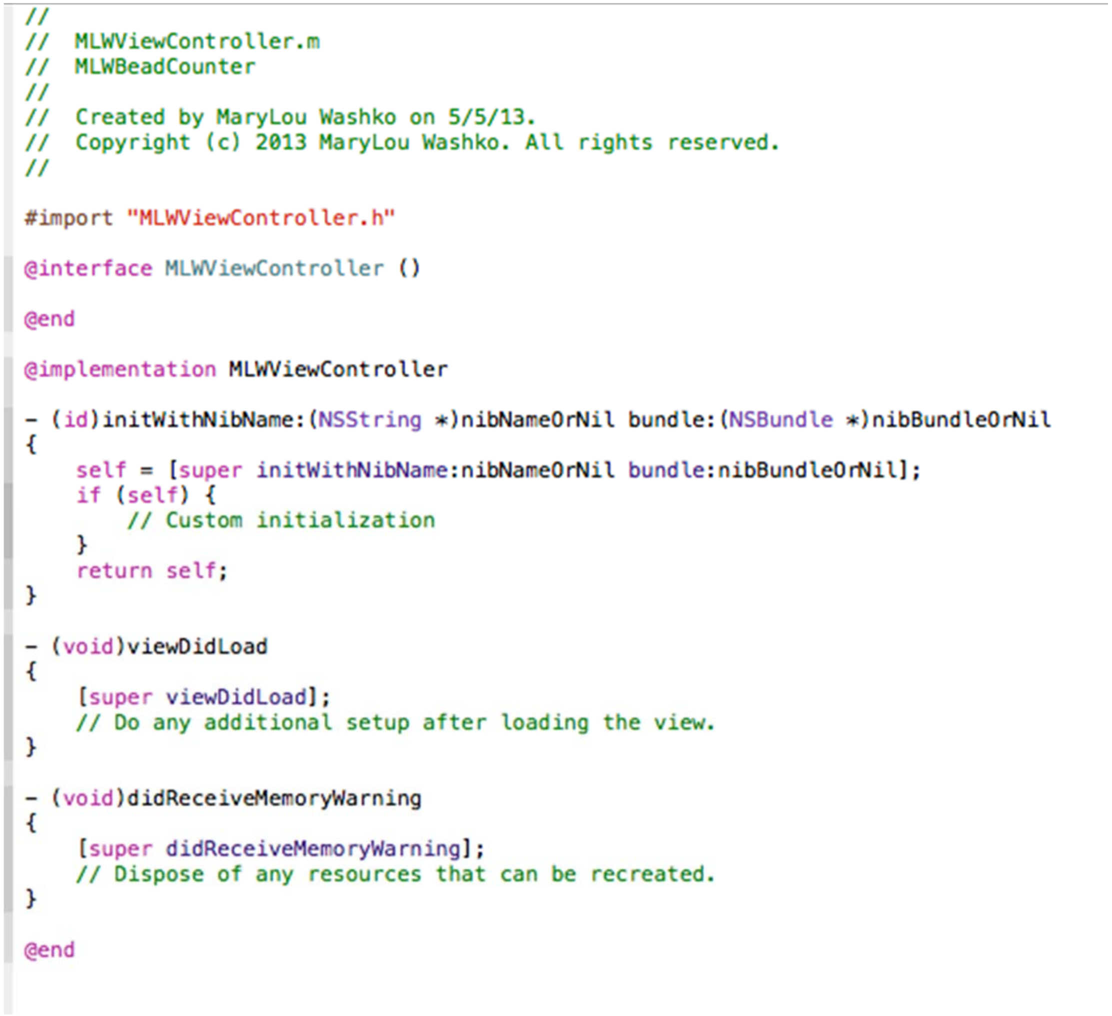
- **BeadData** - Describes a bead: size and color
- **BeadStore** - retains all beads entered
- **NecklaceData** - performs all calcualtions for the creation of a necklace
- **BeadListItem** - Derives each unique bead and the quantity of times it is used in a pattern

- **BeadView** - displays the graphic 10 bead pattern in BeadViewController
- **NecklaceView** - displays the graphic representation of a necklace in repeating the pattern of beads

Preliminary Source Code

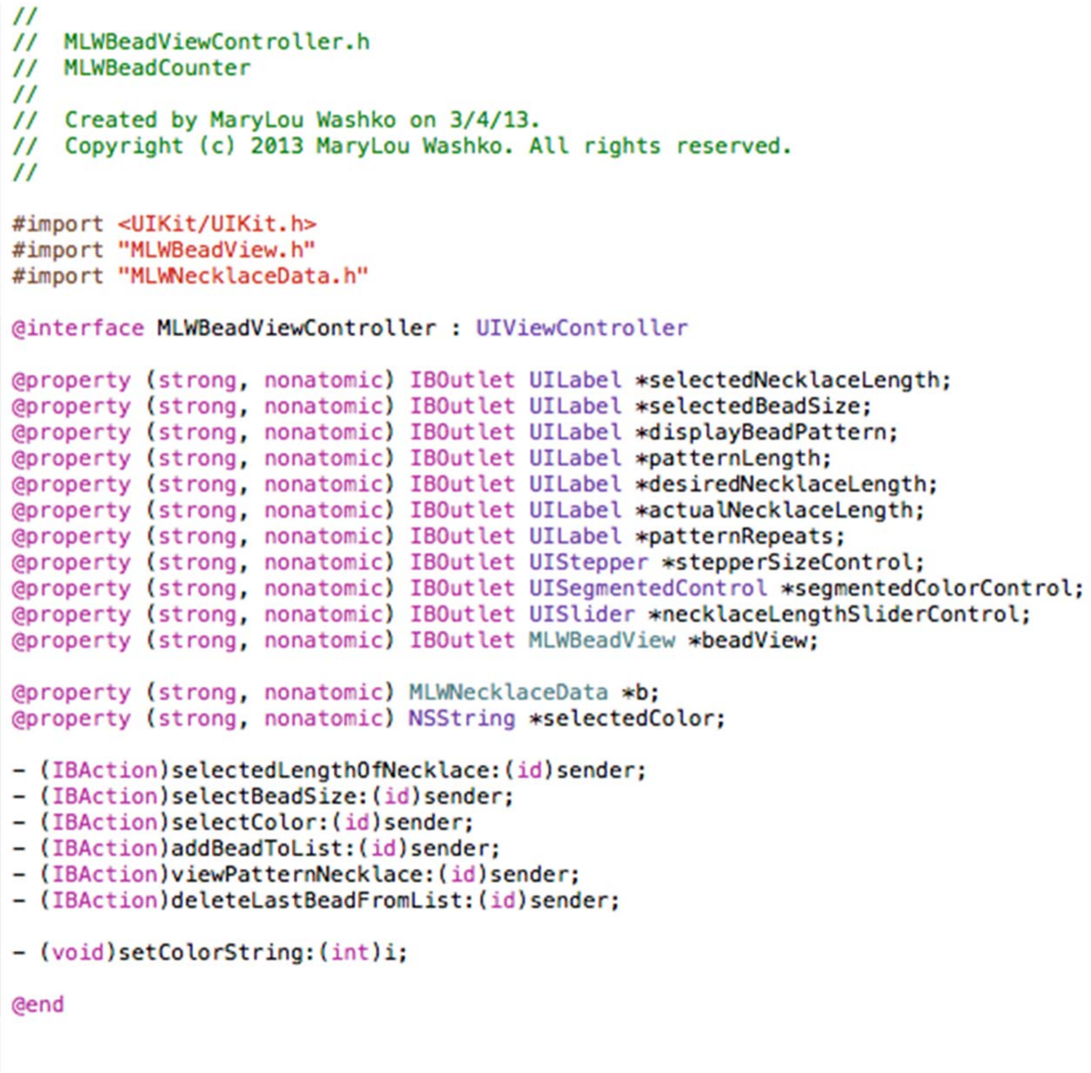


```
//
//  MLWViewController.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MLWViewController : UIViewController
@property (strong, nonatomic) IBOutlet UILabel *titleLabel;

@end
```

