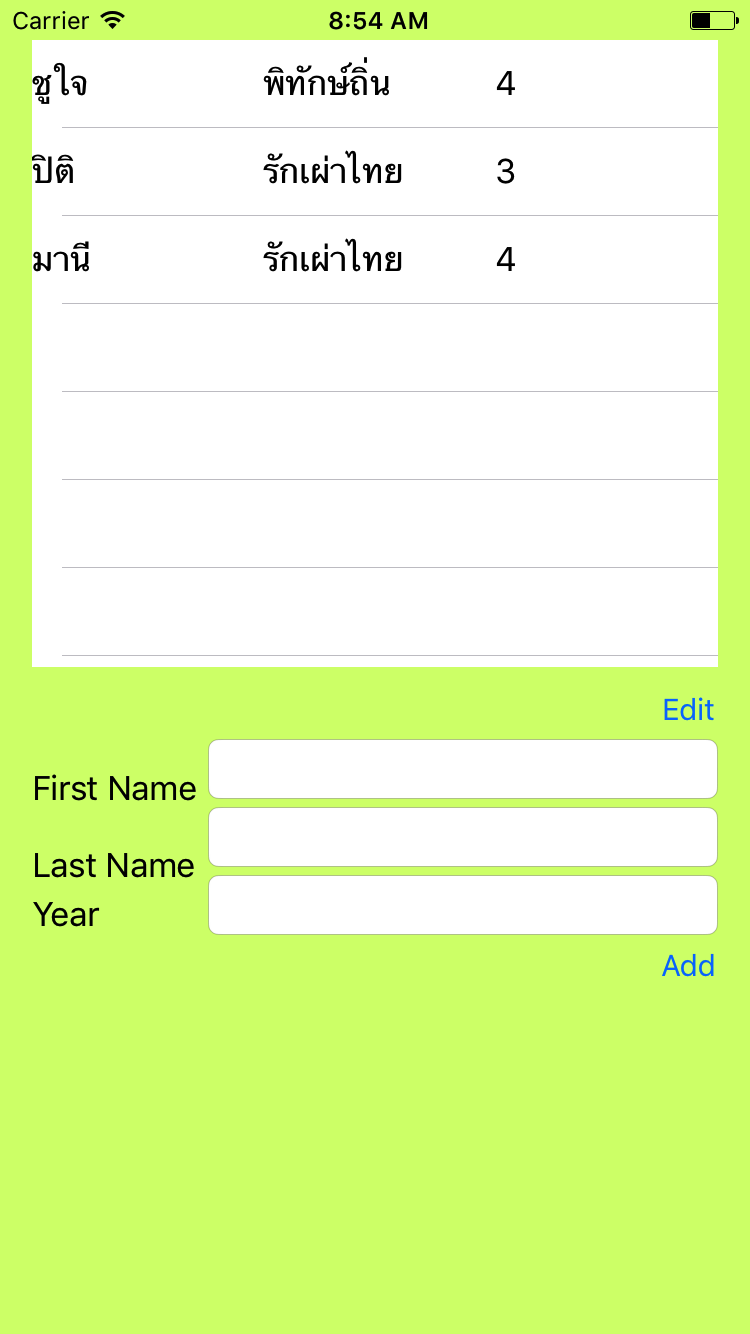
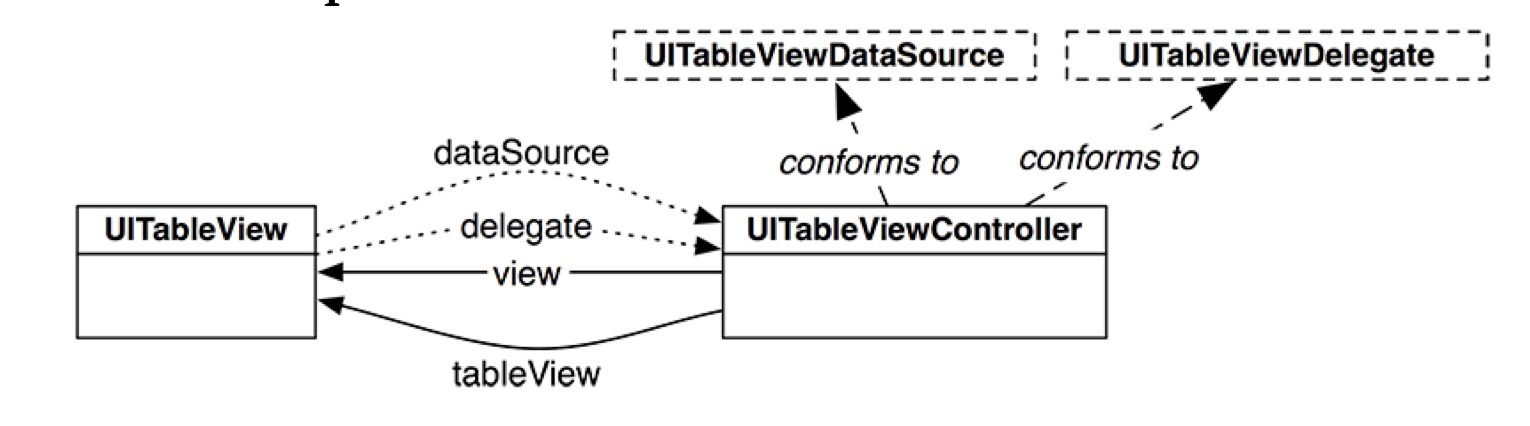
# Table View Exercise

Application that we are going to make in this exercise.



UITableView Class Diagram 

1. สร้าง Project StudentTableView based on Single View Application
2. สร้าง Student class ที่เป็น Subclass ของ NSObject เพื่อเก็บ ชื่อ, นามสกุล, ชั้นปี

สร้าง Constructor (init function) ที่เหมาะสม และ convenience init เพื่อใช้สร้าง Object ด้วย data สุ่มๆ เพื่อทดสอบ (เฉลย ท้ายเอกสาร)

https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift\_Programming\_Language/Initialization.html

Question: convenience init ต่างกับ init ธรรมดาอย่างไร

ตัวอย่าง:

class MyPet:NSObject{

    var name:String

    init(name:String) {

        self.name=name

    }

    convenience init(random: Bool = false) {

        if random {

            let n:[String] = ["Pii", "Mike", "Momo"]

            var idx = arc4random\_uniform( UInt32(n.count))

            let name = n[Int(idx)]

            self.init(name: name)

        }else{

            self.init(name: "-")

