Meal.swift

// FoodTracker

//

// Created by Jane Appleseed on 11/10/16.

// Copyright © 2016 Apple Inc. All rights reserved.

//

import UIKit

import os.log

class Meal: NSObject, NSCoding {

//MARK: Properties

var name: String

var photo: UIImage?

var rating: Int

//MARK: Archiving Paths

static let DocumentsDirectory = FileManager().urls(for: .documentDirectory, in: .userDomainMask).first!

static let ArchiveURL = DocumentsDirectory.appendingPathComponent("meals")

//MARK: Types

struct PropertyKey {

static let name = "name"

static let photo = "photo"

static let rating = "rating"

}

//MARK: Initialization

init?(name: String, photo: UIImage?, rating: Int) {

// The name must not be empty

guard !name.isEmpty else {

return nil

}

// The rating must be between 0 and 5 inclusively

guard (rating >= 0) && (rating <= 5) else {

return nil

}

// Initialization should fail if there is no name or if the rating is negative.

if name.isEmpty || rating < 0 {

return nil

}

// Initialize stored properties.

self.name = name

self.photo = photo

self.rating = rating

}

//MARK: NSCoding

func encode(with aCoder: NSCoder) {

aCoder.encode(name, forKey: PropertyKey.name)

aCoder.encode(photo, forKey: PropertyKey.photo)

aCoder.encode(rating, forKey: PropertyKey.rating)

}

required convenience init?(coder aDecoder: NSCoder) {

// The name is required. If we cannot decode a name string, the initializer should fail.

guard let name = aDecoder.decodeObject(forKey: PropertyKey.name) as? String else {

os\_log("Unable to decode the name for a Meal object.", log: OSLog.default, type: .debug)

return nil

}

// Because photo is an optional property of Meal, just use conditional cast.

let photo = aDecoder.decodeObject(forKey: PropertyKey.photo) as? UIImage

let rating = aDecoder.decodeInteger(forKey: PropertyKey.rating)

// Must call designated initializer.

self.init(name: name, photo: photo, rating: rating)

}

}