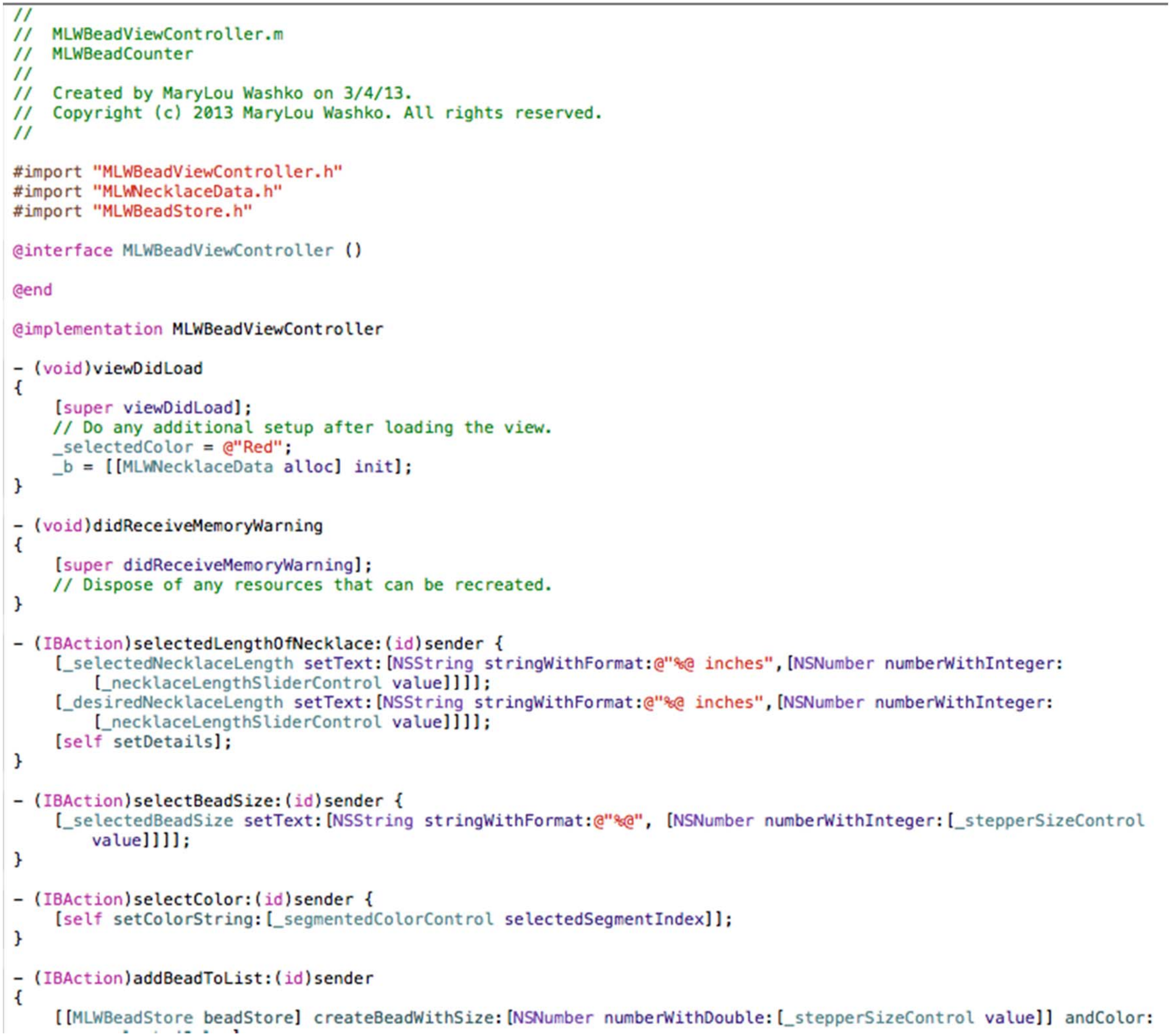


```
//
//  MLWViewController.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWViewController.h"

@interface MLWViewController ()

@end

@implementation MLWViewController

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

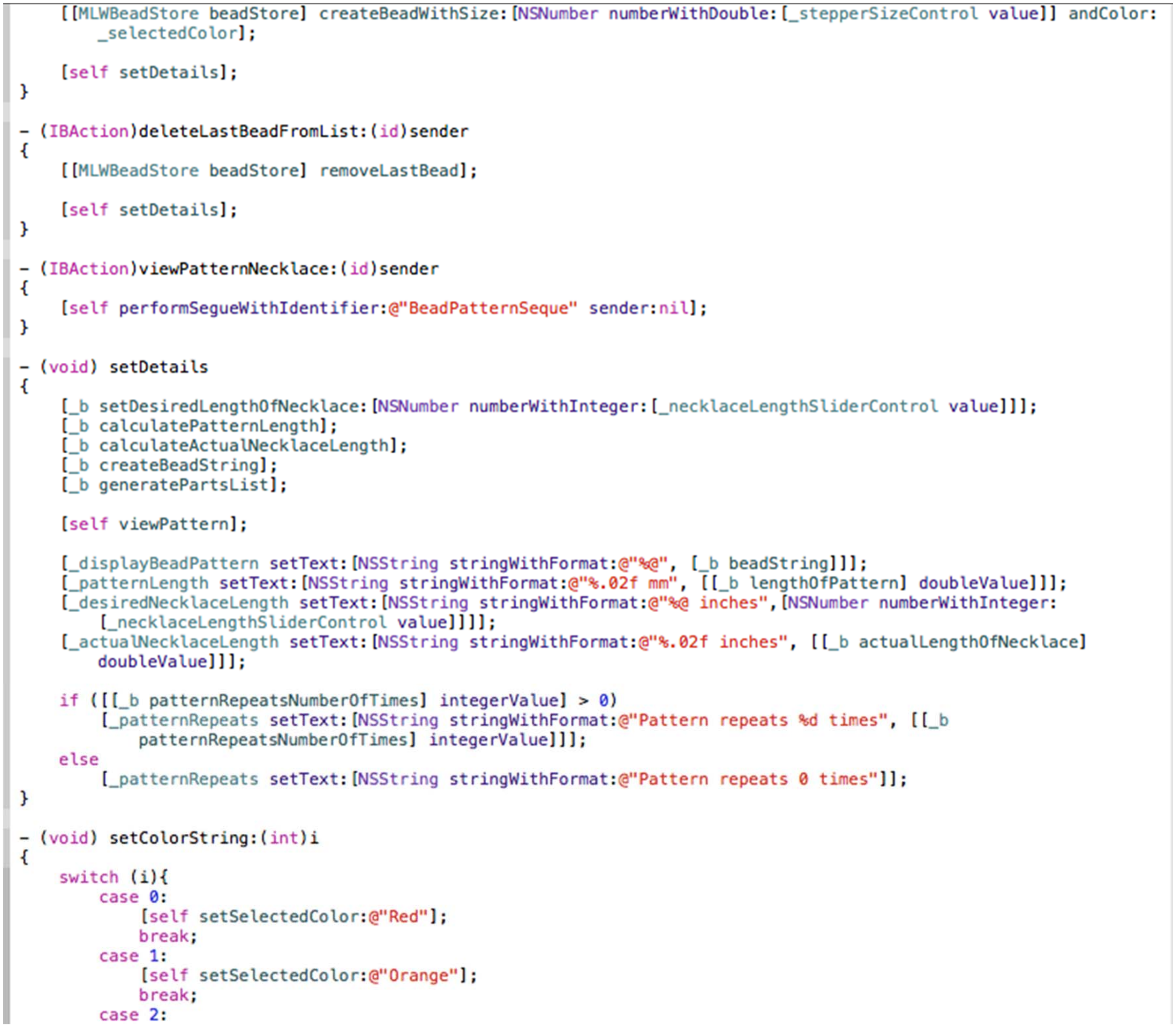
Preliminary Source Code



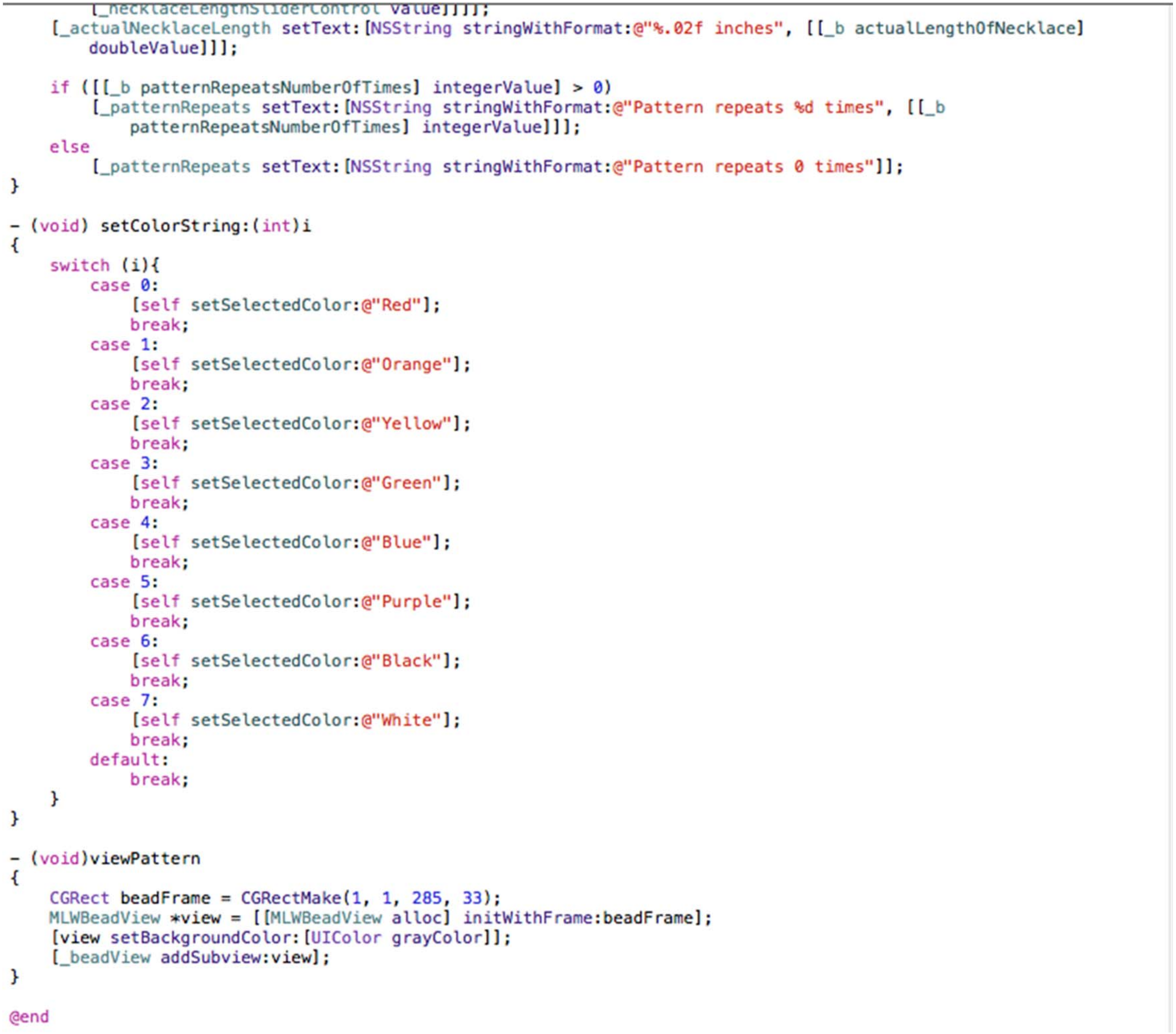
```
//
//  MLWBeadViewController.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "MLWBeadView.h"
#import "MLWNecklaceData.h"

@interface MLWBeadViewController : UIViewController

@property (strong, nonatomic) IBOutlet UILabel *selectedNecklaceLength;
@property (strong, nonatomic) IBOutlet UILabel *selectedBeadSize;
@property (strong, nonatomic) IBOutlet UILabel *displayBeadPattern;
@property (strong, nonatomic) IBOutlet UILabel *patternLength;
@property (strong, nonatomic) IBOutlet UILabel *desiredNecklaceLength;
@property (strong, nonatomic) IBOutlet UILabel *actualNecklaceLength;
@property (strong, nonatomic) IBOutlet UILabel *patternRepeats;
@property (strong, nonatomic) IBOutlet UIStepper *stepperSizeControl;
@property (strong, nonatomic) IBOutlet UISegmentedControl *segmentedColorControl;
@property (strong, nonatomic) IBOutlet UISlider *necklaceLengthSliderControl;
@property (strong, nonatomic) IBOutlet MLWBeadView *beadView;

@property (strong, nonatomic) MLWNecklaceData *b;
@property (strong, nonatomic) NSString *selectedColor;

- (IBAction)selectedLengthOfNecklace:(id)sender;
- (IBAction)selectBeadSize:(id)sender;
- (IBAction)selectColor:(id)sender;
- (IBAction)addBeadToList:(id)sender;
- (IBAction)viewPatternNecklace:(id)sender;
- (IBAction)deleteLastBeadFromList:(id)sender;

- (void)setColorString:(int)i;

@end
```



```
//
//  MLWBeadViewController.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWBeadViewController.h"
#import "MLWNecklaceData.h"
#import "MLWBeadStore.h"

@interface MLWBeadViewController ()

@end

@implementation MLWBeadViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    _selectedColor = @"Red";
    _b = [[MLWNecklaceData alloc] init];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

- (IBAction)selectedLengthOfNecklace:(id)sender {
    [_selectedNecklaceLength setText:[NSString stringWithFormat:@"%@ inches",[NSNumber numberWithInteger:
        [_necklaceLengthSliderControl value]]]];
    [_desiredNecklaceLength setText:[NSString stringWithFormat:@"%@ inches",[NSNumber numberWithInteger:
        [_necklaceLengthSliderControl value]]]];
    [self setDetails];
}

- (IBAction)selectBeadSize:(id)sender {
    [_selectedBeadSize setText:[NSString stringWithFormat:@"%@", [NSNumber numberWithInteger:[_stepperSizeControl
        value]]]];
}

- (IBAction)selectColor:(id)sender {
    [self setColorString:[_segmentedColorControl selectedSegmentIndex]];
}

- (IBAction)addBeadToList:(id)sender
{
    [[MLWBeadStore beadStore] createBeadWithSize:[NSNumber numberWithDouble:[_stepperSizeControl value]] andColor:
```

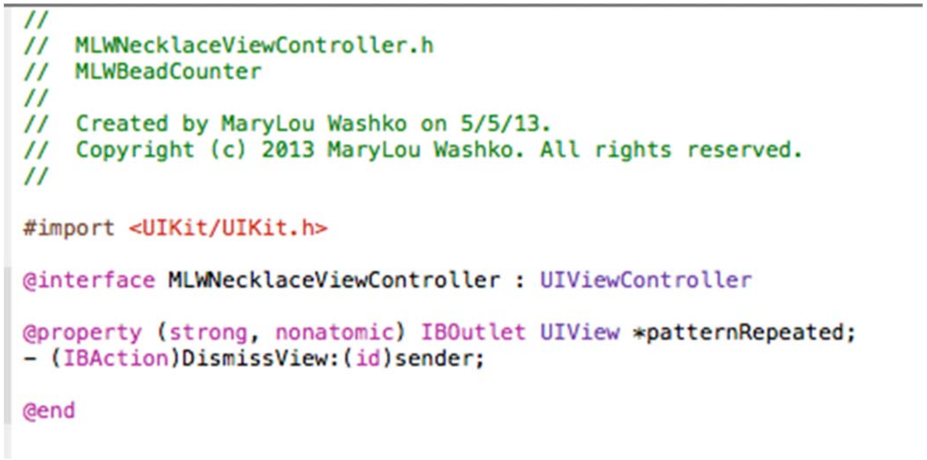


```
    [[MLWBeadStore beadStore] createBeadWithSize:[NSNumber numberWithDouble:[_stepperSizeControl value]] andColor:
        _selectedColor];

    [self setDetails];
}

- (IBAction)deleteLastBeadFromList:(id)sender
{
    [[MLWBeadStore beadStore] removeLastBead];

    [self setDetails];
}

- (IBAction)viewPatternNecklace:(id)sender
{
    [self performSegueWithIdentifier:@"BeadPatternSeque" sender:nil];
}

- (void) setDetails
{
    [_b setDesiredLengthOfNecklace:[NSNumber numberWithInteger:[_necklaceLengthSliderControl value]]];
    [_b calculatePatternLength];
    [_b calculateActualNecklaceLength];
    [_b createBeadString];
    [_b generatePartsList];

    [self viewPattern];

