

## 第二轮考核任务

### 1. Swift基础语法学习

- 本轮学习完**Swift指南**全部内容
- 另外，学习时并不用把全部都复制下来当作学习笔记。你觉得需要打的就跟着打一遍。

### 2. 本轮题目

- 涉及知识较多，题目要求较长，题意不理解可以提问

- 需要实现以下 类、枚举、协议...

- 一个枚举(**enum**) VehicleError // 继承自Error
- 一个基类Vehicle
- 一个类 ElectricCar // 表示电动汽车 继承自Vehicle
- 一个协议(**protocol**) Rechargeable // 表示可充电的， ElectricCar要遵循此协议
- 一个协议 ElectricCarDelegate // ElectricCar的**委托(delegate)**所要遵循的协议
- 一个类SuperCharger // 表示充电桩

- **具体要求如下**

#### VehicleError

- 继承自**Error**

```
//有以下类型
case outOfPower //表示没电
case other //该类型需使用关联值(关联值类型为String)，关联值用来表示一些其他错误
```

- 有一个**计算属性** var description: String
- 根据**错误类型**给description一个值
- 如outOfPower，则在getter里return “你的车没电啦”
- 若是other，则return关联值

#### Vehicle

- 拥有属性:

```
let name: String //名称
let seat: Int //座位数
var mileage: Float = 0 //公里数(从出厂开始经过的路程)
```

- 用init初始化name 和 seat
- 有一个方法:

```
// 本次行驶miles公里
func drive(miles: Float) {
    self.mileage += miles
}
```

#### ElectricCar

- 继承自**Vehicle**

- 拥有属性:

```
var delegate: ElectricCarDelegate? //委托

var batteryPercent: Float { //表示剩余电量百分比 范围0-100
    //这里要求用属性观察器,实现每次修改这个值后自动输出 "当前剩余电量: \(batteryPercent)"
}

let percentPerKilometer: Float = 0.15 //常量 每公里耗0.15%的电
override var mileage: Float { //重写 mileage
    //这里要用属性观察器 实现计算差值(即本次行驶的里程数) distance
    //然后计算耗电量 (如果耗电量大于剩余电量,耗电量则为剩余电量)
    //输出 "本次行驶耗电:\(usageOfPercent)"
    //然后电池扣掉相应的电量
}
```

- 有一个抛出函数(throws)

```
func launch
//若没电 则抛出没电的错误
//否则输出 print("Your car has been launched")
```

- 用init初始化
- 初始电量batteryPercent为60
- 注意什么时候调用super.init

#### Rechargeable

- 该协议有一个方法

```
func recharge(percent: Float) //表示电量增加percent,范围0-100
```

#### ElectricCarDelegate

- ElectricCar的**委托(delegate)**所要遵循的协议
- 该协议有一个方法

```
func electricCar(_ car: ElectricCar,willCharge percent: Float)
```

- **\*\*扩展(extension)\*\***ElectricCar 遵循协议 Rechargeable
- 当调用**recharge**方法时，不要直接在该方法里修改**batteryPercent**，而是通知**delegate**的 **electricCar(\_ : willCharge:)**，将具体的操作交给**delegate**来操作

- 类SuperCharger

- 拥有属性

```
let identifier: String //表示充电桩名字
```

- 该类要遵循协议 **ElectricCarDelegate**

```
//函数体如下
{
    print("Welcome to SuperCharger \(identifier)")
    //计算最终电量 finalPercent (若超过100,则最终为100)
    //输出 "当前电量为: \(当前电量),最终将会充到: \(finalPercent)"
    //然后修改车的电量
}
```

- **创建实例以及执行代码**

- 创建ElectricCar实例 modelx

- name为 “Model X” seat为 6

- 创建SuperCharger实例 superCharger

- identifier 为 “福州市名城中心站”

- 将superCharger设为modelx的委托(delegate)

- 接着执行以下代码

```
modelx.launch() // 这里自己修改
modelx.recharge(percent: 76)
modelx.drive(miles: 300)
modelx.drive(miles: 600)
modelx.launch() //这里自己修改
modelx.recharge(percent: 96)
modelx.drive(miles: 200)

//最后 体会一下关联值
let error = //VehicleError的other类型 关联值为 "121212"
print(error.description)
```

- 上面代码中 modelx.launch() 的修改
- 要用到do-catch
- 虽然你抛出的错误类型属于**VehicleError**，但catch里error只知道是**Error**类型，并没有description 这个属性，**VehicleError**类型才有。
- 所以需要类型转换 as? 结合可选项展开 **\*\*if let 变量名 = \*\*** 将error转换为VehicleError类型
- 然后catch里输出 print(变量名.description)

- 没出错的话输出如下：

```
Your car has been launched
Welcome to SuperCharger 福州市名城中心站
当前电量为: 60.0,最终将会充到: 100.0
当前剩余电量: 100.0
本次行驶耗电:45.0
当前剩余电量: 55.0
本次行驶耗电:55.0
当前剩余电量: 0.0
你的车没电啦
Welcome to SuperCharger 福州市名城中心站
当前电量为: 0.0,最终将会充到: 96.0
当前剩余电量: 96.0
本次行驶耗电:30.000002
当前剩余电量: 66.0
121212
```

### 本轮需提交内容

- 学习语法过程中所使用的**playground**文件
- 题目解答写在另一个**playground**文件

文件命名为 年级姓名\_iOS第x轮考核 (如19何凯\_iOS第2轮考核)

提交邮箱 563589126@qq.com

- 在学习语法过程中遇到不太理解的，统一在旁边的注释里打上 **MARK:** 再写上问题。在你们提交 playground文件后，方便我们进行定位解答

## 第二次考核任务截止时间: 2021年11月21日 24:00