iPhone プログラミング UIKit 詳解リファレンス （第 1 版第 1 刷）

〔訂正情報〕

320～323 頁

「9.4.8　セルにコントロールを追加」のサンプルコード（320 頁冒頭の【interface】から 323 頁 9 行目の「@end」まで）に、誤りがありました。お詫び申し上げます。
正しいコードにつきましては、下記の通りとなります。

〔interface〕

```
#pragma mark ----- CellWithImageView -----

@interface CellWithImageView : UITableViewCell
{
 @private
  UIImageView* imageView_;
}

@property (nonatomic, retain) UIImageView* imageView;

@end

#pragma mark ----- CellWithSwitch -----

@interface CellWithSwitch : UITableViewCell
{
 @private
  UISwitch* theSwitch_;
}

@property (nonatomic, retain) UISwitch* theSwitch;

@end

#pragma mark ----- CellWithSlider -----

@protocol CellWithSliderDelegate
@required
- (void)slider:(UISlider*)slider valueDidChange:(float)value forRow:(NSUInteger)row;
@end
```

```
@interface CellWithSlider : UITableViewCell
{
 @private
  UISlider* slider_;
  NSUInteger row_;
  id delegate_;
}

@property (nonatomic, retain) UISlider* slider;
@property (nonatomic, assign) NSUInteger row;
@property (nonatomic, assign) id delegate;

@end

#pragma mark ----- SampleForCustomizedCell -----

@interface SampleForCustomizedCell : UITableViewController <CellWithSliderDelegate>
{
 @private
  NSArray* sections_;
  NSArray* dataSource_;
  NSMutableArray* sliderValues_;
}

@property (nonatomic, retain) NSArray* sections;
@property (nonatomic, retain) NSArray* dataSource;
@property (nonatomic, retain) NSMutableArray* sliderValues;

@end
```