    [_displayBeadPattern setText:[NSString stringWithFormat:@"%@", [_b beadString]]];
    [_patternLength setText:[NSString stringWithFormat:@"%.02f mm", [[_b lengthOfPattern] doubleValue]]];
    [_desiredNecklaceLength setText:[NSString stringWithFormat:@"%@ inches",[NSNumber numberWithInteger:
        [_necklaceLengthSliderControl value]]]];
    [_actualNecklaceLength setText:[NSString stringWithFormat:@"%.02f inches", [[_b actualLengthOfNecklace]
        doubleValue]]];

    if ([[_b patternRepeatsNumberOfTimes] integerValue] > 0)
        [_patternRepeats setText:[NSString stringWithFormat:@"Pattern repeats %d times", [[_b
            patternRepeatsNumberOfTimes] integerValue]]];
    else
        [_patternRepeats setText:[NSString stringWithFormat:@"Pattern repeats 0 times"]];
}

- (void) setColorString:(int)i
{
    switch (i){
        case 0:
            [self setSelectedColor:@"Red"];
            break;
        case 1:
            [self setSelectedColor:@"Orange"];
            break;
        case 2:
```

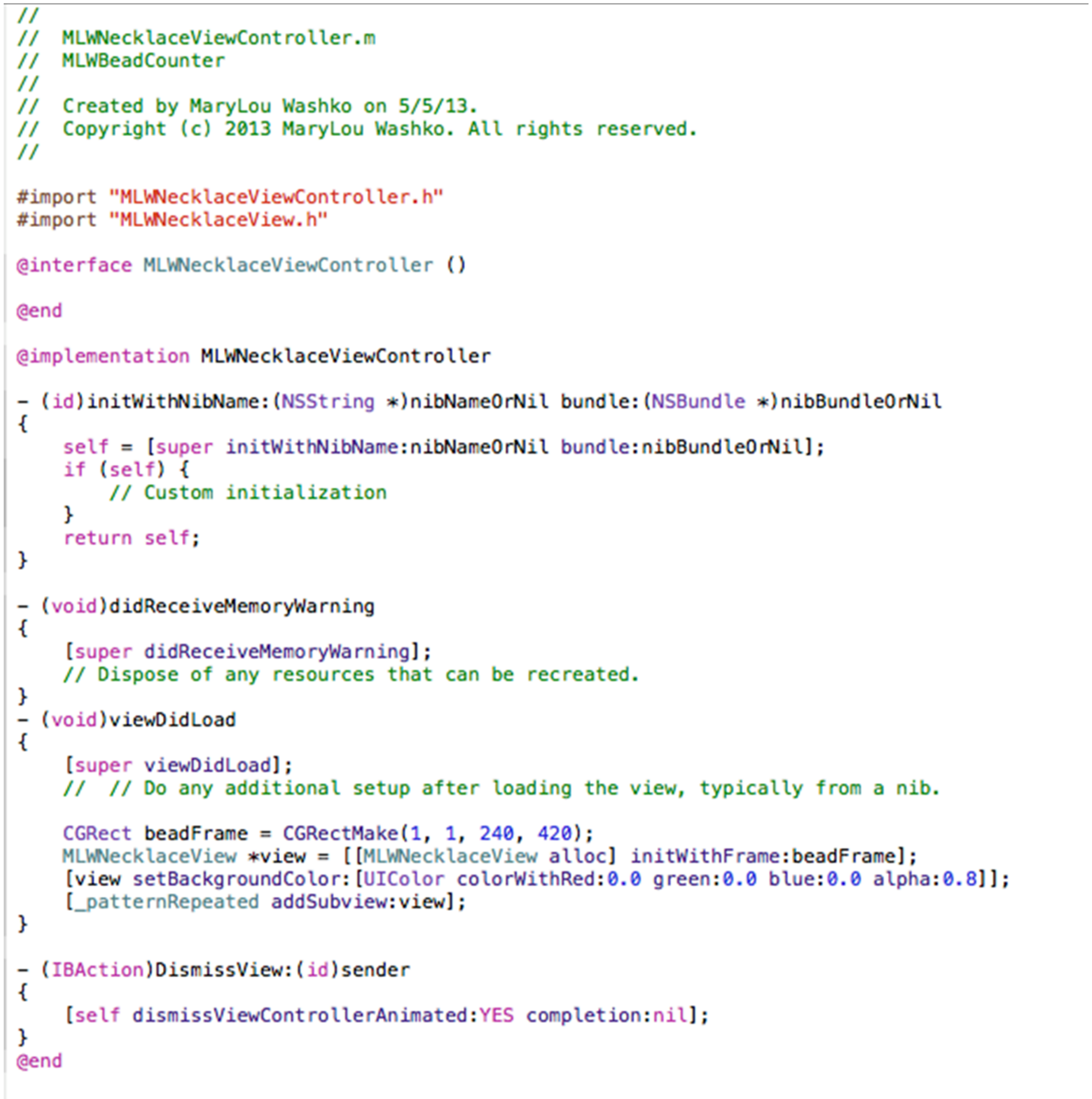


```
            [_necklaceLengthSliderControl value]]]];
    [_actualNecklaceLength setText:[NSString stringWithFormat:@"%.02f inches", [[_b actualLengthOfNecklace]
        doubleValue]]];

    if ([[_b patternRepeatsNumberOfTimes] integerValue] > 0)
        [_patternRepeats setText:[NSString stringWithFormat:@"Pattern repeats %d times", [[_b
            patternRepeatsNumberOfTimes] integerValue]]];
    else
        [_patternRepeats setText:[NSString stringWithFormat:@"Pattern repeats 0 times"]];
}

- (void) setColorString:(int)i
{
    switch (i){
        case 0:
            [self setSelectedColor:@"Red"];
            break;
        case 1:
            [self setSelectedColor:@"Orange"];
            break;
        case 2:
            [self setSelectedColor:@"Yellow"];
            break;
        case 3:
            [self setSelectedColor:@"Green"];
            break;
        case 4:
            [self setSelectedColor:@"Blue"];
            break;
        case 5:
            [self setSelectedColor:@"Purple"];
            break;
        case 6:
            [self setSelectedColor:@"Black"];
            break;
        case 7:
            [self setSelectedColor:@"White"];
            break;
        default:
            break;
    }
}

- (void)viewPattern
{
    CGRect beadFrame = CGRectMake(1, 1, 285, 33);
    MLWBeadView *view = [[MLWBeadView alloc] initWithFrame:beadFrame];
    [view setBackgroundColor:[UIColor grayColor]];
    [_beadView addSubview:view];
}

@end
```

Preliminary Source Code

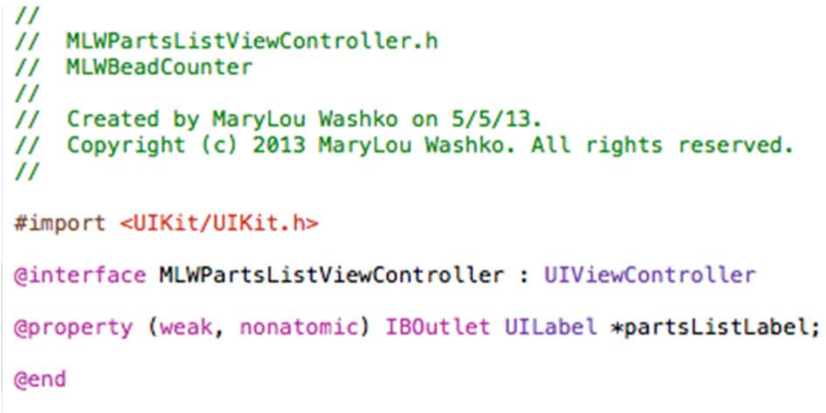


```
//
//  MLWNecklaceViewController.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MLWNecklaceViewController : UIViewController

@property (strong, nonatomic) IBOutlet UIView *patternRepeated;
- (IBAction)DismissView:(id)sender;

@end
```

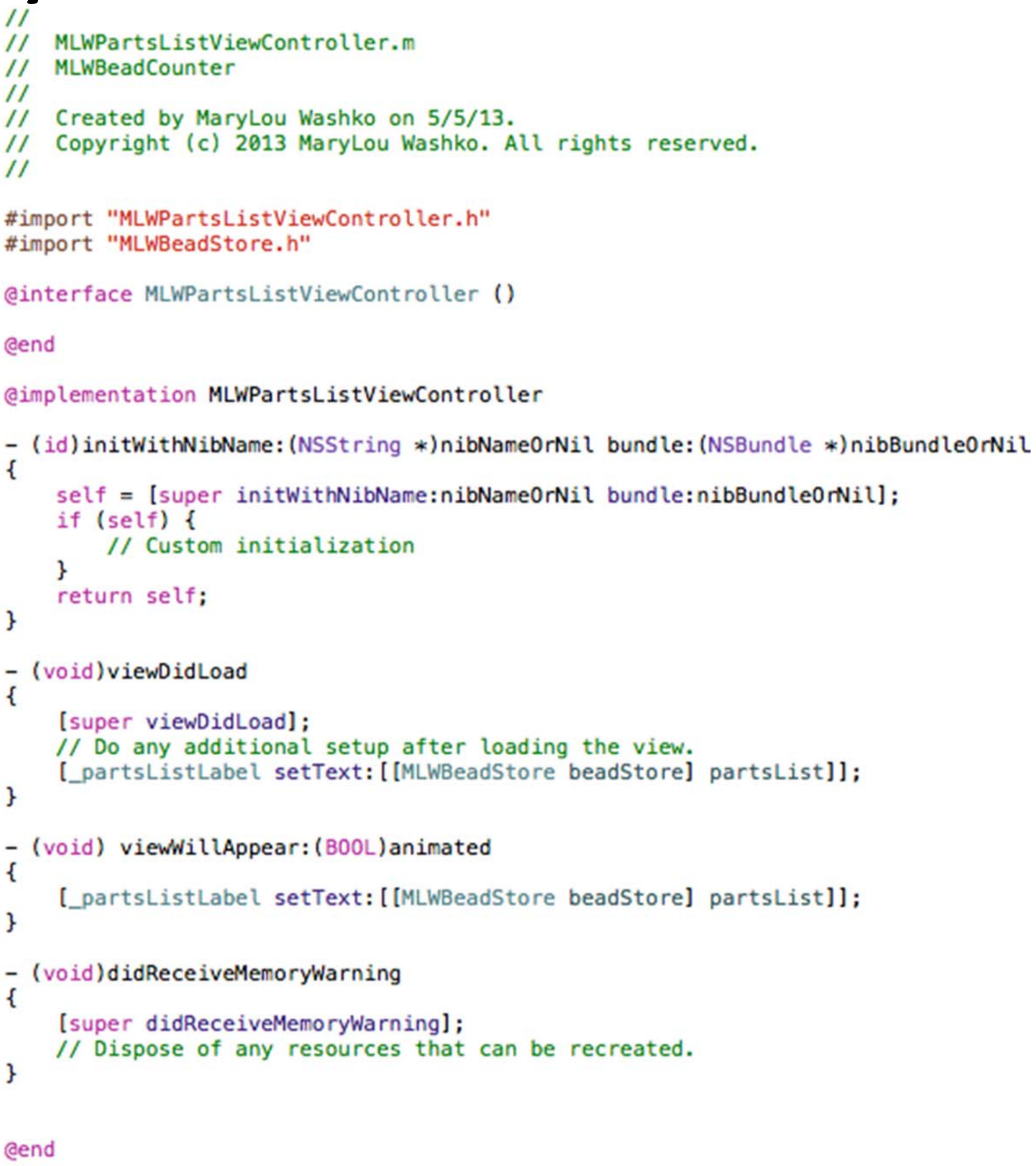


```
//
//  MLWNecklaceViewController.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWNecklaceViewController.h"
#import "MLWNecklaceView.h"

@interface MLWNecklaceViewController ()

@end

@implementation MLWNecklaceViewController

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}
- (void)viewDidLoad
{
    [super viewDidLoad];
    //  // Do any additional setup after loading the view, typically from a nib.

