Mobile Application Development
Aileen Pierce

OBJECT-ORIENTED PROGRAMMING

Object Oriented Programming

- Object-Oriented Programming models the natural way humans think about things in terms of attributes and behavior.
- Object-Oriented Programming (OOP) makes building complex, modular and reusable applications much easier.
- OOP combines data and operations on the data (behavior) into one unit, a class.

Class

- A class provides a template, or blueprint for its objects.
- A class defines the characteristics (data properties) and behavior (methods) of its objects.
  - The data properties provide the class attributes, or characteristics.
  - Similar to functions, methods are the actions or behaviors of the class

Objects

- An object is an instance, or occurrence, of a given class.
  - An object of a given class has the structure and behavior defined by the class
  - Many different objects can be defined for a given class
  - All objects of the same class have the same structure

Objects

- Properties store constant or variable data as part of an object.
- Methods are functions that are associated with a specific class.
  - Instance methods are functions that operate on an object.
  - Type methods are functions that operate on the class itself.

Initialization

- Initialization is the process of preparing an object of a class for use.
- Initializers/Constructors are special methods that create a new object of a class.

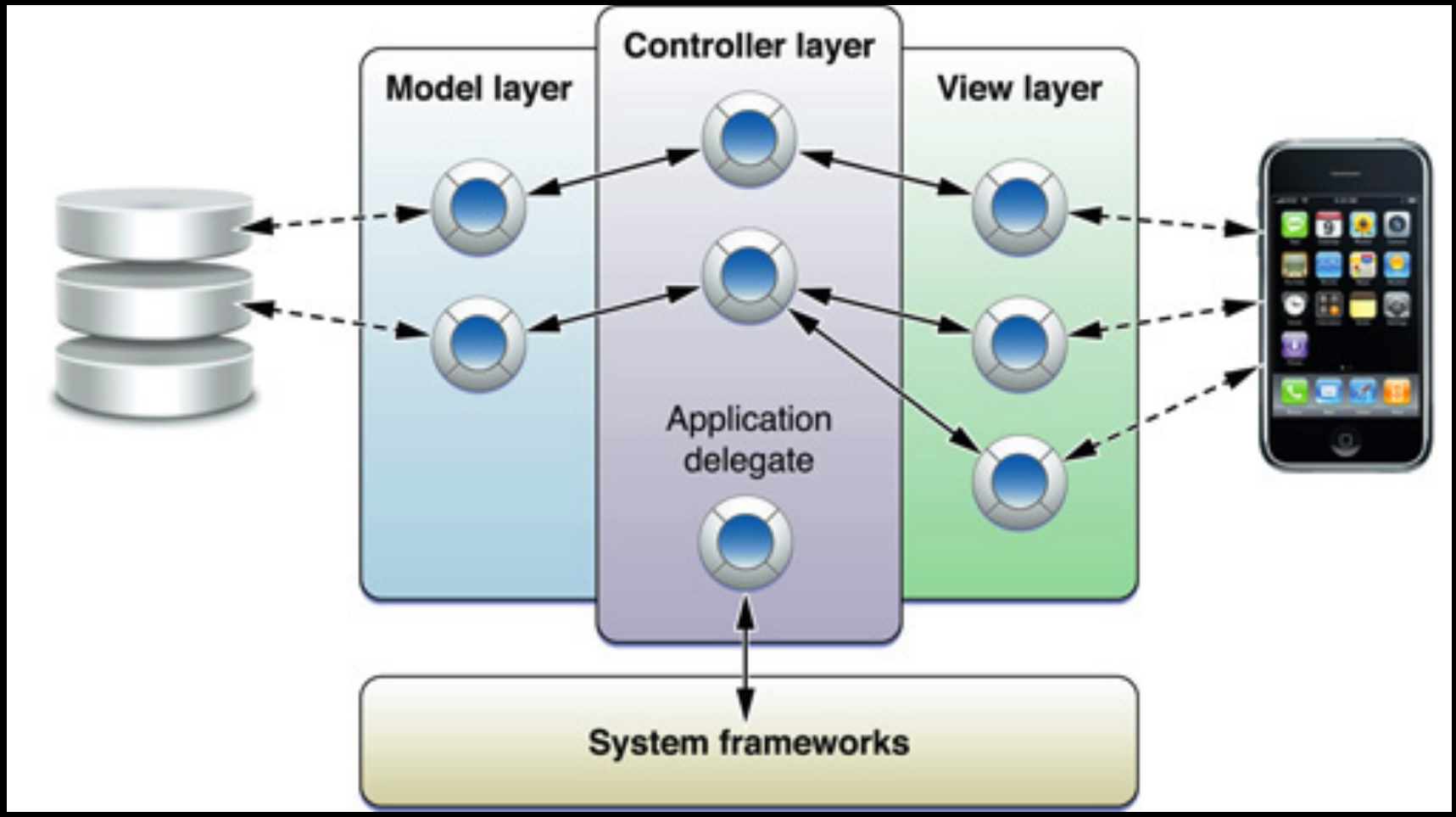
Encapsulation

- Modular
  - data properties and behavior are packaged into a single well-defined programming unit
- Information Hiding
  - Methods provide the public interface to objects of a class
  - Implementation is kept private

Inheritance

- Inheritance enables classes to form a hierarchy like a family tree.
- Allows subclasses to share the structure and behavior of its superclass.
  - Superclass is the parent class
  - A subclass extends a class
    - Inherits from the superclass
    - Can add properties and methods
    - Can override a method with a new one
- Inheritance allows you to easily reuse code

Model-View-Controller(MVC)



Model-View-Controller(MVC)

- MVC is the foundation for iOS app code
- Model: holds the data and classes
  - Should be UI independent
- View: all items for the user interface
- Controller: links the model and the view together. The backbone or brain of the app.
- These categories should never overlap.
- The goal of MVC is to have any object be in only one of these categories.