        }

    }

}

1. สร้าง StudentData class ที่เป็น Subclass ของ NSObject เพื่อเก็บ Array of Student. แล้วสร้าง method

add(student:Student)

add(firstName:String, lastName:String, year:Int)

addAt(index:Int,student:Student)

getAt(index:Int) -> Student

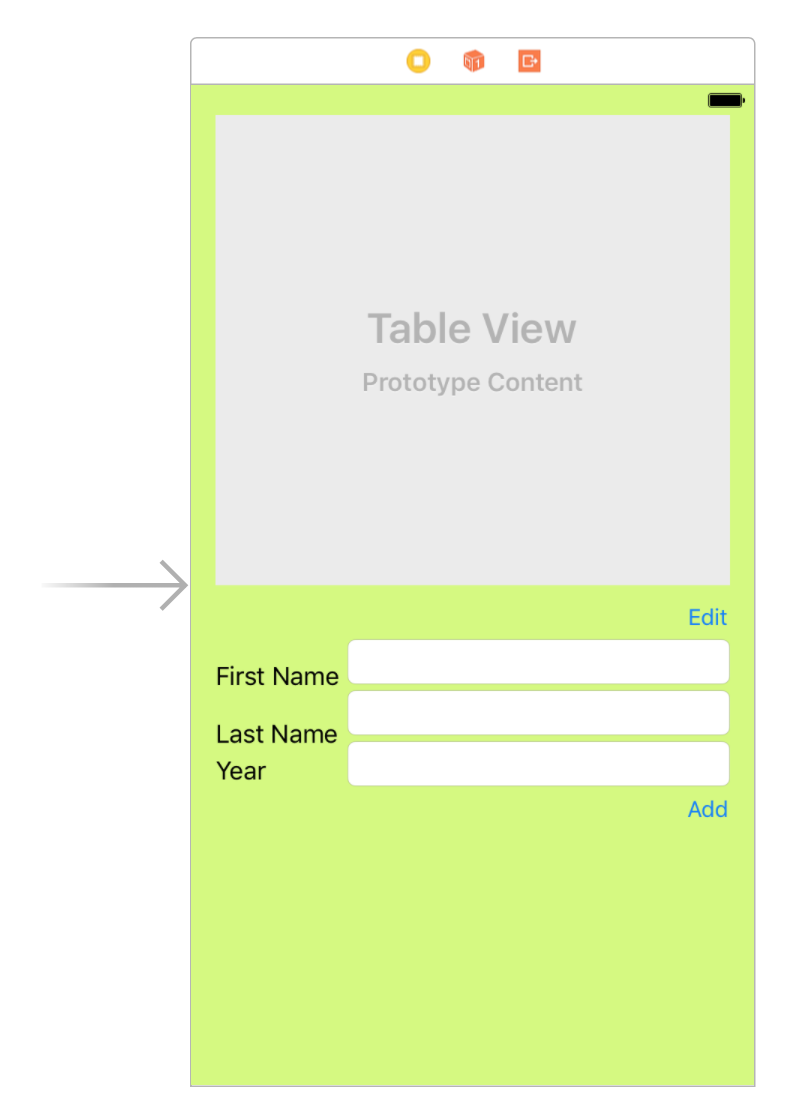
deleteAt(index:Int) -> Student

total()->Int

(เฉลย ท้ายเอกสาร)

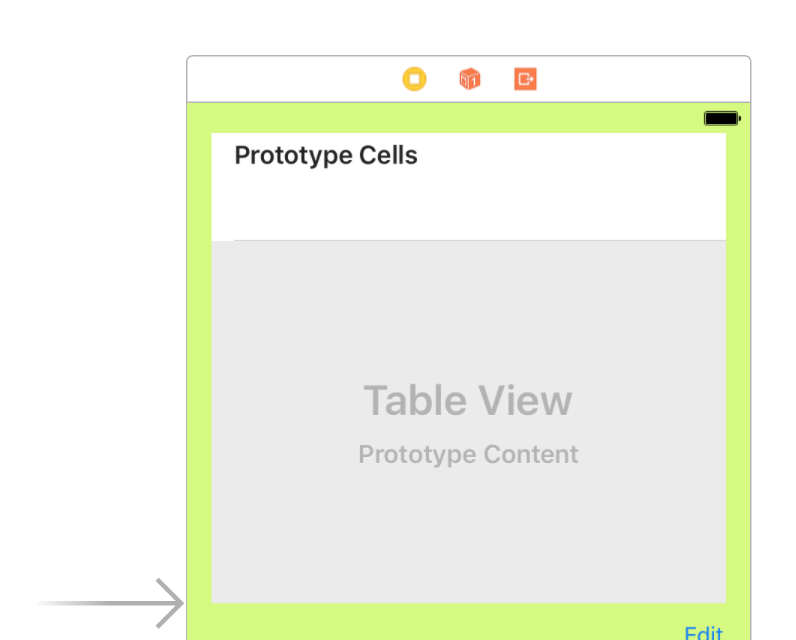
Question: Why this is better than just use Array? ( ทำไมสร้าง Object ห่อ Array ดีกว่า Array เปล่าๆ)

1. สร้าง StudentData class ที่เป็น Subclass ของ NSObject เพื่อเก็บ Array of Student. แล้วสร้าง method
2. Create a page as shown. (If you don't want trouble later, make sure that the green part is a view on top of main view with all constrains. You will need it for moving view from under a keyboard later.)

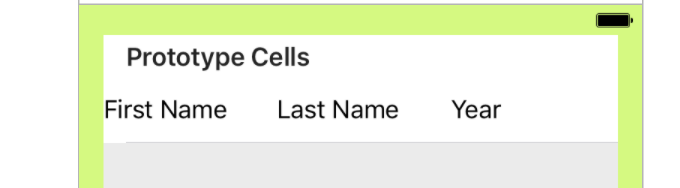


1. Create all button event handlers and outlets for all text fields and Table View.
2. Put one Table View Cell in Table View.

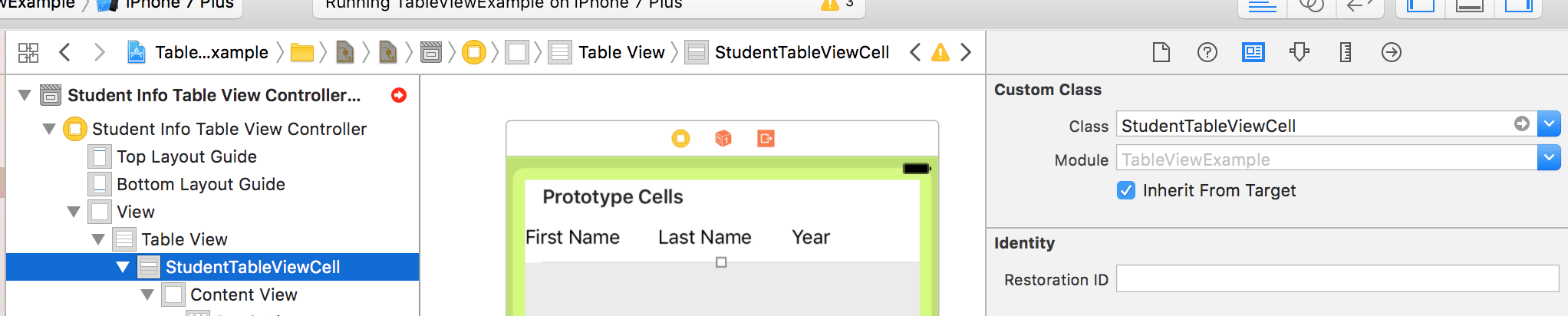
Table View Cell = ต้นแบบของแถวในตาราง ถ้าอยาก มีหลายแบบก็ให้ใส่เพิ่มตามจำนวนรูปแบบ ไม่ใช่ใส่ตามจำนวนแถวที่อยากได้



