混合应用开发

--- Angular组件

河北师范大学软件学院
Software College of Hebei Normal University

内容提纲

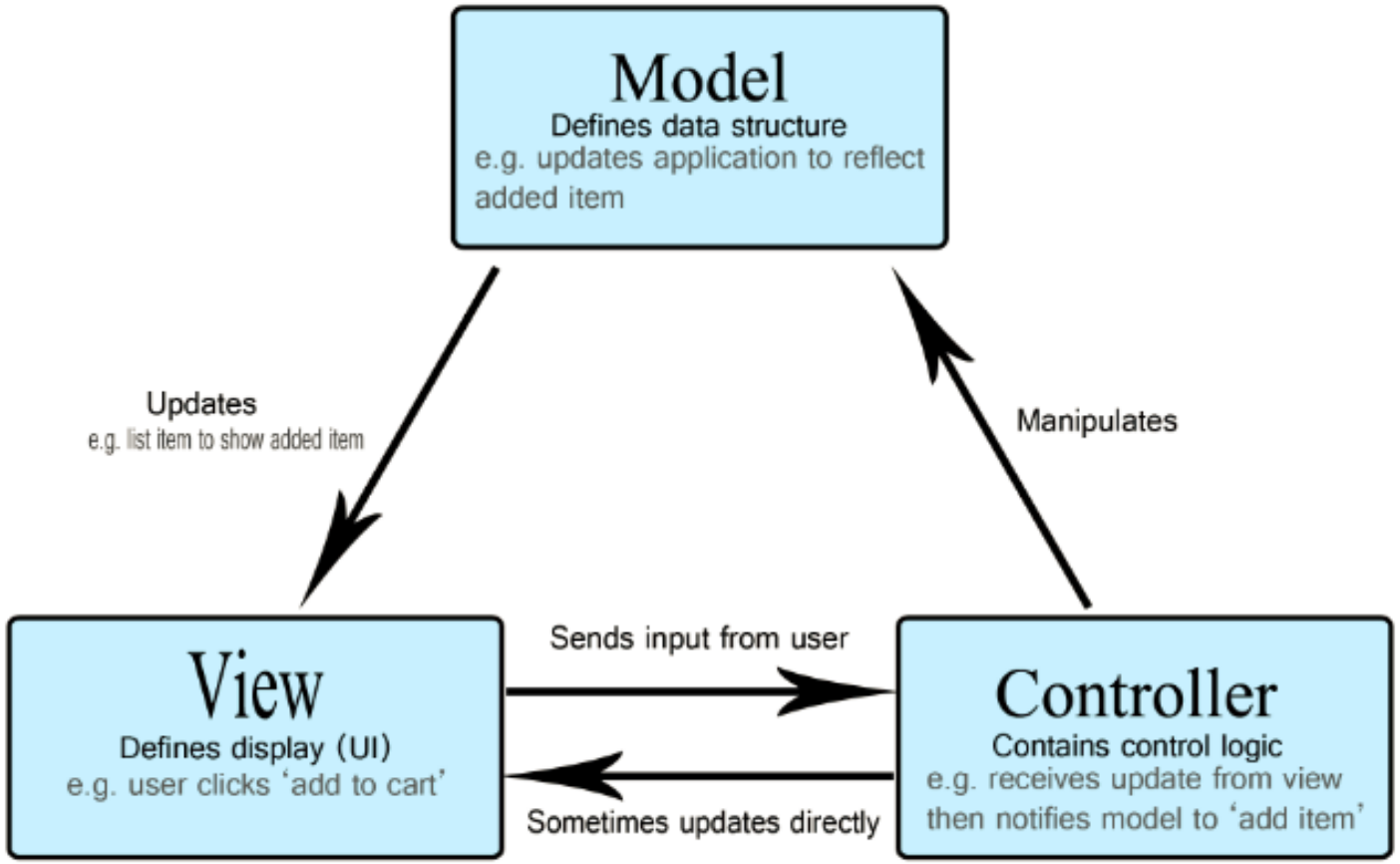
- **框架模式**
- **组件**
- **生命周期**

框架模式

- 框架模式
  - MVC
    - Model（模型）、View（视图）、Controller（控制器）
  - MVP
    - Model（模型）、View（视图）、Presenter（呈现）
  - MVVM
    - Model（模型）、View（视图）、View-Model（视图-模型）

