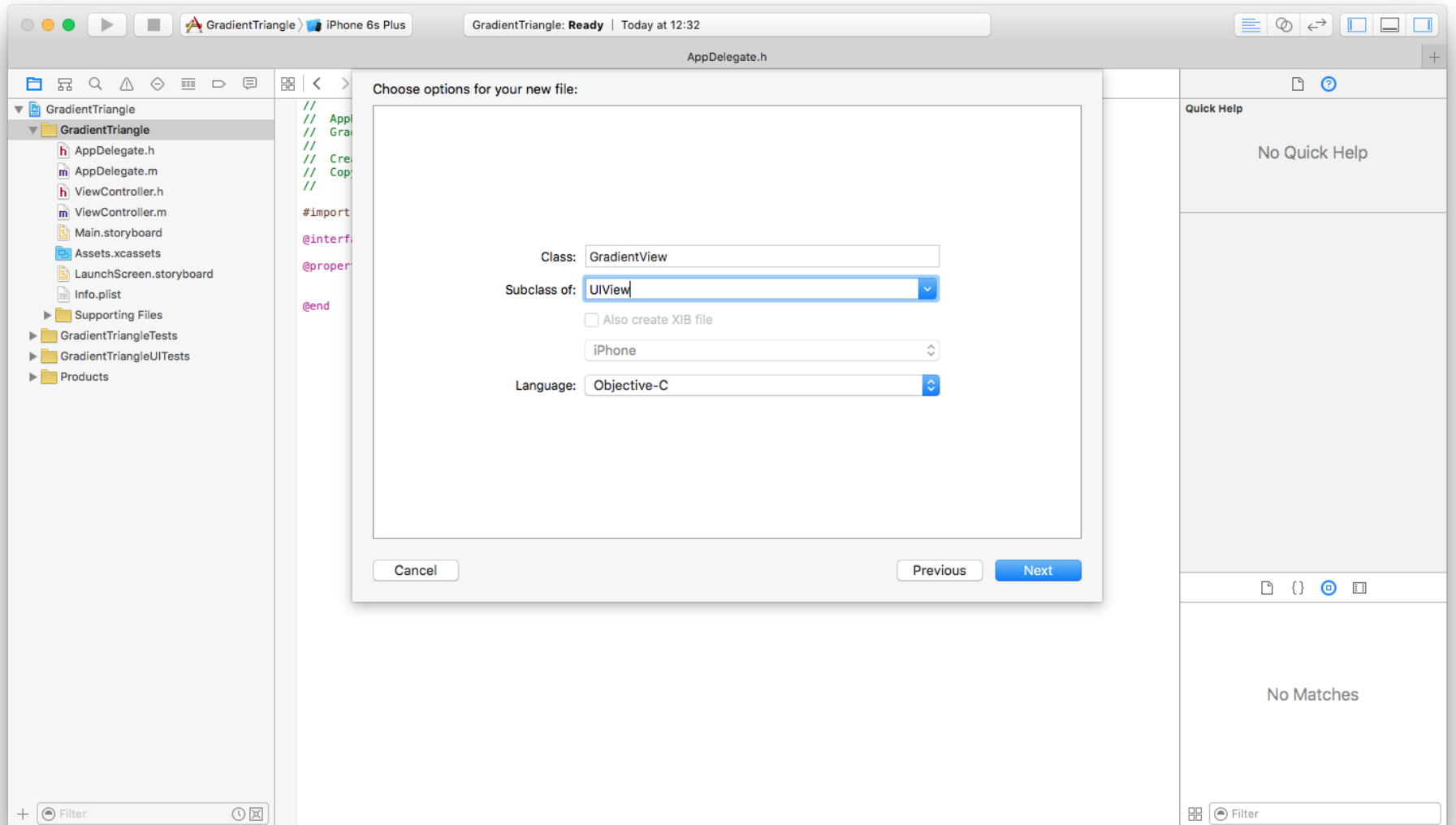


Praktikum iOS-Entwicklung

Aufgabe 5a

Susanne Böhmer



```
#import "GradientView.h"

@implementation GradientView

// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during animation.
- (void)drawRect:(CGRect)rect {

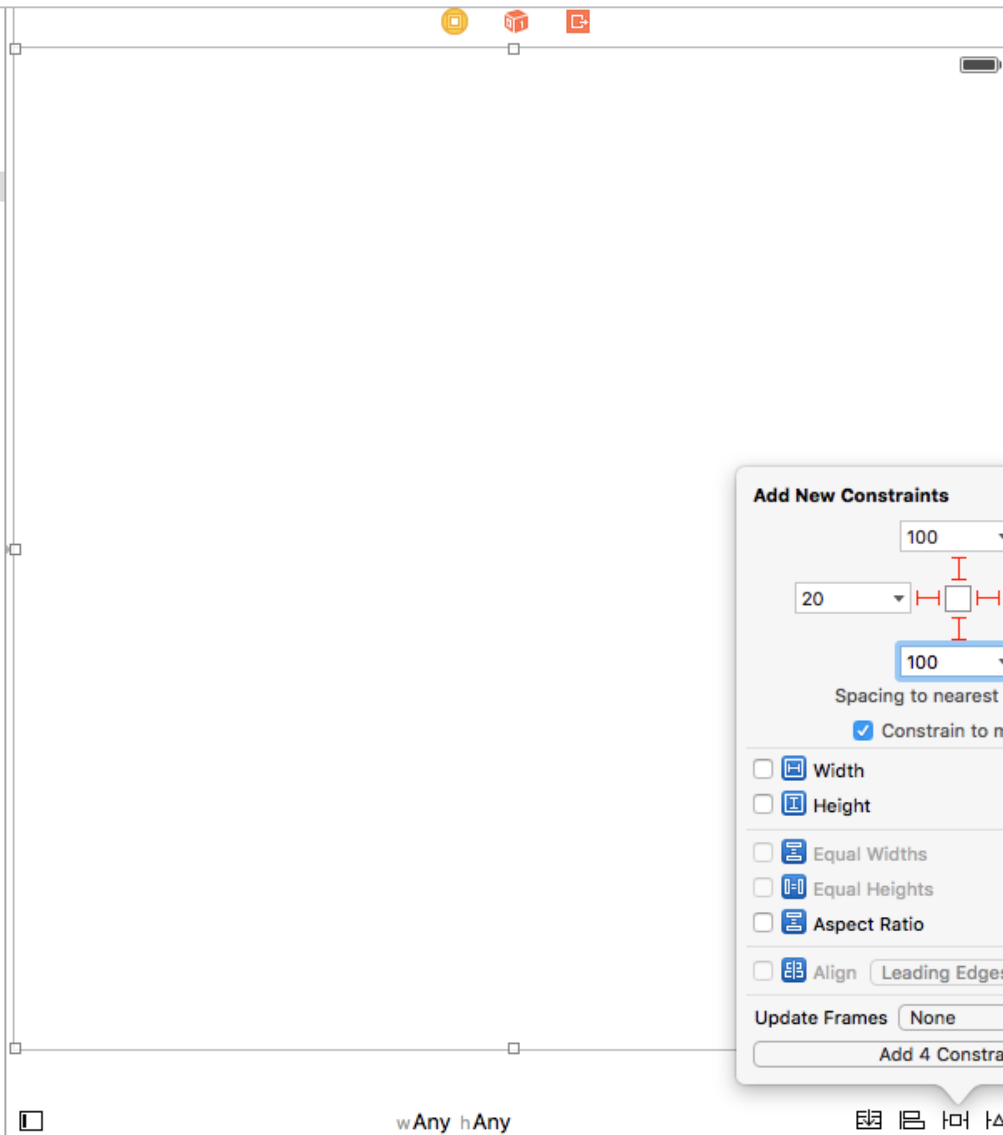
    // Höhe und Breite der View zur Laufzeit
    CGFloat width = self.frame.size.width;
    CGFloat height = self.frame.size.height;

    UIBezierPath *trianglePath = [UIBezierPath bezierPath];
    [trianglePath moveToPoint:CGPointMake(0, 0)];
    [trianglePath addLineToPoint:CGPointMake(width, 0)];
    [trianglePath addLineToPoint:CGPointMake(width/2, height)];
    [trianglePath closePath];

}

@end
```

- View Controller Scene
 - View Controller
 - Top Layout Guide
 - Bottom Layout Guide
 - View
 - Gradient View
 - First Responder
 - Exit
 - Storyboard Entry Point



Custom Class

Class + ▼

Module ▼

Identity

Restoration ID

User Defined Runtime Attributes

Key Path	Type	Value

Document

Code Specific Label

wl-LX-UX3

Inherited - (Nothing) ▼

to Font

{ }

ning (dragging) gestures whi...

Press Gesture Recognizer -

es a recognizer for long press

es which are invoked on the v...