    CGRect beadFrame = CGRectMake(1, 1, 240, 420);
    MLWNecklaceView *view = [[MLWNecklaceView alloc] initWithFrame:beadFrame];
    [view setBackgroundColor:[UIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.8]];
    [_patternRepeated addSubview:view];
}

- (IBAction)DismissView:(id)sender
{
    [self dismissViewControllerAnimated:YES completion:nil];
}
@end
```

Preliminary Source Code

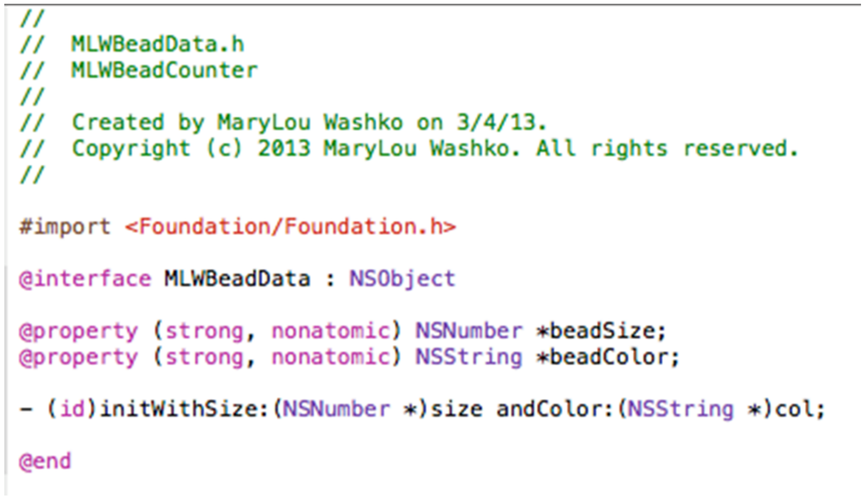


```
//
//  MLWPartsListViewController.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MLWPartsListViewController : UIViewController

@property (weak, nonatomic) IBOutlet UILabel *partsListLabel;

@end
```

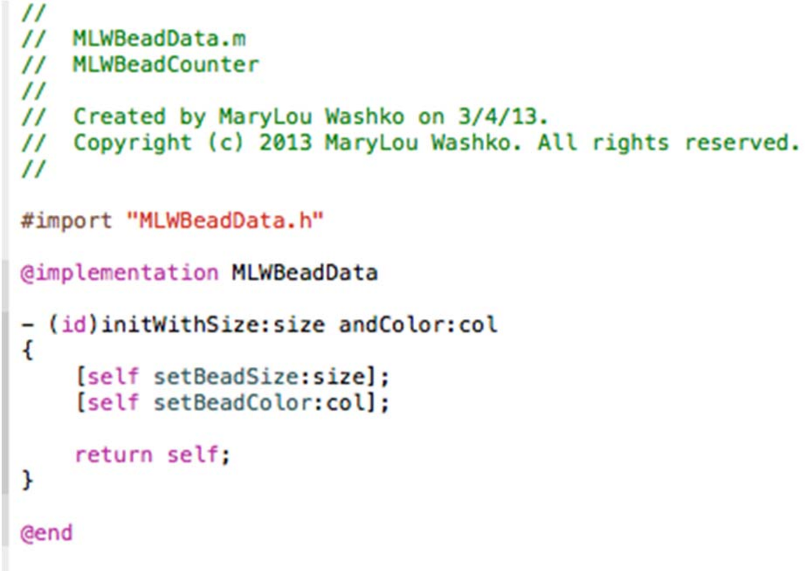


```
//
//  MLWPartsListViewController.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWPartsListViewController.h"
#import "MLWBeadStore.h"

@interface MLWPartsListViewController ()

@end

@implementation MLWPartsListViewController

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
    if (self) {
        // Custom initialization
    }
    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    [_partsListLabel setText:[[MLWBeadStore beadStore] partsList]];
}

- (void) viewWillAppear:(BOOL)animated
{
    [_partsListLabel setText:[[MLWBeadStore beadStore] partsList]];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}


@end
```

Preliminary Source Code

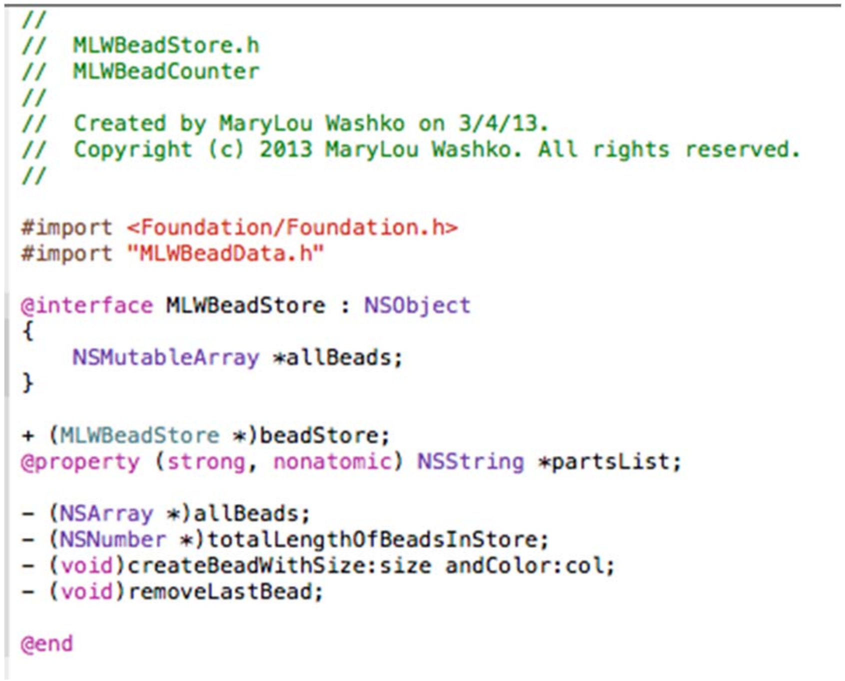


```
//
//  MLWBeadData.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface MLWBeadData : NSObject

@property (strong, nonatomic) NSNumber *beadSize;
@property (strong, nonatomic) NSString *beadColor;

- (id)initWithSize:(NSNumber *)size andColor:(NSString *)col;

@end
```

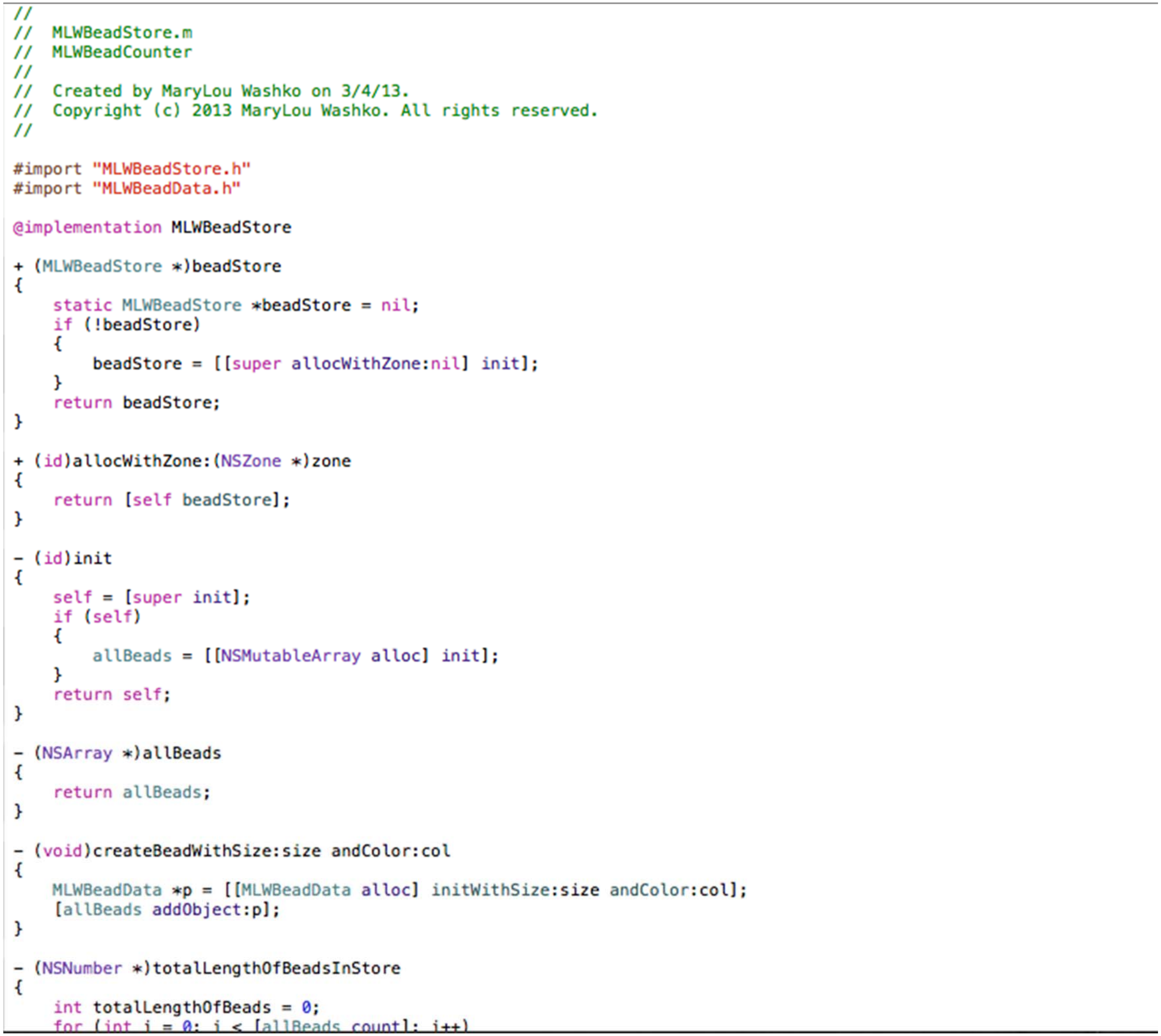


```
//
//  MLWBeadData.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWBeadData.h"

@implementation MLWBeadData

- (id)initWithSize:size andColor:col
{
    [self setBeadSize:size];
    [self setBeadColor:col];

    return self;
}

@end
```

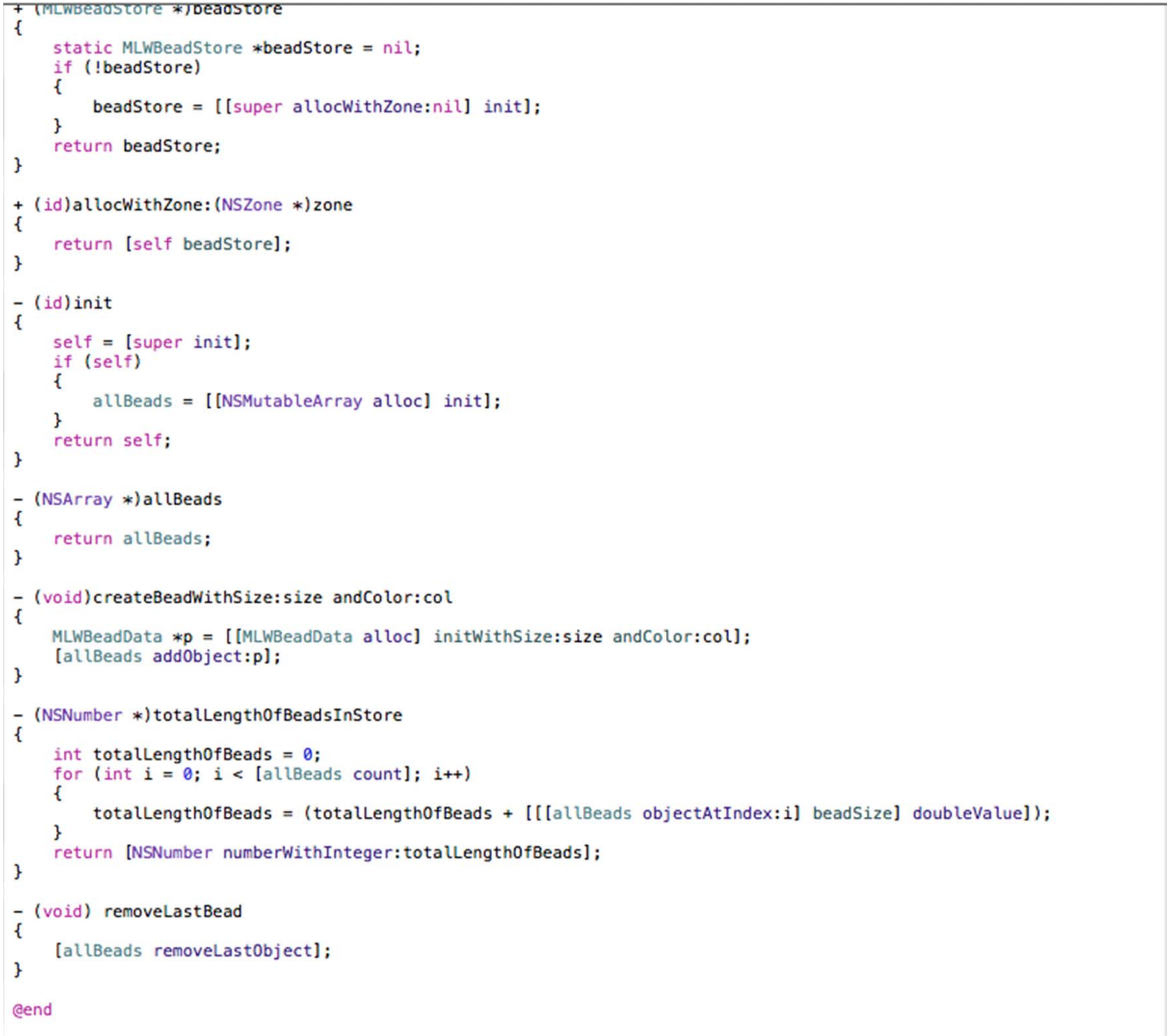
Preliminary Source Code



```
//
//  MLWBeadStore.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <Foundation/Foundation.h>
#import "MLWBeadData.h"