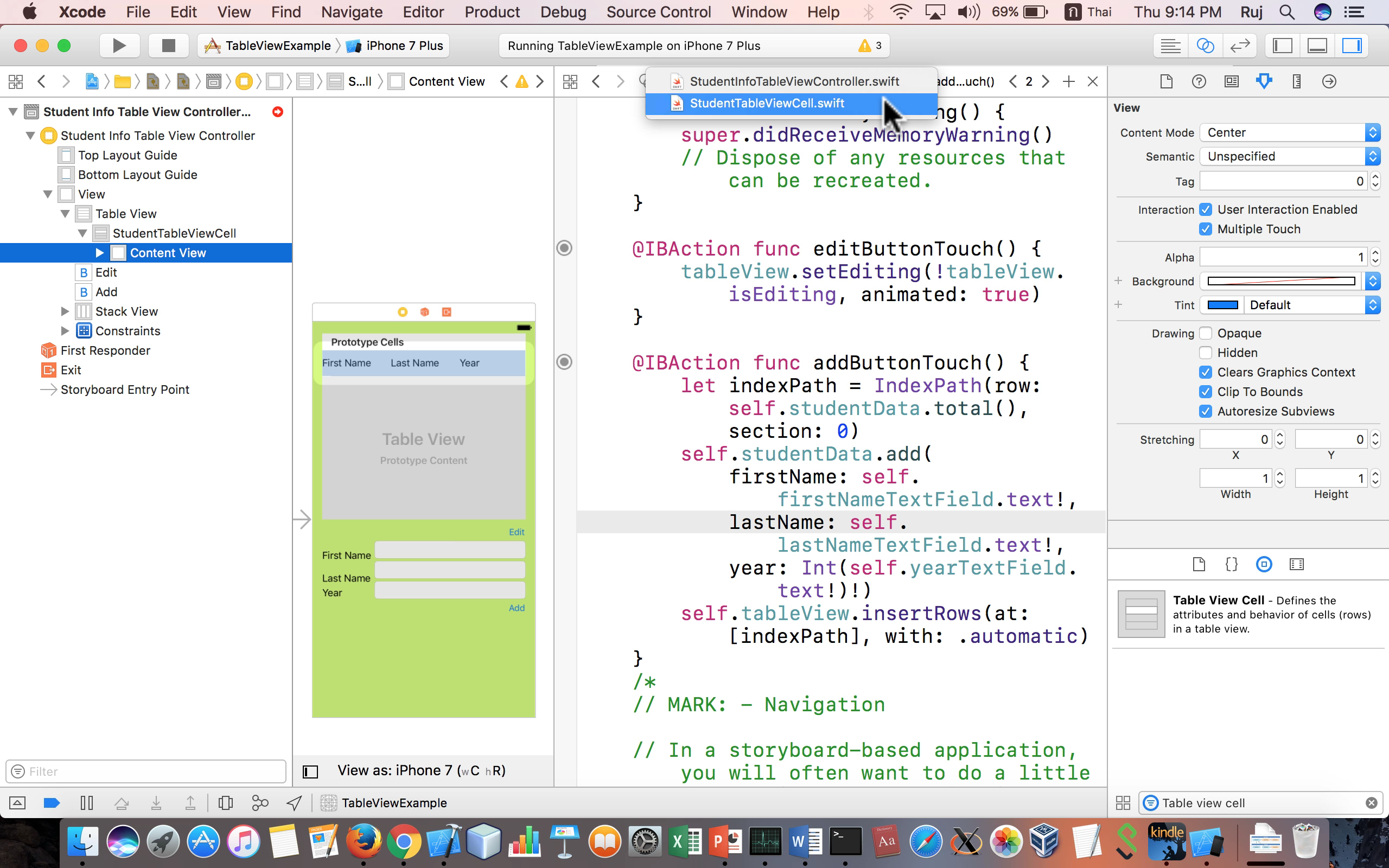
1. Put 3 Labels in Table View Cell with appropriate constrains.