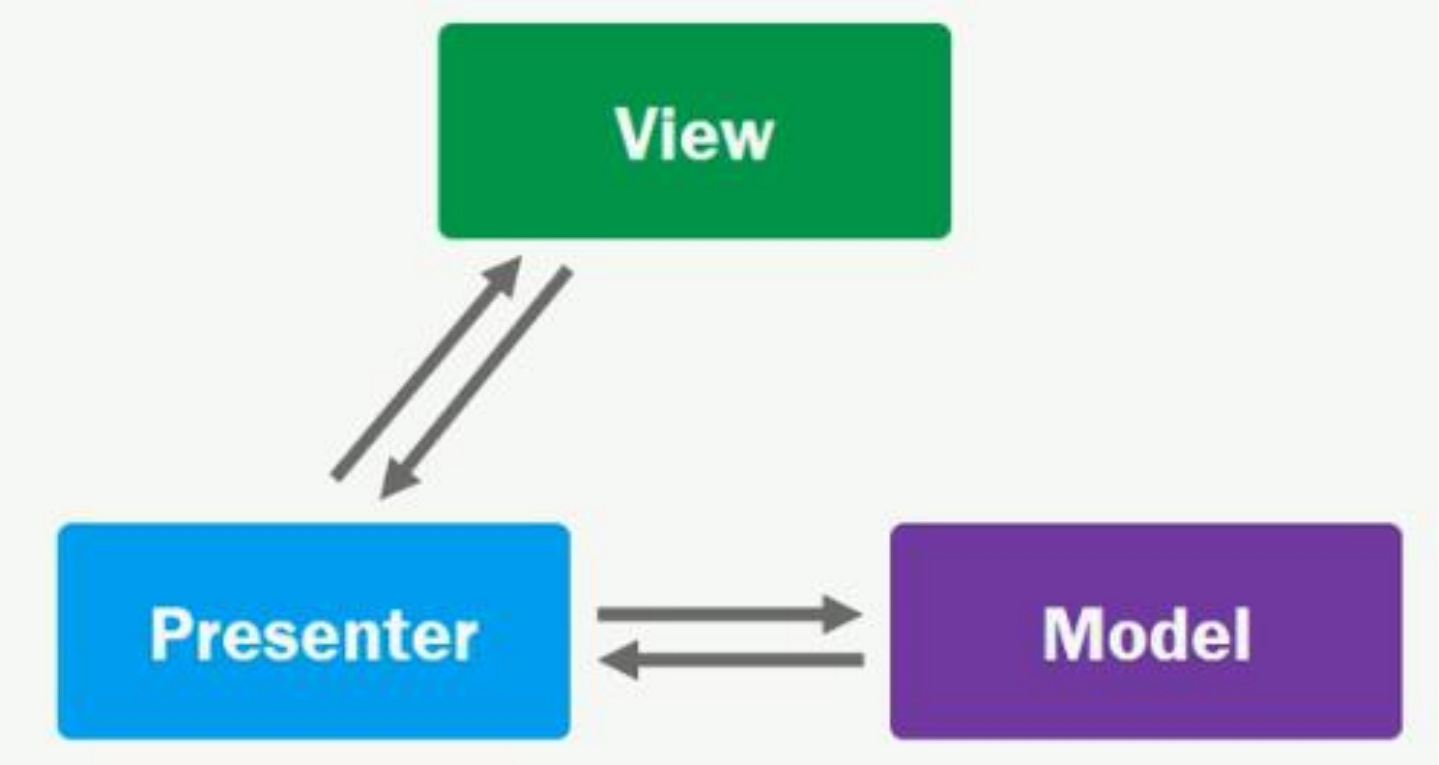
框架模式

- MVC


Model
Defines data structure
e.g. updates application to reflect added item
Updates
e.g. list item to show added item
Manipulates
View
Defines display (UI)
e.g. user clicks 'add to cart'
Sends input from user
Sometimes updates directly
Controller
Contains control logic
e.g. receives update from view then notifies model to 'add item'

