

JSON Model 生成器

1. 编程要求

根据给定的JSON文件，生成对应的Swift 模型（对象的类型）。例如：

给定GroceryProduct.json,

```
{
    "name": "Durian",
    "points": 600,
    "description": "A fruit with a distinctive scent."
}
```

生成的模型文件为： GroceryProduct.swift,

```
struct GroceryProduct: Codable {
    var name: String
    var points: Int
    var description: String?
}
```

(1) Model生成器至少要支持对象和数组，数组中的值可以为字符串、数值、null和对象。对象最多支持一级嵌套，数组不必支持嵌套，见第2部分的订单产品。

(2) 作业在课上检查。支持更多特性的作业有加分，如更多层次的嵌套、更多值类型的支持等。

(3) 项目类型为macOS的命令行工具。

(4) 在github建立个人代码库，积累🌟，对工作和考研有很大帮助。

(5) 输入输出文件及参数请自行设计。JSON文件的扩展名不一定是.json。只要文件符合JSON语法，就是JSON文件。根据需要自行指定。

2. JSON介绍

JSON (JavaScript Object Notation) 是一种轻量级的数据交换格式。轻量级是相对XML文档而言的，同样的文档，JSON描述的项目字符少。由于移动端设计的要求是流量尽可能少、传输速度尽可能快，轻量级的JSON就成了理想的数据交换格式。

例如，订单产品的JSON描述可以设计为：

```
{
  "orderId": 104,
  "totalPrice": 103.45,
  "products": [
    {
      "id": 123,
      "name": "Product #1",
      "price": 12.95
    },
    {
      "id": 137,
      "name": "Product #2",
      "price": 82.95
    }
  ]
}
```

JSON文档由两种结构构成：对象和数组。对象是一个“名称-值”对的无序集合，以“{”开始，以 “}”结束，名称与值之间用一个“:”分隔，“名称-值”对之间用“,”分隔。数组是值的有序集合，以“[”开始，以 “]”结束，值之间用“,”分隔。在数组中，值可以是字符串、数值、true/false、null、对象或者是数组，而且这些结构可以嵌套。

3. JSON编码和解码

Swift 对JSON文档提供支持，包括JSONEncoder、JSONDecoder、示例等，具体可参考Xcode的Developer Documentation。JSON编码示例如下：

```
struct GroceryProduct: Codable {
    var name: String
    var points: Int
    var description: String?
}

let pear = GroceryProduct(name: "Pear", points: 250,
description: "A ripe pear.")

let encoder = JSONEncoder()
encoder.outputFormatting = .prettyPrinted

let data = try encoder.encode(pear)
print(String(data: data, encoding: .utf8)!)

/* Prints:
{
  "name" : "Pear",
  "points" : 250,
  "description" : "A ripe pear."
}
*/
```

JSON解码示例如下：

```
let json = """
{
  "name": "Durian",
  "points": 600,
  "description": "A fruit with a distinctive scent."
}
""".data(using: .utf8)!

let decoder = JSONDecoder()
let product = try decoder.decode(GroceryProduct.self, from: json)
```

```
print(product.name) // Prints "Durian"
```

JSON解码过程中，需要用到对象的类型，也称为Model。在上例中，解码用到的类型是**struct** GroceryProduct

4. 材料支持

(1) 参考项目

<https://github.com/jsonmodel/jsonmodel>

(2) 文件访问

参考FileManager、Bundle、FileHandle等类。部分示例代码如下。

```
let str = "Hello, World!"

let data = str.data(using: .utf8)

let manager = FileManager.default //Bundle.main.bundlePath

let baseUrl = URL(fileURLWithPath: manager.currentDirectoryPath)

let filename = "text.txt"
let file = baseUrl.appendingPathComponent(filename)

let success = manager.createFile(atPath: file.path, contents:
data, attributes: nil)

print(success)
```