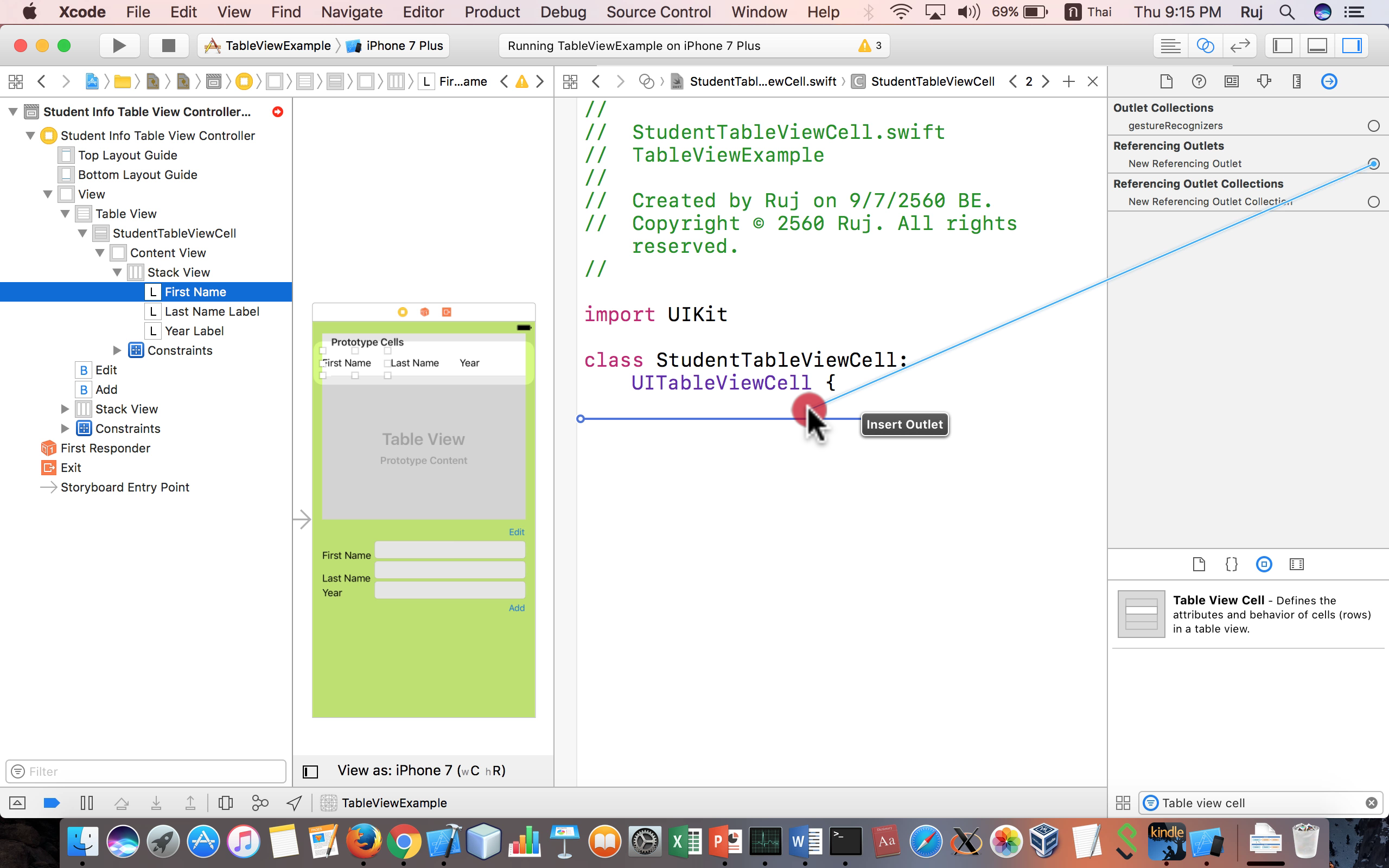


1. Create StudentTableViewCell ที่เป็น subclass ของ UITableViewCell
2. เปลี่ยน Class ของ Table View Cell เป็น StudentTableViewCell



1. สร้าง Outlet ของ Label ทั้งสามอันใน StudentTableViewCell

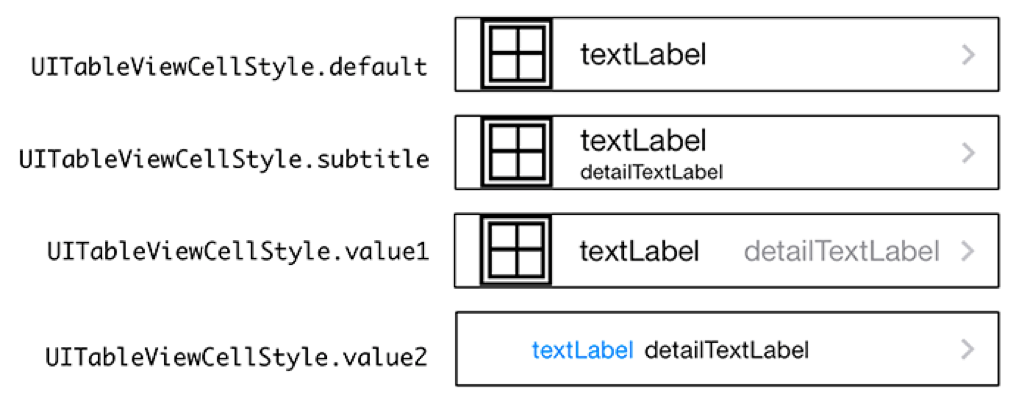
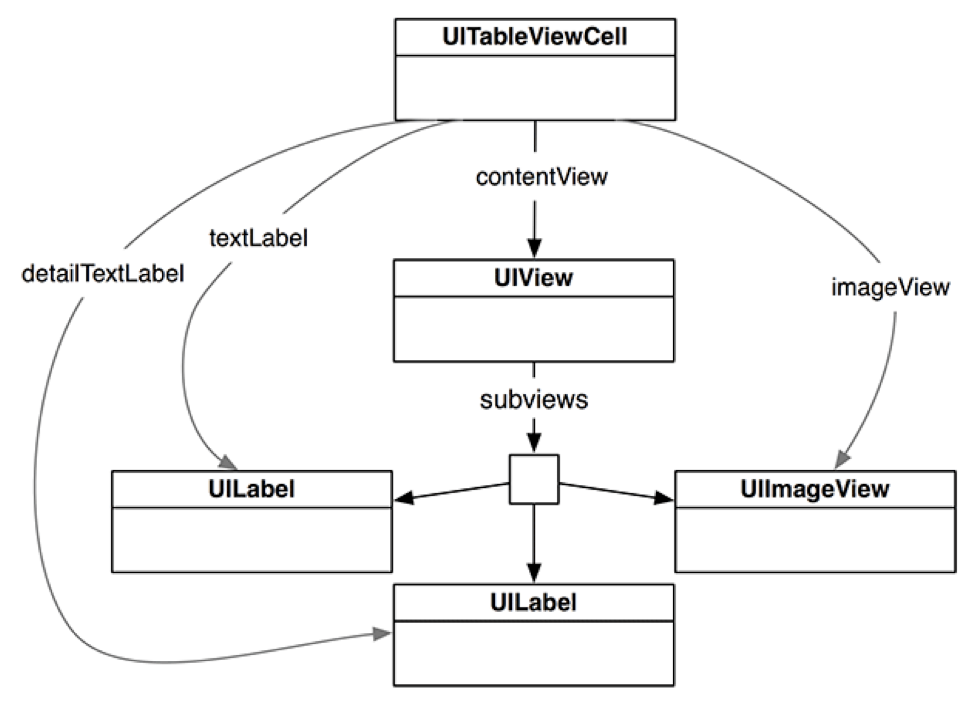




1. สร้าง Outlet ของ Label ทั้งสามอันใน StudentTableViewCell
2. สร้าง methods ใน StudentTableViewCell เพื่อใช้ตั้งค่าของ Label โดยง่าย

(เฉลย ท้ายเอกสาร)

โดยปกติถ้าไม่อยากออกแบบสามารถใช้ Table View Cell แบบสำเร็จรูปได้ มีให้เลือก 4 แบบ: Basic, Left Detail, Right Detail, Subtitle และ มี Fields ดังแสดงในรูป



1. สร้าง Field ใน View Controller เพื่อเก็บ StudentData (ส่วนนี้คือ Model ใน MVC) แล้ว เรียก self.ชื่อที่ตั้ง.add(student: Student(random: true)) ใน viewDidLoad() สักหลายๆครั้งตามชอบใจเพื่อใส่ข้อมูลสุ่มลงใน StudentData
2. Extends your View Controller with UITableViewDelegate and UITableViewDataSource using extension. ใช้ extension จะทำให้ Code อ่านได้ง่ายกว่าใช้วิธีอื่น ใส่ extension

(เฉลย ท้ายเอกสาร)

ตัวอย่าง (View Controller ในตัวอย่าง ชื่อ StudentInfoTableViewController )

class StudentInfoTableViewController: UIViewController {

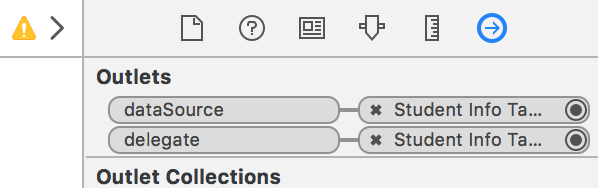
...