框架模式

- MVP



框架模式

- MVVM

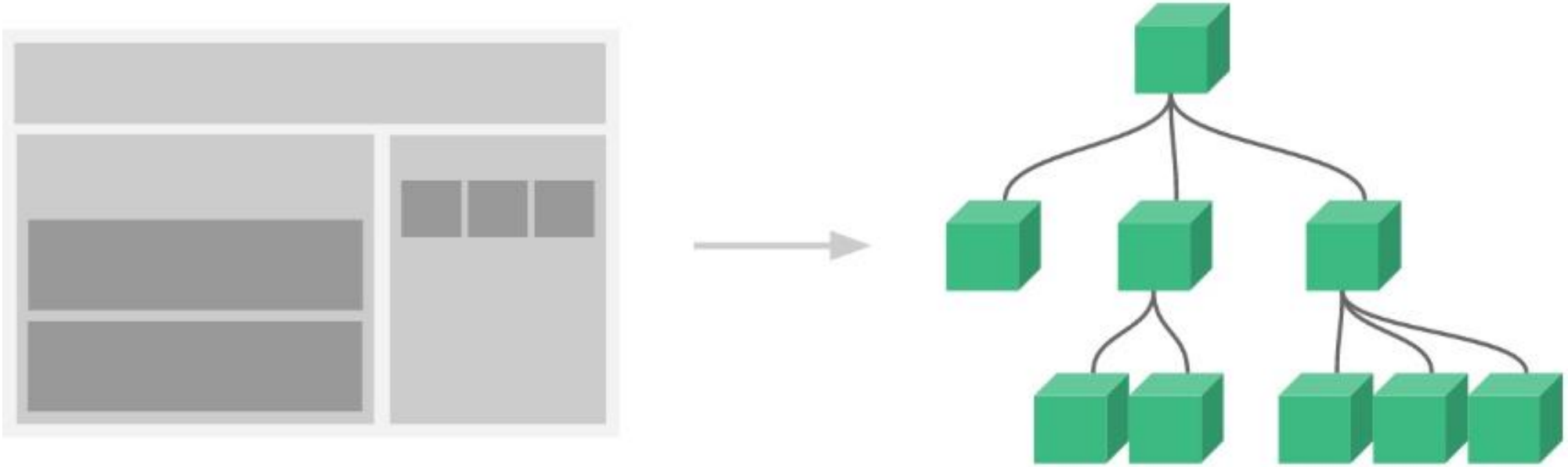

View
双向数据绑定
ViewModel
Ajax
JSON
Model
页面视图
视图数据模型和展现逻辑
业务逻辑和数据库

内容提纲

- 框架模式
- 组件
- 生命周期

组件

- 组件
  - 组件负责控制屏幕上的某一块区域，即视图
  - 组件是 Angular 的核心理念，模块化机制是为组件化服务的



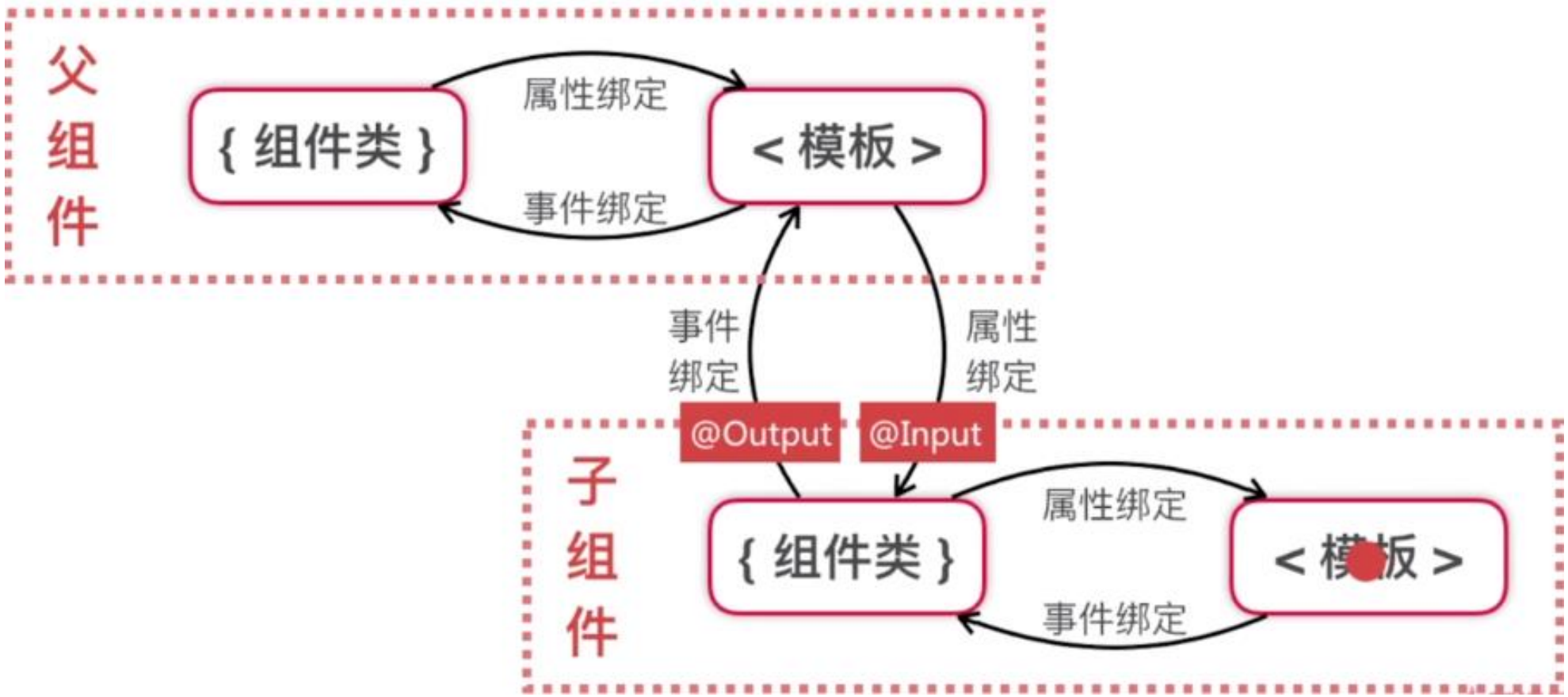
组件

- 创建组件
  - 命令：ng g component components/componentName
  - 在AppModule中声明组件
    - 引入：import { ComponentName } from ‘./components/……';
    - 声明：在 declarations 数组中声明
    - 如果通过命令创建组件，angular会自动添加声明
- 使用组件
  - <app-组件名></app-组件名>

组件

- 组件的基础构成
  - 组件装饰器 -- @component( )
    - 每个组件类必须用 @component 修饰才能成为 Angular 组件
  - 组件元数据
    - selector、template、styles、providers、templateUrl、styleUrls
  - 模板
    - 每个组件只能关联一个模板，这个模板会渲染到页面上
  - 组件类：定义组件的逻辑

组件交互


父组件
{ 组件类 }
属性绑定
事件绑定
< 模板 >
事件绑定
属性绑定
@Output
@Input
子组件
{ 组件类 }
属性绑定
事件绑定
< 模板 >

组件交互

- 组件交互（父组件到子组件） -- @Input
  - 子组件引入 Input
    - import { Input } from '@angular/core';
  - 声明属性
    - @Input( ) 属性名;
  - 绑定属性
    - <app-组件名 [属性名] = “属性值” ></app-组件名>