@interface MLWBeadStore : NSObject
{
    NSMutableArray *allBeads;
}

+ (MLWBeadStore *)beadStore;
@property (strong, nonatomic) NSString *partsList;

- (NSArray *)allBeads;
- (NSNumber *)totalLengthOfBeadsInStore;
- (void)createBeadWithSize:size andColor:col;
- (void)removeLastBead;

@end
```

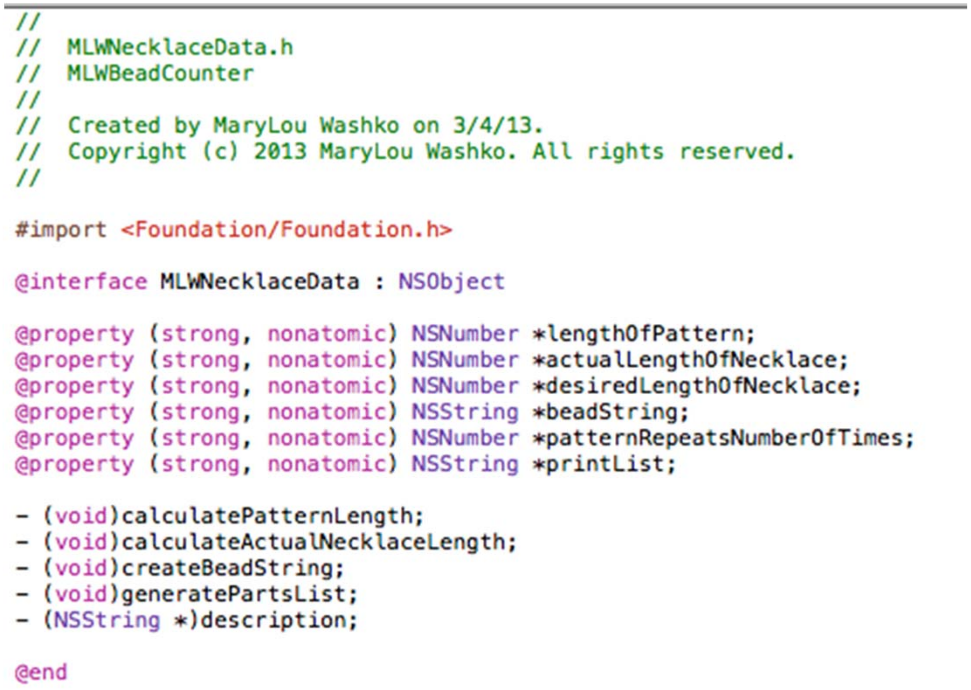


```
//
//  MLWBeadStore.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWBeadStore.h"
#import "MLWBeadData.h"

@implementation MLWBeadStore

+ (MLWBeadStore *)beadStore
{
    static MLWBeadStore *beadStore = nil;
    if (!beadStore)
    {
        beadStore = [[super allocWithZone:nil] init];
    }
    return beadStore;
}

+ (id)allocWithZone:(NSZone *)zone
{
    return [self beadStore];
}

- (id)init
{
    self = [super init];
    if (self)
    {
        allBeads = [[NSMutableArray alloc] init];
    }
    return self;
}

- (NSArray *)allBeads
{
    return allBeads;
}

- (void)createBeadWithSize:size andColor:col
{
    MLWBeadData *p = [[MLWBeadData alloc] initWithSize:size andColor:col];
    [allBeads addObject:p];
}

- (NSNumber *)totalLengthOfBeadsInStore
{
    int totalLengthOfBeads = 0;
    for (int i = 0; i < [allBeads count]; i++)
```

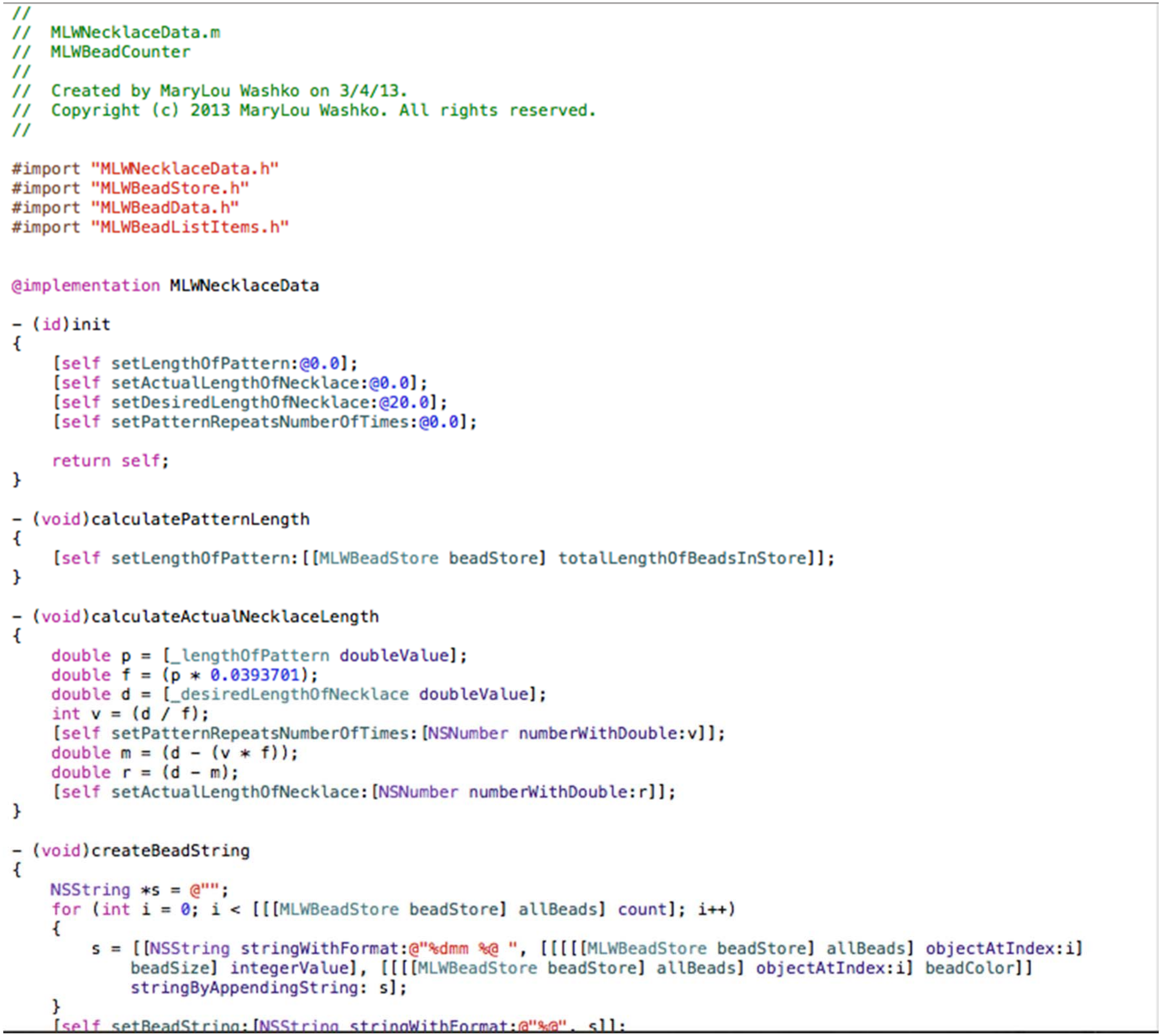


```
+ (MLWBeadStore *)beadStore
{
    static MLWBeadStore *beadStore = nil;
    if (!beadStore)
    {
        beadStore = [[super allocWithZone:nil] init];
    }
    return beadStore;
}

+ (id)allocWithZone:(NSZone *)zone
{
    return [self beadStore];
}

- (id)init
{
    self = [super init];
    if (self)
    {
        allBeads = [[NSMutableArray alloc] init];
    }
    return self;
}

- (NSArray *)allBeads
{
    return allBeads;
}

- (void)createBeadWithSize:size andColor:col
{
    MLWBeadData *p = [[MLWBeadData alloc] initWithSize:size andColor:col];
    [allBeads addObject:p];
}

- (NSNumber *)totalLengthOfBeadsInStore
{
    int totalLengthOfBeads = 0;
    for (int i = 0; i < [allBeads count]; i++)
    {
        totalLengthOfBeads = (totalLengthOfBeads + [[[allBeads objectAtIndex:i] beadSize] doubleValue]);
    }
    return [NSNumber numberWithInteger:totalLengthOfBeads];
}

- (void) removeLastBead
{
    [allBeads removeLastObject];
}

@end
```

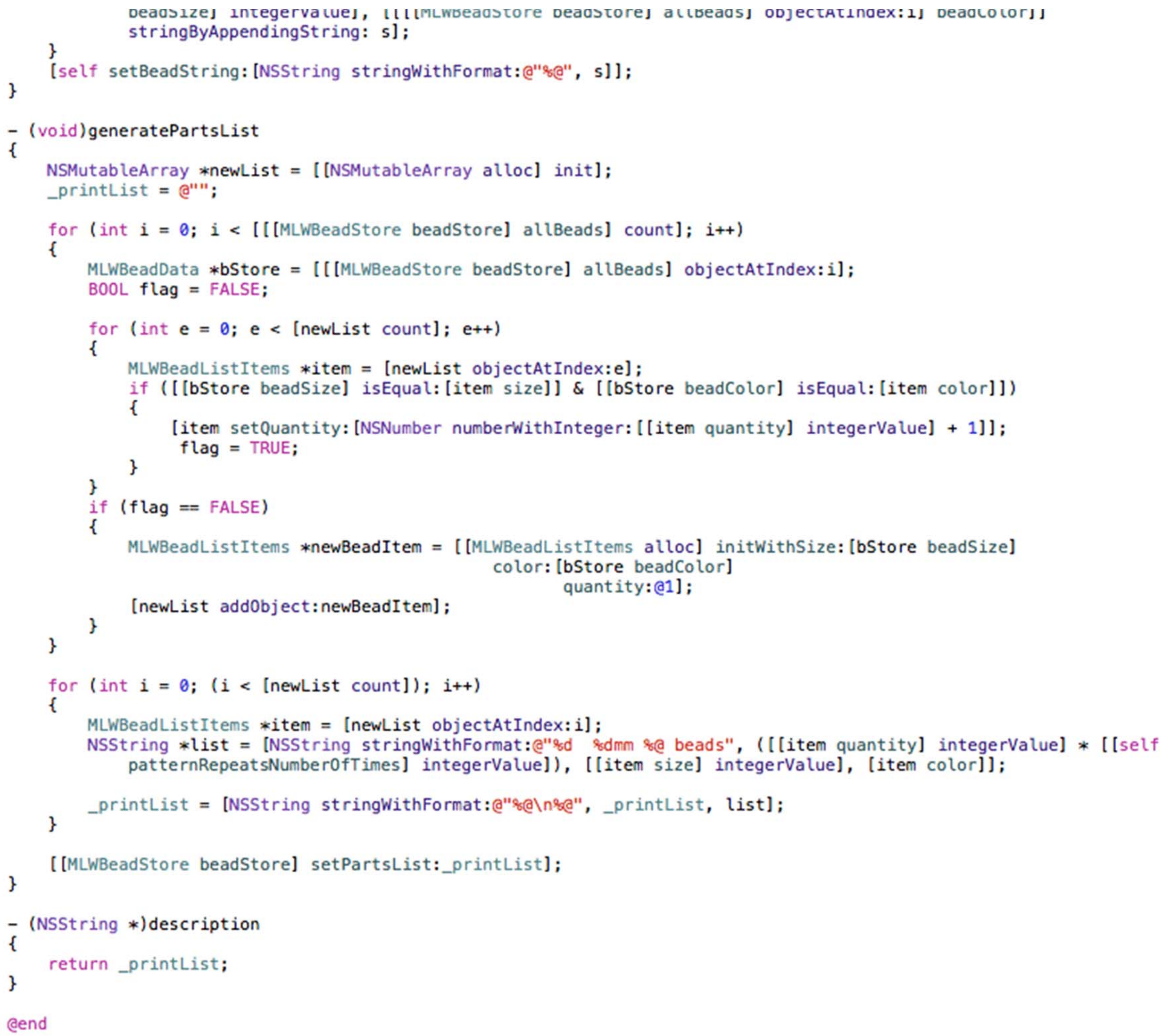
Preliminary Source Code



```
//
//  MLWNecklaceData.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface MLWNecklaceData : NSObject