[implementation]

```
#pragma mark ----- CellWithImageView -----

@implementation CellWithImageView

@synthesize imageView = imageView_;

- (void)dealloc {
  self.imageView = nil;
  [super dealloc];
}

- (id)initWithReuseIdentifier:(NSString*)identifier {
  if ( (self = [super initWithStyle:UITableViewCellStyleDefault reuseIdentifier:identifier]) ) {
    self.imageView = [[[UIImageView alloc] init] autorelease];
    self.imageView.autoresizingMask = UIViewAutoresizingFlexibleLeftMargin |
                                      UIViewAutoresizingFlexibleRightMargin |
                                      UIViewAutoresizingFlexibleTopMargin |
                                      UIViewAutoresizingFlexibleBottomMargin;
    [self addSubview:self.imageView];
  }
  return self;
}

- (void)layoutSubviews {
  [super layoutSubviews];
  [self.imageView sizeToFit];
  CGPoint newCenter = self.contentView.center;
  newCenter.x += 80;
  self.imageView.center = newCenter;
}

@end

#pragma mark ----- CellWithSwitch -----

@implementation CellWithSwitch

@synthesize theSwitch = theSwitch_;

- (void)dealloc {
  self.theSwitch = nil;
  [super dealloc];
}

- (id)initWithReuseIdentifier:(NSString*)identifier {
  if ( (self = [super initWithStyle:UITableViewCellStyleDefault reuseIdentifier:identifier]) ) {
    self.theSwitch = [[[UISwitch alloc] init] autorelease];
    self.theSwitch.on = YES;
    self.theSwitch.autoresizingMask = UIViewAutoresizingFlexibleLeftMargin |
                                      UIViewAutoresizingFlexibleRightMargin |
                                      UIViewAutoresizingFlexibleTopMargin |
                                      UIViewAutoresizingFlexibleBottomMargin;
    [self addSubview:self.theSwitch];
  }
  return self;
}

- (void)layoutSubviews {
  [super layoutSubviews];
  CGPoint newCenter = self.contentView.center;
  newCenter.x += 80;
  self.theSwitch.center = newCenter;
}

@end

#pragma mark ----- CellWithSlider -----

@implementation CellWithSlider

@synthesize slider = slider_;
@synthesize row = row_;
@synthesize delegate = delegate_;

- (void)dealloc {
  self.slider = nil;
  [super dealloc];
}

- (id)initWithReuseIdentifier:(NSString*)identifier {
  if ( (self = [super initWithStyle:UITableViewCellStyleDefault reuseIdentifier:identifier]) ) {
    self.slider = [[[UISlider alloc] init] autorelease];
    self.slider.frame = CGRectMake( 0, 0, 160, 20 );
    self.slider.autoresizingMask = UIViewAutoresizingFlexibleLeftMargin |
                                   UIViewAutoresizingFlexibleRightMargin |
                                   UIViewAutoresizingFlexibleTopMargin |
                                   UIViewAutoresizingFlexibleBottomMargin;
    [self.slider addTarget:self
                    action:@selector(sliderValueDidChange:)
          forControlEvents:UIControlEventValueChanged];
    [self addSubview:self.slider];
  }
  return self;
}

- (void)layoutSubviews {
  [super layoutSubviews];
  CGPoint newCenter = self.contentView.center;
  newCenter.x += 50;
  self.slider.center = newCenter;
}

- (void)sliderValueDidChange:(UISlider*)slider {
  [self.delegate slider:slider valueDidChange:slider.value forRow:self.row];
}

@end

#pragma mark ----- SampleForCustomizedCell -----

@implementation SampleForCustomizedCell

@synthesize sections = sections_;
@synthesize dataSource = dataSource_;
@synthesize sliderValues = sliderValues_;

- (void)dealloc {
  self.sections = nil;
  self.dataSource = nil;
  self.sliderValues = nil;
  [super dealloc];
}

- (id)init {
  if ( (self = [super initWithStyle:UITableViewStyleGrouped]) ) {
  }
  return self;
}

- (void)viewDidLoad {
  [super viewDidLoad];
  // 表示するデータを作成
  self.sections = [NSArray arrayWithObjects:@"名前", @"必殺技", @"強さ", nil];
  NSArray* rows1 = [[NSArray alloc] initWithObjects:@"宮本武蔵", nil];
  NSArray* rows2 = [[NSArray alloc] initWithObjects:@"二刀流", nil ];
  NSArray* rows3 = [[NSArray alloc] initWithObjects:
                                             @"攻撃力", @"防御力",
                                             @"etc1", @"etc2",
                                             @"etc3", @"etc4",
                                             @"etc5", @"etc6",
                                             @"etc7", @"etc8",
                                             @"etc9", @"etc10", nil];
  self.dataSource = [NSArray arrayWithObjects: rows1, rows2, rows3, nil];
  self.sliderValues = [NSMutableArray arrayWithObjects:
                          [NSNumber numberWithFloat:0.9], [NSNumber numberWithFloat:0.8],
                          [NSNumber numberWithFloat:0.1], [NSNumber numberWithFloat:0.2],
                          [NSNumber numberWithFloat:0.3], [NSNumber numberWithFloat:0.4],
                          [NSNumber numberWithFloat:0.5], [NSNumber numberWithFloat:0.6],
                          [NSNumber numberWithFloat:0.7], [NSNumber numberWithFloat:0.8],
                          [NSNumber numberWithFloat:0.9], [NSNumber numberWithFloat:1.0], nil];
  [rows1 release];
  [rows2 release];
  [rows3 release];
}

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
  return [self.sections count];
}

- (NSInteger)tableView:(UITableView*)tableView numberOfRowsInSection:(NSInteger)section {
  return [[self.dataSource objectAtIndex:section] count];
}

- (UITableViewCell*)tableView:(UITableView*)tableView
  cellForRowAtIndexPath:(NSIndexPath*)indexPath
{
  // カスタムセルの種類によって再利用のためのIDを変えること
  static const id identifiers[3] = { @"image-cell", @"switch-cell", @"slider-cell" };
  NSString* identifier = identifiers[ indexPath.section ];
  UITableViewCell* cell = [tableView dequeueReusableCellWithIdentifier:identifier];
  if ( nil == cell ) {
    switch ( indexPath.section ) {
      case 0:
        cell = [[[CellWithImageView alloc] initWithReuseIdentifier:identifier] autorelease];
        break;
      case 1:
        cell = [[[CellWithSwitch alloc] initWithReuseIdentifier:identifier] autorelease];
        break;
      case 2:
      default:
        cell = [[[CellWithSlider alloc] initWithReuseIdentifier:identifier] autorelease];
        [(CellWithSlider*)cell setDelegate:self];
        break;
    }
  }
  switch ( indexPath.section ) {
    case 0:
      [[cell imageView] setImage:[UIImage imageNamed:@"Samurai.png"]];
      break;
    case 2:
      {
        // UISlider付のセルなら値を設定
        NSNumber* value = [self.sliderValues objectAtIndex:indexPath.row];
        CellWithSlider* cellWithSlider = (CellWithSlider*)cell;
        cellWithSlider.slider.value = [value floatValue];
        cellWithSlider.row = indexPath.row;
      }
      break;
    default:
      break;
  }
  cell.textLabel.text =
    [[self.dataSource objectAtIndex:indexPath.section] objectAtIndex:indexPath.row];
  return cell;
}

// セルによって縦幅を変更する
- (CGFloat)tableView:(UITableView*)tableView heightForRowAtIndexPath:(NSIndexPath*)indexPath {
  if ( 0 == indexPath.section ) {
    return 100.0;
  } else {
    return 44.0;
  }
```

```
}

#pragma mark ----- Private Methods -----

- (void)slider:(UISlider*)slider valueDidChange:(float)value forRow:(NSUInteger)row {
  // 変更された値を保存
  [self.sliderValues replaceObjectAtIndex:row
                               withObject:[NSNumber numberWithFloat:value]];
}

@end
```