}

extension StudentInfoTableViewController:UITableViewDelegate{

}

1. Link dataSource and delegate ของ UITableView ไปที่ ViewController ของเรา



ในตัวอย่าง StudentInfoTableViewController เป็น ViewController class

1. UITableViewDataSource ต้อง implements อย่างน้อย 3 methods นี้ (ใช้ Autofill อย่าพยายามพิมพ์เอง จะผิดได้ง่าย) งานของเรามี Section เดียว

func numberOfSections(in tableView: UITableView) -> Int

func tableView(\_ tableView: UITableView,

numberOfRowsInSectio section: Int) -> Int

func tableView(\_ tableView: UITableView,

cellForRowAt indexPath: IndexPath) -> UITableViewCell {

let cell = tableView.dequeueReusableCell(

withIdentifier: "StudentTableViewCell", for: indexPath)

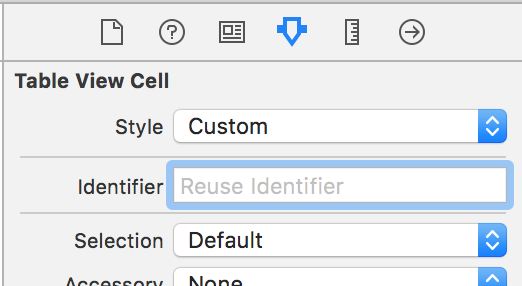
as! StudentTableViewCell

        // ใส่ code สำหรับตั้งค่า StudentTableViewCell ตรงนี้

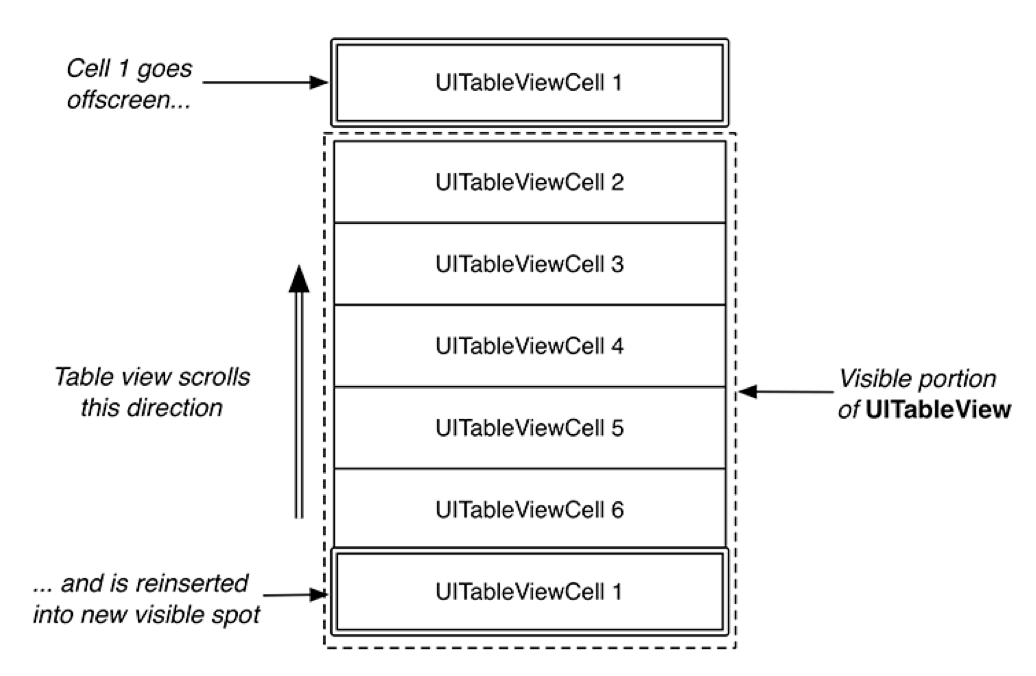
        return cell

}

Function ล่างสุด ( func tableView(\_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ) มีหน้าที่ recycle และสร้าง Table View Cell ตามต้องการโดยใช้คำสั่ง tableView.dequeueReusableCell เพื่อสร้าง Table View Cell ตามชนิดที่ระบุโดย withIdentifier. String ที่ระบุจะเป็นอะไรก็ได้แต่โดยทั่วไปจะใช้ชื่อ class จะได้เข้าใจง่าย ชื่อนี้จะต้องตรงกับ reuse identifier ที่ระบุใน Storyboard (ดังแสดงในรูป).



การทำงานของ recycle ของ Table View Cell เป็นดังในรูป Cell ที่หลุดหน้าจอจะถูกเอาไป recycle มาแสดงอีกด้านหนึ่งของ Table



เราสามารถมี Template ของ Table View Cell หลายแบบได้แต่ต้องตั้ง reuse identifier ให้ต่างกัน และเรียกมาใช้ ตามค่าที่ใส่ใน withIdentifier

1. ลอง Run ดู โดยยังไม่ตั้ง reuse identifier ของ Table View Cell ใน Storyboard

Question: ทำไม Error

1. ตั้ง reuse identifier ของ Table View Cell ใน Storyboard ให้ตรงกับ withIdentifier ใน code แล้ว run ใหม่
2. เติมเต็ม event listener ของ ปุ่ม Add ให้เพิ่ม Row ใน Table

(เฉลย ท้ายเอกสาร)

การเติม Row ทำได้โดยใช้ method ต่อไปนี้ของ UITabelView

insertRows(at: [IndexPath], with: UITableViewRowAnimation)

Note: ต้องใส่ข้อมูลใน Model (ในที่นี้คือ StudentData) ก่อนเรียก method insertRows

การทดลอง: ให้ลอง เรียก method insertRows แล้วค่อยใส่ข้อมูลใน StudentData เปรียบเทียบกับ ใส่ข้อมูลใน StudentData แล้วค่อยเรียก method insertRows

1. เติมเต็ม event listener ของ ปุ่ม Edit ให้เปลี่ยนไปกลับ mode ของ TableView ระหว่าง edit mode กับ ธรรมดา (ให้เปลี่ยน Text ของ Button ระหว่าง Edit <-> Done ได้ด้วยตาม Mode)

การเปลี่ยน Mode ใช้ method setEditing(\_ editing: Bool, animated: Bool) ของ TableView

(เฉลย ท้ายเอกสาร)

Mode Editing จะทำให้สามารถ ลบและย้าย Row ได้

Note: ปุ่มใช้ method setTitle เพื่อเปลี่ยน คำ https://developer.apple.com/documentation/uikit/uicontrolstate