@property (strong, nonatomic) NSNumber *lengthOfPattern;
@property (strong, nonatomic) NSNumber *actualLengthOfNecklace;
@property (strong, nonatomic) NSNumber *desiredLengthOfNecklace;
@property (strong, nonatomic) NSString *beadString;
@property (strong, nonatomic) NSNumber *patternRepeatsNumberOfTimes;
@property (strong, nonatomic) NSString *printList;

- (void)calculatePatternLength;
- (void)calculateActualNecklaceLength;
- (void)createBeadString;
- (void)generatePartsList;
- (NSString *)description;

@end
```

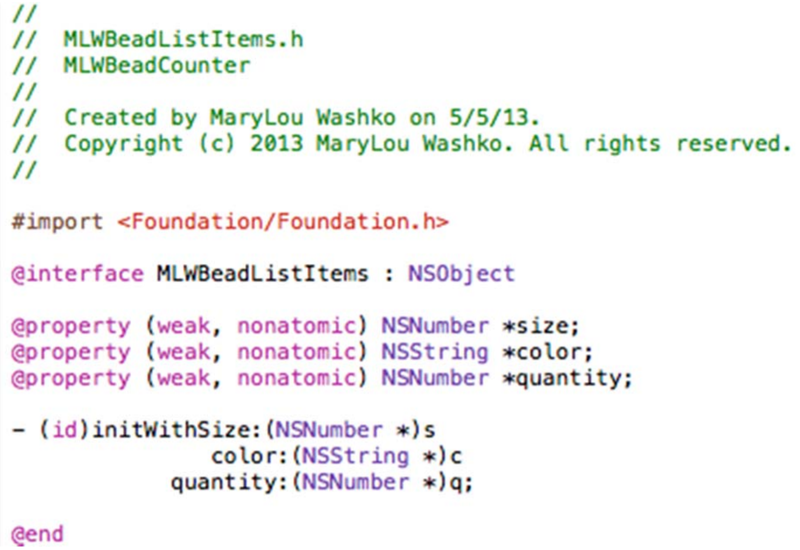


```
//
//  MLWNecklaceData.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 3/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWNecklaceData.h"
#import "MLWBeadStore.h"
#import "MLWBeadData.h"
#import "MLWBeadListItems.h"


@implementation MLWNecklaceData

- (id)init
{
    [self setLengthOfPattern:@0.0];
    [self setActualLengthOfNecklace:@0.0];
    [self setDesiredLengthOfNecklace:@20.0];
    [self setPatternRepeatsNumberOfTimes:@0.0];

    return self;
}

- (void)calculatePatternLength
{
    [self setLengthOfPattern:[[MLWBeadStore beadStore] totalLengthOfBeadsInStore]];
}

- (void)calculateActualNecklaceLength
{
    double p = [_lengthOfPattern doubleValue];
    double f = (p * 0.0393701);
    double d = [_desiredLengthOfNecklace doubleValue];
    int v = (d / f);
    [self setPatternRepeatsNumberOfTimes:[NSNumber numberWithDouble:v]];
    double m = (d - (v * f));
    double r = (d - m);
    [self setActualLengthOfNecklace:[NSNumber numberWithDouble:r]];
}

- (void)createBeadString
{
    NSString *s = @"";
    for (int i = 0; i < [[[MLWBeadStore beadStore] allBeads] count]; i++)
    {
        s = [[NSString stringWithFormat:@"%dmm %@ ", [[[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i]
            beadSize] integerValue], [[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i] beadColor]]
            stringByAppendingString: s];
    }
    [self setBeadString:[NSString stringWithFormat:@"%@", s]];
```

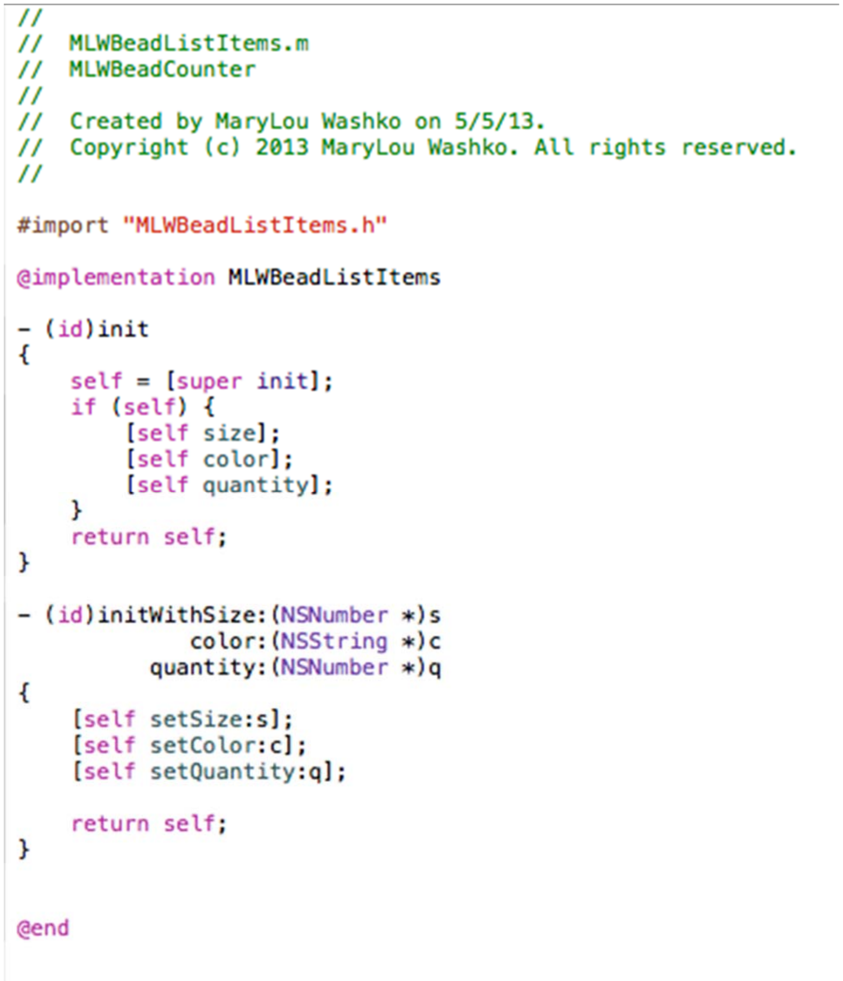


```
            beadSize] integerValue], [[[MLWBeadStore beadStore] allBeads] objectAtIndex:i] beadColor]]
            stringByAppendingString: s];
    }
    [self setBeadString:[NSString stringWithFormat:@"%@", s]];
}

- (void)generatePartsList
{
    NSMutableArray *newList = [[NSMutableArray alloc] init];
    _printList = @"";

    for (int i = 0; i < [[[MLWBeadStore beadStore] allBeads] count]; i++)
    {
        MLWBeadData *bStore = [[[MLWBeadStore beadStore] allBeads] objectAtIndex:i];
        BOOL flag = FALSE;

        for (int e = 0; e < [newList count]; e++)
        {
            MLWBeadListItems *item = [newList objectAtIndex:e];
            if ([[bStore beadSize] isEqual:[item size]] & [[bStore beadColor] isEqual:[item color]])
            {
                [item setQuantity:[NSNumber numberWithInteger:[[item quantity] integerValue] + 1]];
                 flag = TRUE;
            }
        }
        if (flag == FALSE)
        {
            MLWBeadListItems *newBeadItem = [[MLWBeadListItems alloc] initWithSize:[bStore beadSize]
                                                color:[bStore beadColor]
                                                       quantity:@1];
            [newList addObject:newBeadItem];
        }
    }

    for (int i = 0; (i < [newList count]); i++)
    {
        MLWBeadListItems *item = [newList objectAtIndex:i];
        NSString *list = [NSString stringWithFormat:@"%d  %dmm %@ beads", ([[item quantity] integerValue] * [[self
            patternRepeatsNumberOfTimes] integerValue]), [[item size] integerValue], [item color]];

        _printList = [NSString stringWithFormat:@"%@\n%@", _printList, list];
    }

    [[MLWBeadStore beadStore] setPartsList:_printList];
}

- (NSString *)description
{
    return _printList;
}

@end
```

Preliminary Source Code

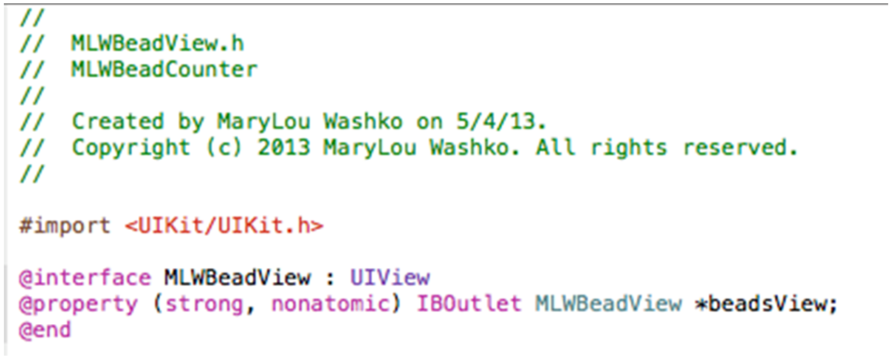


```
//
//  MLWBeadListItems.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface MLWBeadListItems : NSObject

@property (weak, nonatomic) NSNumber *size;
@property (weak, nonatomic) NSString *color;
@property (weak, nonatomic) NSNumber *quantity;

- (id)initWithSize:(NSNumber *)s
               color:(NSString *)c
            quantity:(NSNumber *)q;