Represents a rectangular

in which it draws and receives

Container View - Defines a region of a view controller that can include a

```

#import "GradientView.h"

@implementation GradientView

// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during animation.
- (void)drawRect:(CGRect)rect {

    // Höhe und Breite der View zur Laufzeit
    CGFloat width = self.frame.size.width;
    CGFloat height = self.frame.size.height;

    UIBezierPath *trianglePath = [UIBezierPath bezierPath];
    [trianglePath moveToPoint:CGPointMake(0, 0)];
    [trianglePath addLineToPoint:CGPointMake(width, 0)];
    [trianglePath addLineToPoint:CGPointMake(width/2, height)];
    [trianglePath closePath];

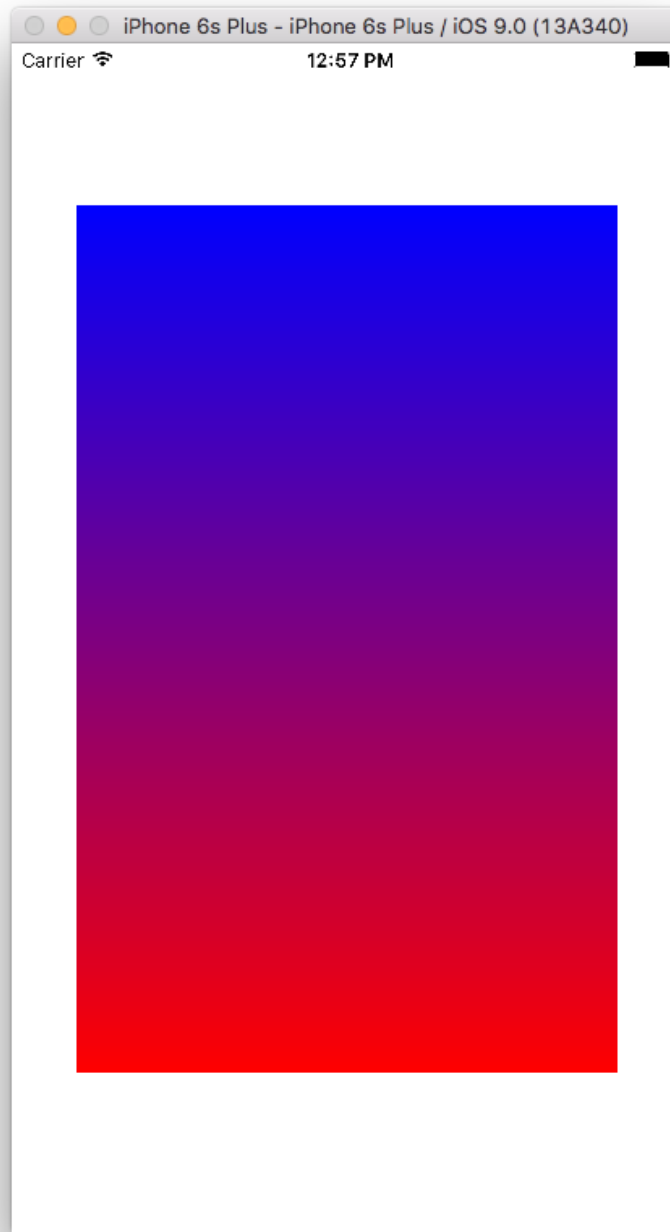
    // gradient erstellen
    CAGradientLayer *gradientLayer = [CAGradientLayer layer];
    gradientLayer.colors = [NSArray arrayWithObjects: (id)[[UIColor blueColor] CGColor], (id)[[UIColor redColor]CGColor], nil];

    gradientLayer.frame = self.bounds;

    // gradient der View hinzufügen
    [self.layer addSublayer:gradientLayer];
}

@end

```



```

#import "GradientView.h"

@implementation GradientView

// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during animation.
- (void)drawRect:(CGRect)rect {

    // Höhe und Breite der View zur Laufzeit
    CGFloat width = self.frame.size.width;
    CGFloat height = self.frame.size.height;

    UIBezierPath *trianglePath = [UIBezierPath bezierPath];
    [trianglePath moveToPoint:CGPointMake(0, 0)];
    [trianglePath addLineToPoint:CGPointMake(width, 0)];
    [trianglePath addLineToPoint:CGPointMake(width/2, height)];
    [trianglePath closePath];

    // gradient erstellen
    CAGradientLayer *gradientLayer = [CAGradientLayer layer];
    gradientLayer.colors = [NSArray arrayWithObjects: (id)[UIColor blueColor] CGColor, (id)[UIColor redColor] CGColor, nil];

    gradientLayer.frame = self.bounds;

    CAShapeLayer* shapeLayer = [[CAShapeLayer alloc] init];
    shapeLayer.path = trianglePath.CGPath;
    gradientLayer.mask = shapeLayer;

    // gradient der View hinzufügen
    [self.layer addSublayer:gradientLayer];
}

@end

```