1. ลอง Run แล้วสังเกตุว่า ลบได้หรือไม่
2. การลบจำเป็นต้อง Implement method ของ UITableViewDataSource

tableView(\_ tableView: UITableView,

commit editingStyle: UITableViewCellEditingStyle,

forRowAt indexPath: IndexPath)

โดย argument editingStyle จะเป็นตัวบอกว่าเป็น event ไหน

https://developer.apple.com/documentation/uikit/uitableviewcelleditingstyle

จง implement method สำหรับ delete

Question: ควรใช้ method ของ UITableView ชื่ออะไรในการลบ และควรลบจาก StudentData ก่อนหรือจาก TableView ก่อน

(เฉลย ท้ายเอกสาร)

1. เราสามารถย้าย Row ได้ใน edit mode (Feature จะไม่แสดงใน UI จนกว่าจะ implement method) โดย implement

tableView(\_ tableView: UITableView,

moveRowAt sourceIndexPath: IndexPath,

to destinationIndexPath: IndexPath)

จง implement method สำหรับย้าย Row

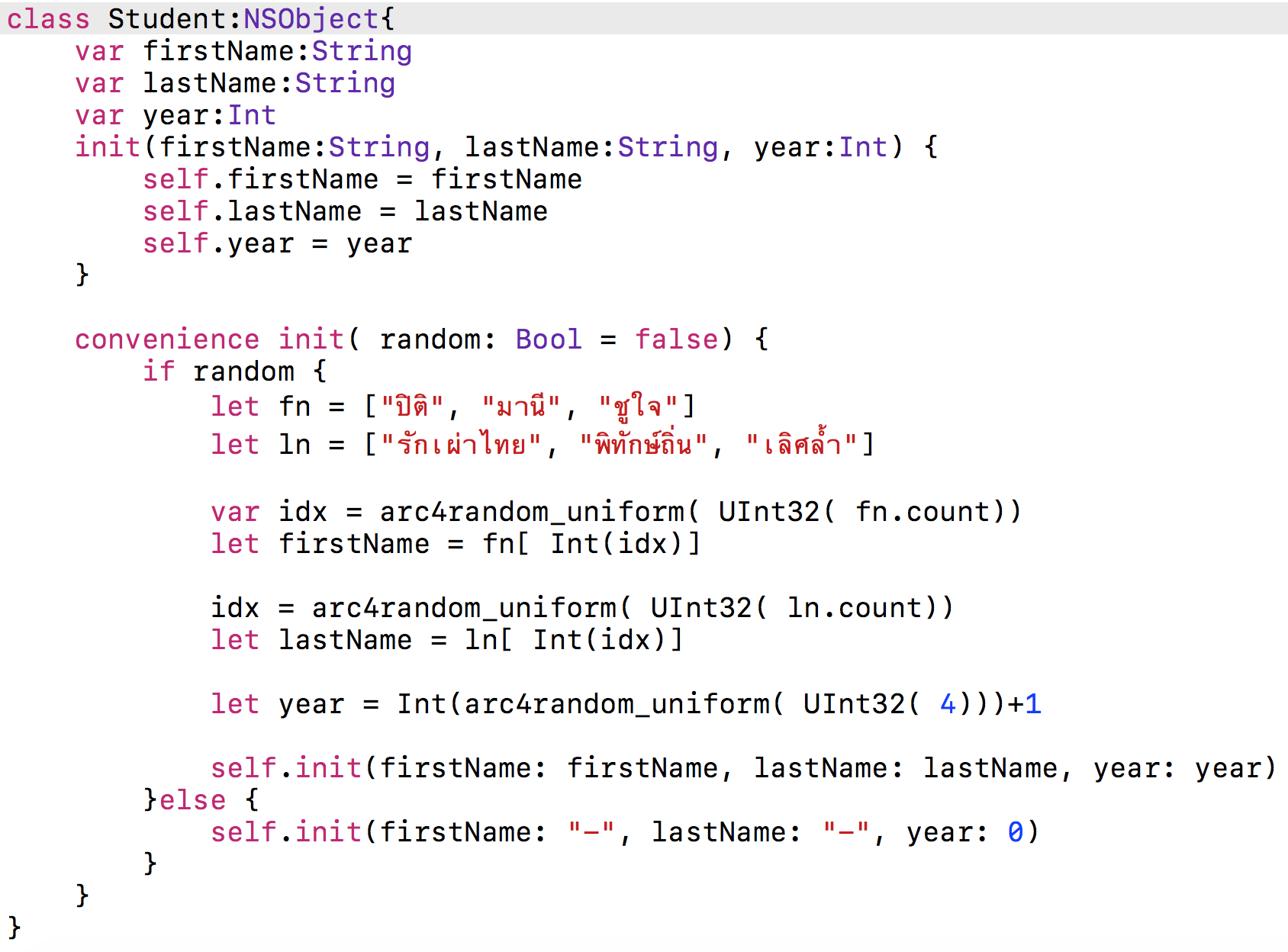
(เฉลย ท้ายเอกสาร)

Homework

1. Implement: ให้ลบ Text Field หลังกด add.
2. Make keyboard not blocking text fields and comple keyboard dismissing
3. เพิ่ม Section ที่สอง ให้ Table เพื่อบันทึก ราคาอาหารในโรงอาหาร โดยที่ Row ใน section ที่สอง เป็นคนละแบบ กับ Section เก่า ให้ใส่ UI เสริมตามสมควร

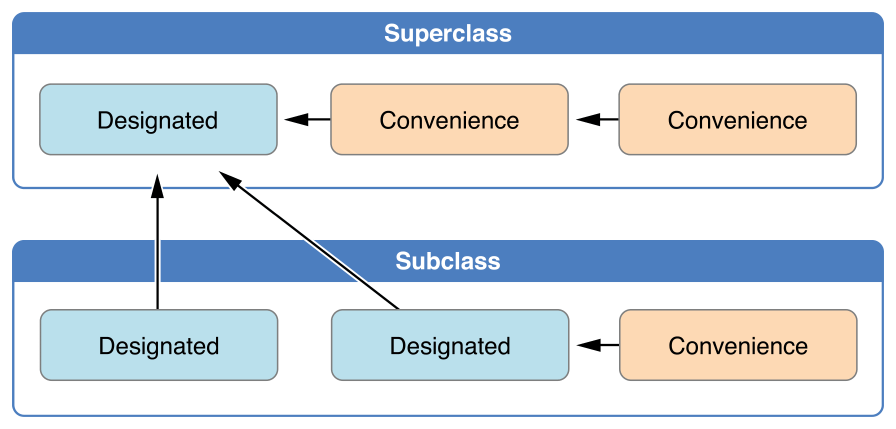
Answer

:



Swift subclasses do not inherit their superclass initializers by default.