@end
```

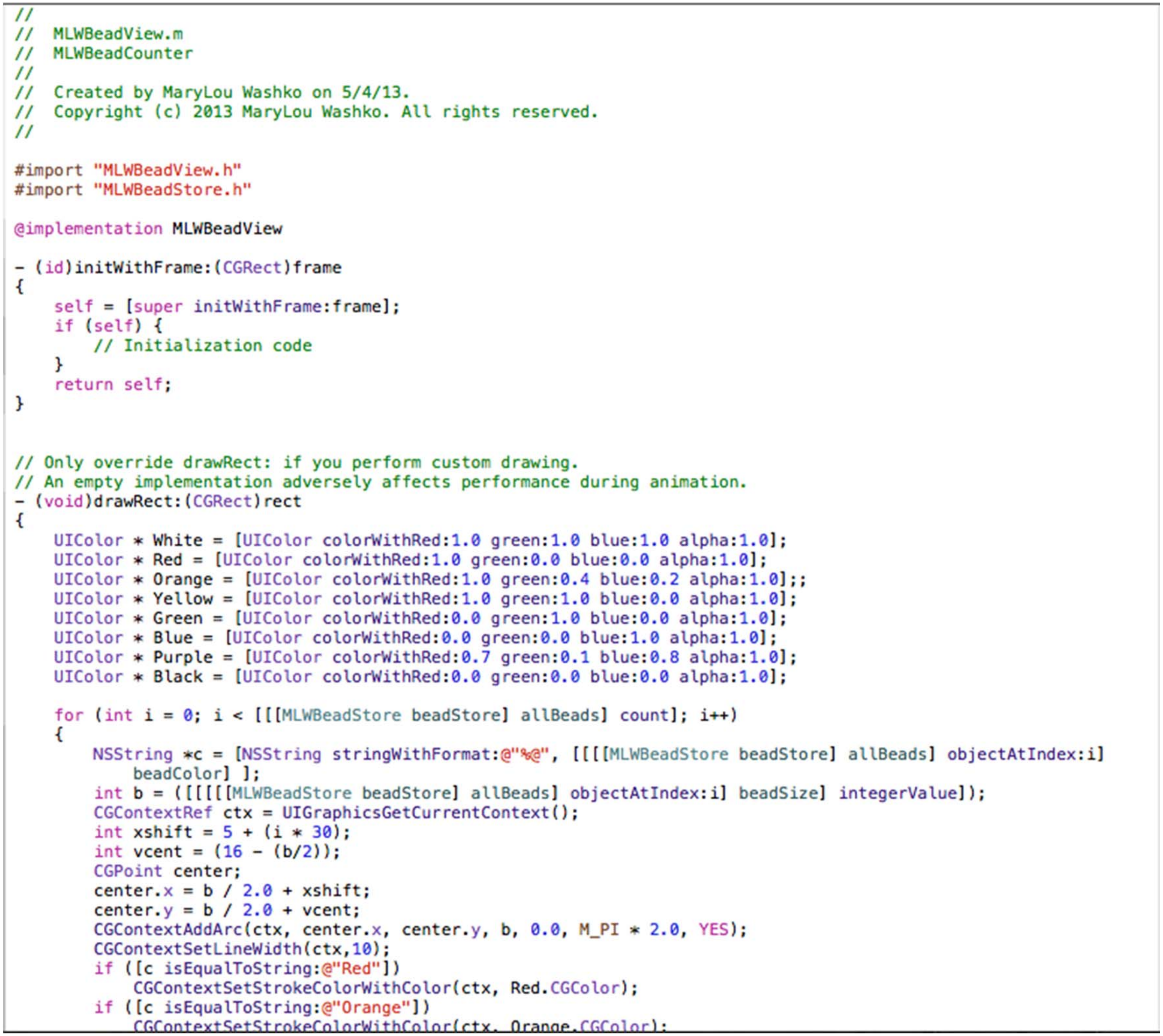


```
//
//  MLWBeadListItems.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWBeadListItems.h"

@implementation MLWBeadListItems

- (id)init
{
    self = [super init];
    if (self) {
        [self size];
        [self color];
        [self quantity];
    }
    return self;
}

- (id)initWithSize:(NSNumber *)s
               color:(NSString *)c
            quantity:(NSNumber *)q
{
    [self setSize:s];
    [self setColor:c];
    [self setQuantity:q];

    return self;
}


@end
```

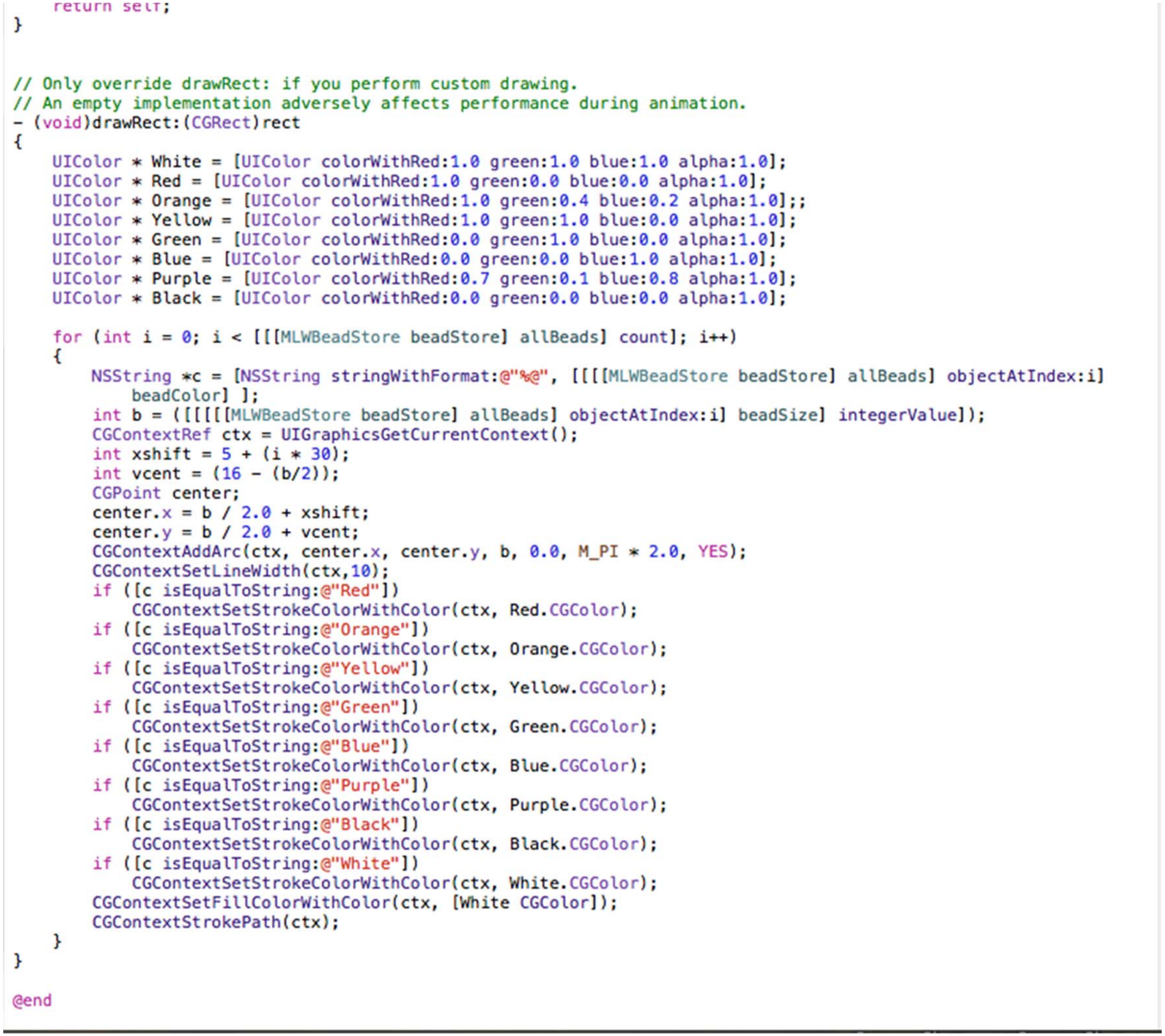
Preliminary Source Code



```
//
//  MLWBeadView.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MLWBeadView : UIView
@property (strong, nonatomic) IBOutlet MLWBeadView *beadsView;
@end
```

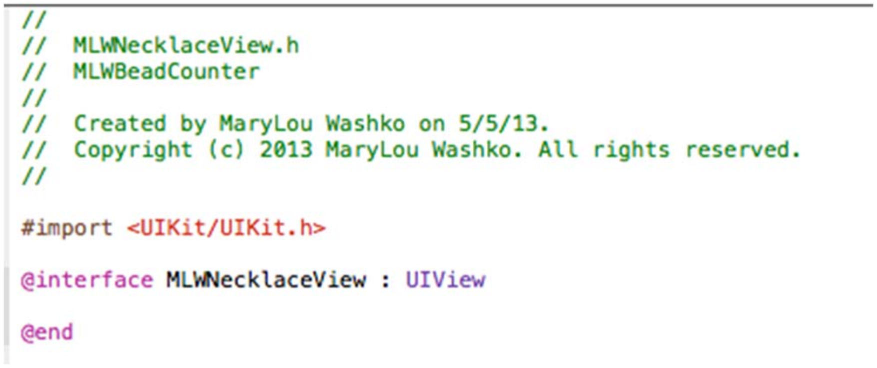


```
//
//  MLWBeadView.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/4/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWBeadView.h"
#import "MLWBeadStore.h"

@implementation MLWBeadView

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self) {
        // Initialization code
    }
    return self;
}


// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during animation.
- (void)drawRect:(CGRect)rect
{
    UIColor * White = [UIColor colorWithRed:1.0 green:1.0 blue:1.0 alpha:1.0];
    UIColor * Red = [UIColor colorWithRed:1.0 green:0.0 blue:0.0 alpha:1.0];
    UIColor * Orange = [UIColor colorWithRed:1.0 green:0.4 blue:0.2 alpha:1.0];;
    UIColor * Yellow = [UIColor colorWithRed:1.0 green:1.0 blue:0.0 alpha:1.0];
    UIColor * Green = [UIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0];
    UIColor * Blue = [UIColor colorWithRed:0.0 green:0.0 blue:1.0 alpha:1.0];
    UIColor * Purple = [UIColor colorWithRed:0.7 green:0.1 blue:0.8 alpha:1.0];
    UIColor * Black = [UIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:1.0];

    for (int i = 0; i < [[[MLWBeadStore beadStore] allBeads] count]; i++)
    {
        NSString *c = [NSString stringWithFormat:@"%@", [[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i]
            beadColor] ];
        int b = ([[[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i] beadSize] integerValue]);
        CGContextRef ctx = UIGraphicsGetCurrentContext();
        int xshift = 5 + (i * 30);
        int vcent = (16 - (b/2));
        CGPoint center;
        center.x = b / 2.0 + xshift;
        center.y = b / 2.0 + vcent;
        CGContextAddArc(ctx, center.x, center.y, b, 0.0, M_PI * 2.0, YES);
        CGContextSetLineWidth(ctx,10);
        if ([c isEqualToString:@"Red"])
            CGContextSetStrokeColorWithColor(ctx, Red.CGColor);
        if ([c isEqualToString:@"Orange"])
            CGContextSetStrokeColorWithColor(ctx, Orange.CGColor);
```

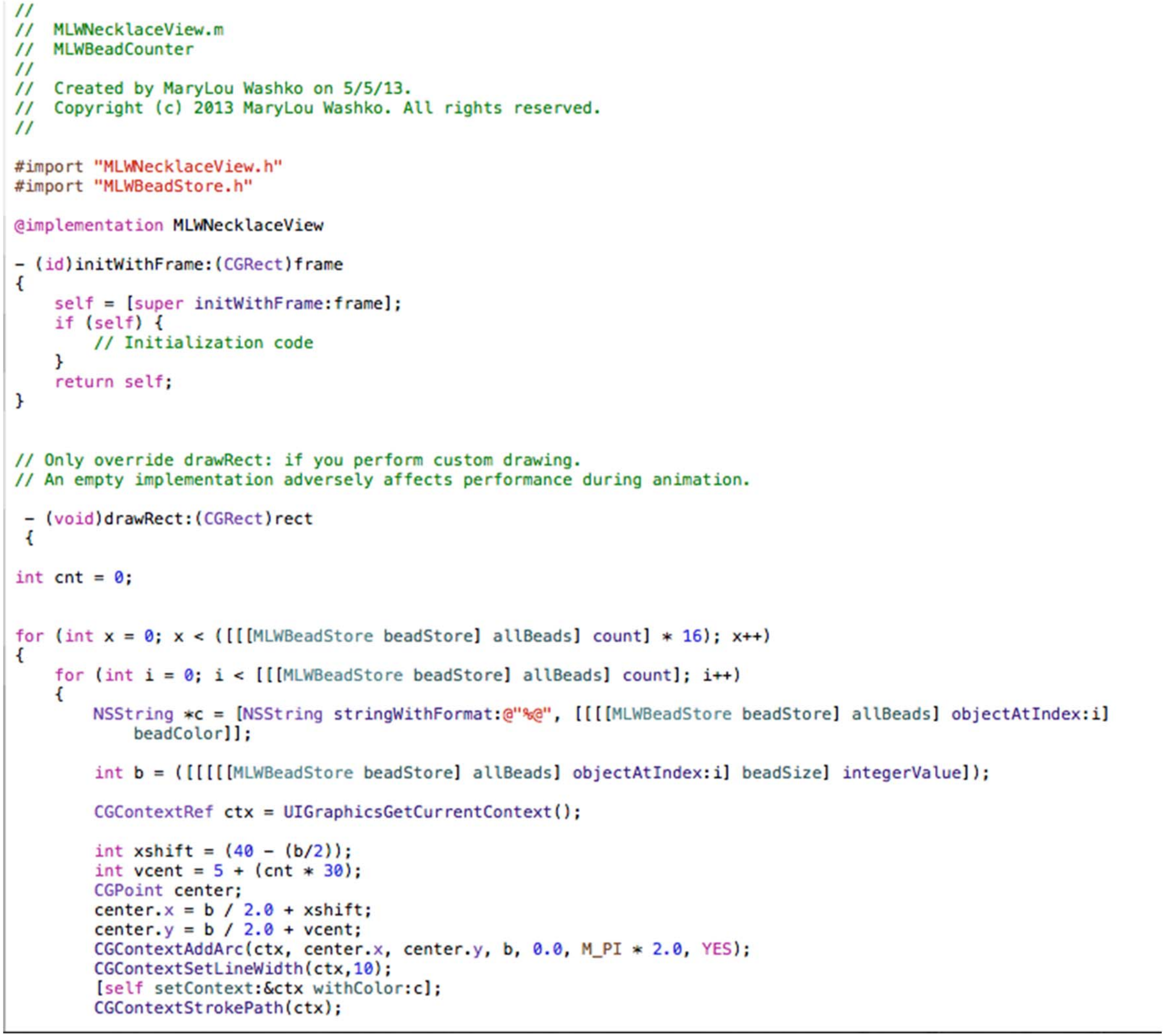