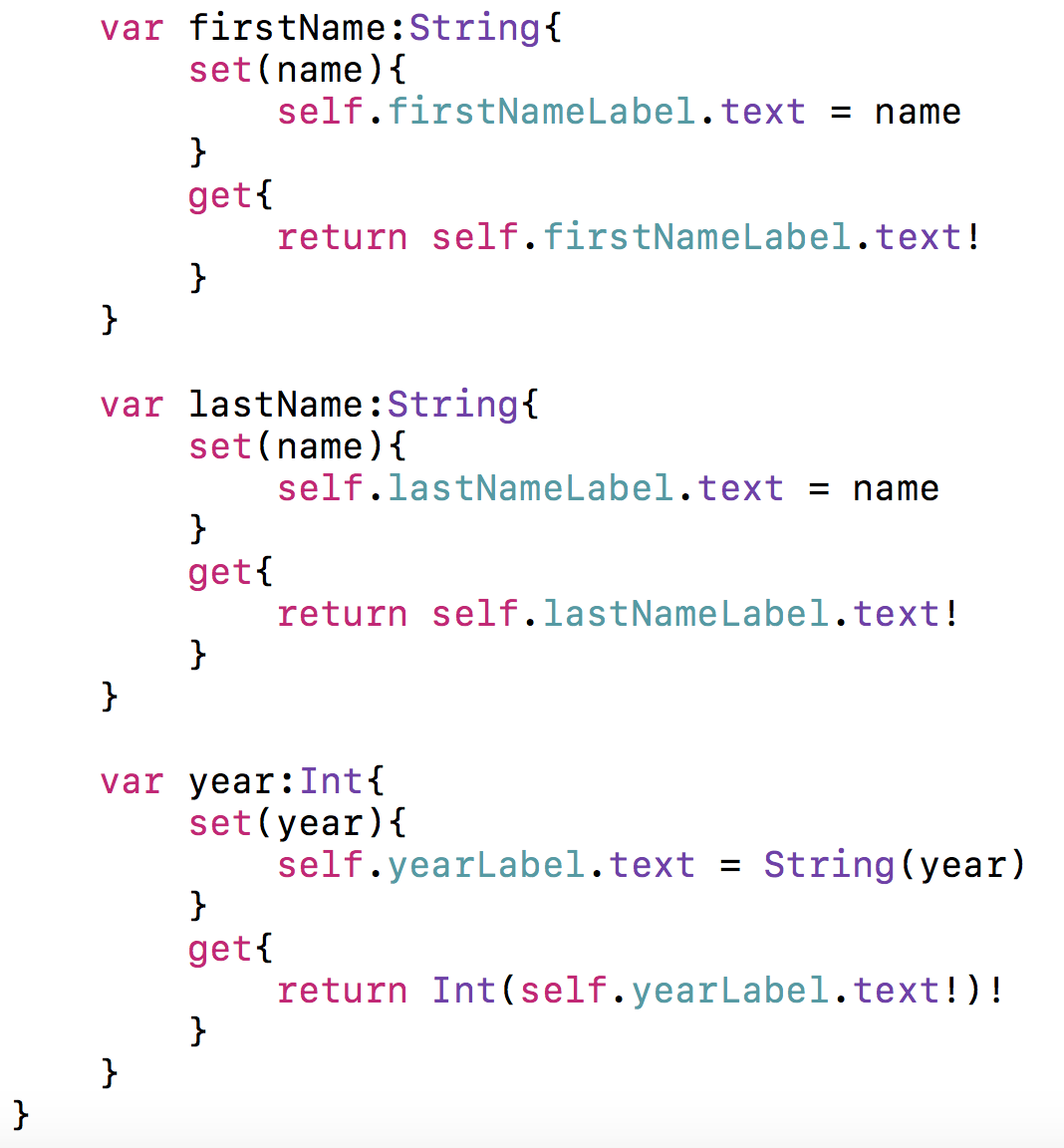
Convenience initializers must call another initializer on the same type, whereas designated initializers must call a designated initializer on its superclass.



Task 2:

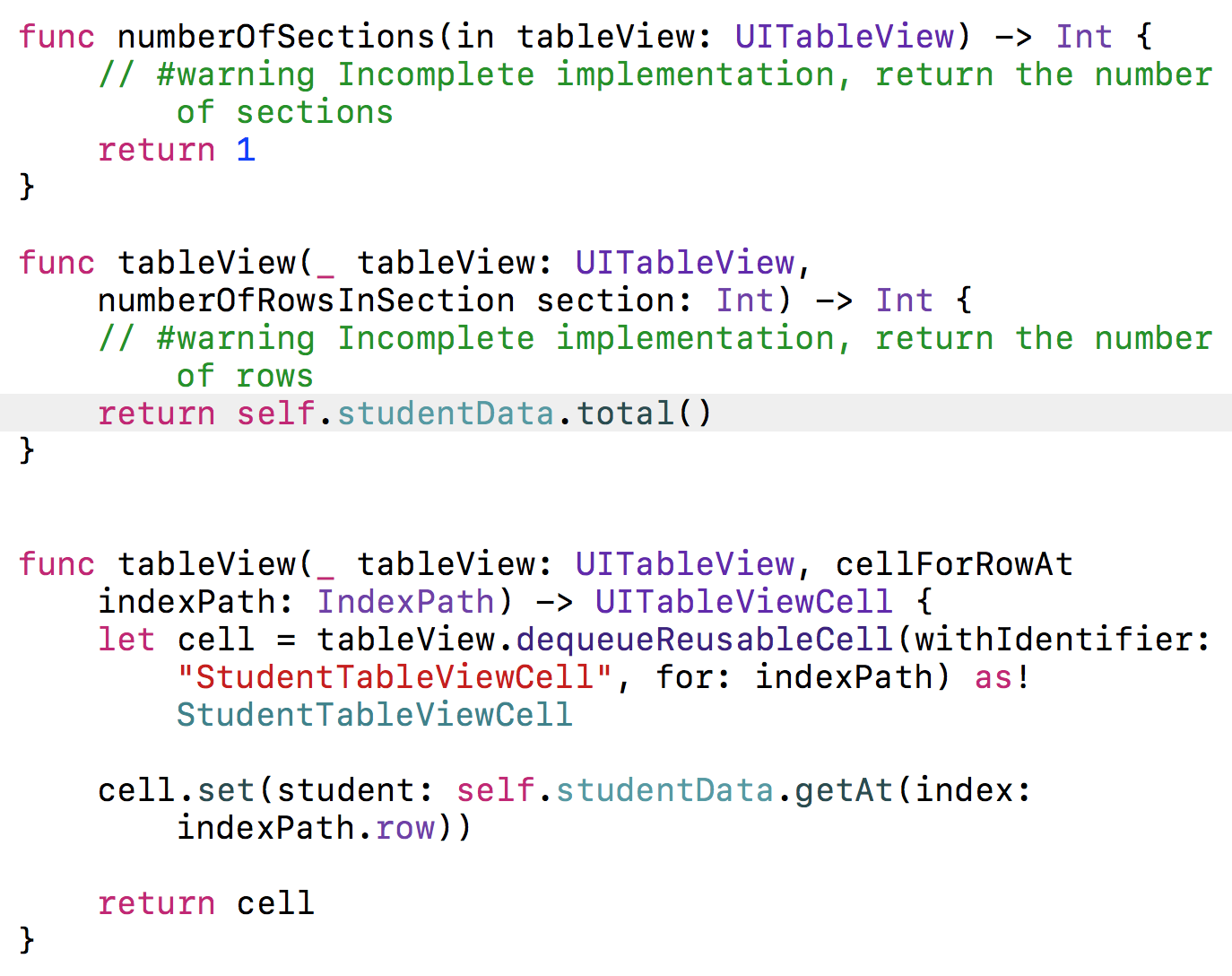


Task 13

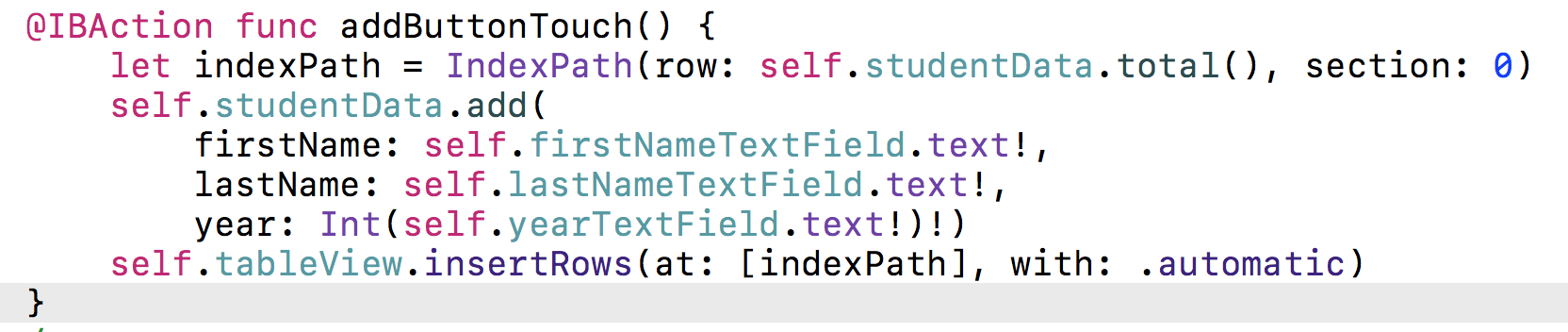


Task 15

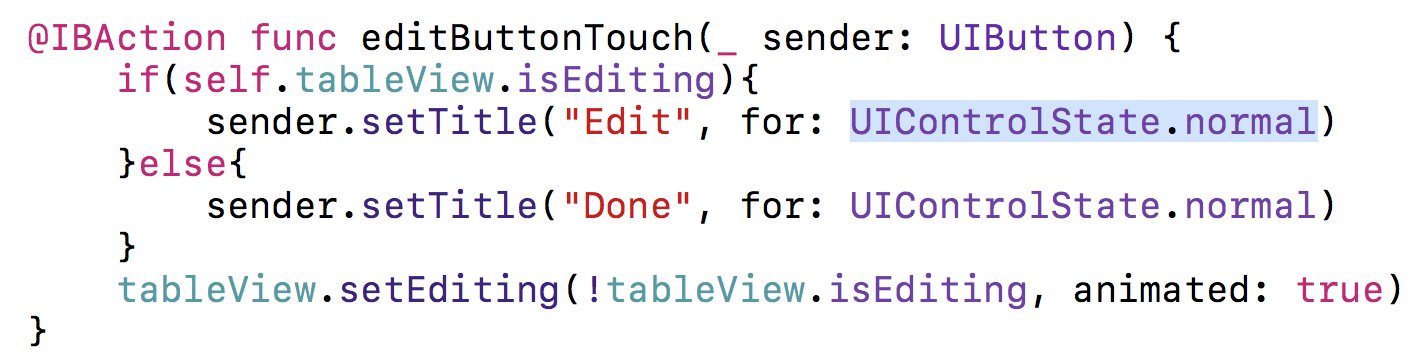
Note: studentData เป็น field ที่เก็บ StudentData



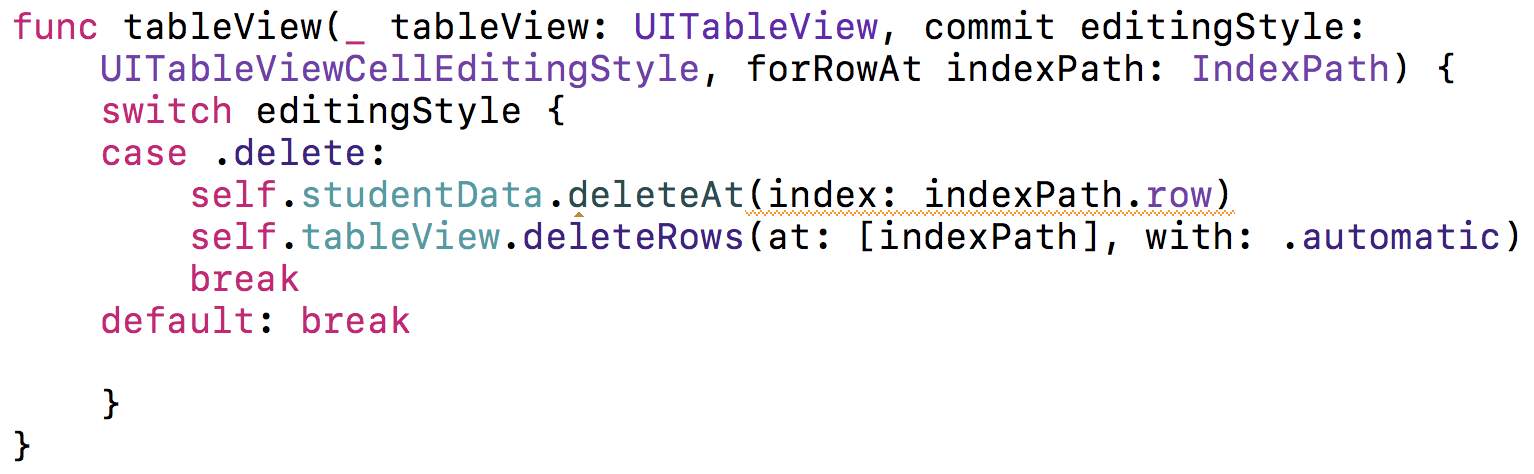
Task 19



Task 20



Task 22



Task 23