```
    return self;
}


// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during animation.
- (void)drawRect:(CGRect)rect
{
    UIColor * White = [UIColor colorWithRed:1.0 green:1.0 blue:1.0 alpha:1.0];
    UIColor * Red = [UIColor colorWithRed:1.0 green:0.0 blue:0.0 alpha:1.0];
    UIColor * Orange = [UIColor colorWithRed:1.0 green:0.4 blue:0.2 alpha:1.0];;
    UIColor * Yellow = [UIColor colorWithRed:1.0 green:1.0 blue:0.0 alpha:1.0];
    UIColor * Green = [UIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0];
    UIColor * Blue = [UIColor colorWithRed:0.0 green:0.0 blue:1.0 alpha:1.0];
    UIColor * Purple = [UIColor colorWithRed:0.7 green:0.1 blue:0.8 alpha:1.0];
    UIColor * Black = [UIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:1.0];

    for (int i = 0; i < [[[MLWBeadStore beadStore] allBeads] count]; i++)
    {
        NSString *c = [NSString stringWithFormat:@"%@", [[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i]
            beadColor] ];
        int b = ([[[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i] beadSize] integerValue]);
        CGContextRef ctx = UIGraphicsGetCurrentContext();
        int xshift = 5 + (i * 30);
        int vcent = (16 - (b/2));
        CGPoint center;
        center.x = b / 2.0 + xshift;
        center.y = b / 2.0 + vcent;
        CGContextAddArc(ctx, center.x, center.y, b, 0.0, M_PI * 2.0, YES);
        CGContextSetLineWidth(ctx,10);
        if ([c isEqualToString:@"Red"])
            CGContextSetStrokeColorWithColor(ctx, Red.CGColor);
        if ([c isEqualToString:@"Orange"])
            CGContextSetStrokeColorWithColor(ctx, Orange.CGColor);
        if ([c isEqualToString:@"Yellow"])
            CGContextSetStrokeColorWithColor(ctx, Yellow.CGColor);
        if ([c isEqualToString:@"Green"])
            CGContextSetStrokeColorWithColor(ctx, Green.CGColor);
        if ([c isEqualToString:@"Blue"])
            CGContextSetStrokeColorWithColor(ctx, Blue.CGColor);
        if ([c isEqualToString:@"Purple"])
            CGContextSetStrokeColorWithColor(ctx, Purple.CGColor);
        if ([c isEqualToString:@"Black"])
            CGContextSetStrokeColorWithColor(ctx, Black.CGColor);
        if ([c isEqualToString:@"White"])
            CGContextSetStrokeColorWithColor(ctx, White.CGColor);
        CGContextSetFillColorWithColor(ctx, [White CGColor]);
        CGContextStrokePath(ctx);
    }
}

@end
```

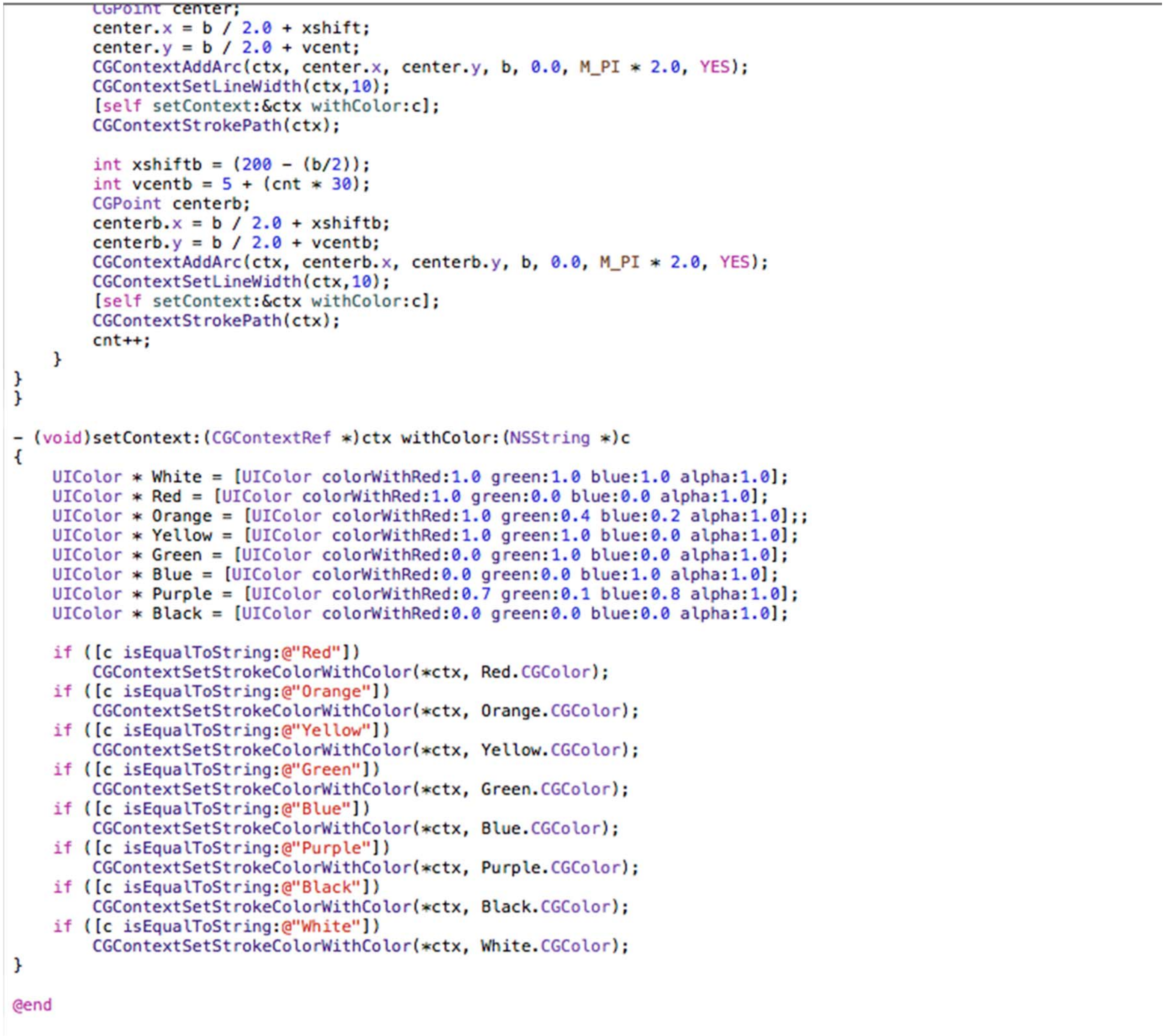
Preliminary Source Code



```
//
//  MLWNecklaceView.h
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MLWNecklaceView : UIView

@end
```



```
//
//  MLWNecklaceView.m
//  MLWBeadCounter
//
//  Created by MaryLou Washko on 5/5/13.
//  Copyright (c) 2013 MaryLou Washko. All rights reserved.
//

#import "MLWNecklaceView.h"
#import "MLWBeadStore.h"

@implementation MLWNecklaceView

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self) {
        // Initialization code
    }
    return self;
}


// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during animation.

 - (void)drawRect:(CGRect)rect
 {

int cnt = 0;


for (int x = 0; x < ([[[MLWBeadStore beadStore] allBeads] count] * 16); x++)
{
    for (int i = 0; i < [[[MLWBeadStore beadStore] allBeads] count]; i++)
    {
        NSString *c = [NSString stringWithFormat:@"%@", [[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i]
            beadColor]];

        int b = ([[[[[MLWBeadStore beadStore] allBeads] objectAtIndex:i] beadSize] integerValue]);

        CGContextRef ctx = UIGraphicsGetCurrentContext();

        int xshift = (40 - (b/2));
        int vcent = 5 + (cnt * 30);
        CGPoint center;
        center.x = b / 2.0 + xshift;
        center.y = b / 2.0 + vcent;
        CGContextAddArc(ctx, center.x, center.y, b, 0.0, M_PI * 2.0, YES);
        CGContextSetLineWidth(ctx,10);
        [self setContext:&ctx withColor:c];
        CGContextStrokePath(ctx);
```



```
        CGPoint center;
        center.x = b / 2.0 + xshift;
        center.y = b / 2.0 + vcent;
        CGContextAddArc(ctx, center.x, center.y, b, 0.0, M_PI * 2.0, YES);
        CGContextSetLineWidth(ctx,10);
        [self setContext:&ctx withColor:c];
        CGContextStrokePath(ctx);

        int xshiftb = (200 - (b/2));
        int vcentb = 5 + (cnt * 30);
        CGPoint centerb;
        centerb.x = b / 2.0 + xshiftb;
        centerb.y = b / 2.0 + vcentb;
        CGContextAddArc(ctx, centerb.x, centerb.y, b, 0.0, M_PI * 2.0, YES);
        CGContextSetLineWidth(ctx,10);
        [self setContext:&ctx withColor:c];
        CGContextStrokePath(ctx);
        cnt++;
    }
}
}

- (void)setContext:(CGContextRef *)ctx withColor:(NSString *)c
{
    UIColor * White = [UIColor colorWithRed:1.0 green:1.0 blue:1.0 alpha:1.0];
    UIColor * Red = [UIColor colorWithRed:1.0 green:0.0 blue:0.0 alpha:1.0];
    UIColor * Orange = [UIColor colorWithRed:1.0 green:0.4 blue:0.2 alpha:1.0];;
    UIColor * Yellow = [UIColor colorWithRed:1.0 green:1.0 blue:0.0 alpha:1.0];
    UIColor * Green = [UIColor colorWithRed:0.0 green:1.0 blue:0.0 alpha:1.0];
    UIColor * Blue = [UIColor colorWithRed:0.0 green:0.0 blue:1.0 alpha:1.0];
    UIColor * Purple = [UIColor colorWithRed:0.7 green:0.1 blue:0.8 alpha:1.0];
    UIColor * Black = [UIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:1.0];

    if ([c isEqualToString:@"Red"])
        CGContextSetStrokeColorWithColor(*ctx, Red.CGColor);
    if ([c isEqualToString:@"Orange"])
        CGContextSetStrokeColorWithColor(*ctx, Orange.CGColor);
    if ([c isEqualToString:@"Yellow"])
        CGContextSetStrokeColorWithColor(*ctx, Yellow.CGColor);
    if ([c isEqualToString:@"Green"])
        CGContextSetStrokeColorWithColor(*ctx, Green.CGColor);
    if ([c isEqualToString:@"Blue"])
        CGContextSetStrokeColorWithColor(*ctx, Blue.CGColor);
    if ([c isEqualToString:@"Purple"])
        CGContextSetStrokeColorWithColor(*ctx, Purple.CGColor);
    if ([c isEqualToString:@"Black"])
        CGContextSetStrokeColorWithColor(*ctx, Black.CGColor);
    if ([c isEqualToString:@"White"])
        CGContextSetStrokeColorWithColor(*ctx, White.CGColor);
}

@end
```

Other considerations to add to code:

- Add error handling
- Save the bead data to disc or database
- Add scroll functionality to the necklace view

Thank you for viewing my Necklace Designer project!

MaryLou Washko