组件交互

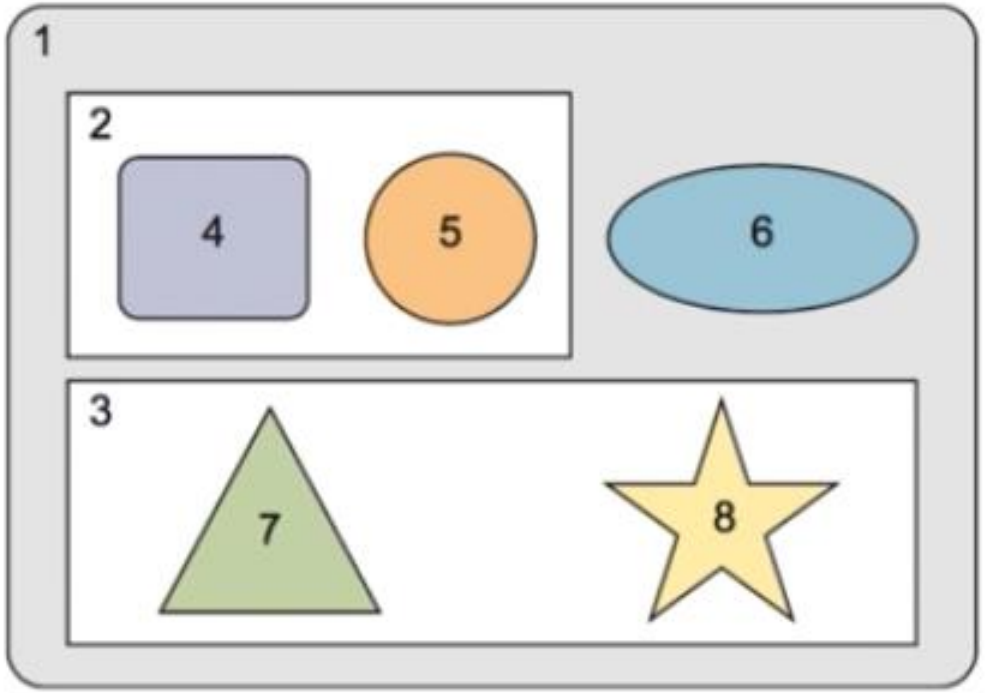
- 组件交互（子组件到父组件） -- @Output
  - 子组件引入 EventEmitter， Output
    - import { Component, EventEmitter, Output } from '@angular/core';
  - 子组件暴露 EventEmitter 属性，并用@Output（ ）装饰
    - @Output( ) eventName = new EventEmitter<T>( );
  - 通过 EventEmitter 的 emit 方法传递数据
    - this.eventName.emit(data)

组件

- 组件交互（子组件到父组件）　（续）
  - 父组件绑定事件，通过事件对象（$event）接收数据
    - <app-组件名（eventName）=“fun($event)”>

组件

- 中间人模式
  - 拥有相同父组件的组件间交互



内容提纲

- 框架模式
- 组件
- **生命周期**

生命周期

- 生命周期
  - Angular 每个组件都存在一个生命周期，从创建，变更到销毁
- 生命周期钩子（lifecycle hook）
  - Angular 提供组件生命周期钩子，把这些关键时刻暴露出来，赋予在这些关键时刻和组件进行交互的能力
  - Angular core 库里定义了**生命周期钩子**接口
  - 每个接口都有唯一的一个钩子方法

生命周期

组件初始化
constructor
ngOnChanges
ngOnInit
ngDoCheck
ngAfterContentInit
ngAfterContentChecked
ngAfterViewInit
ngAfterViewChecked

变更检测
ngOnChanges
ngDoCheck
ngAfterContentChecked
ngAfterViewChecked

组件销毁
ngOnDestroy

生命周期钩子

- ngOnChanges( )
  - 检测到组件输入属性发生变化时调用
- ngOnInit( )
  - 在构造函数之后马上执行复杂的初始化逻辑
  - 在 Angular 设置完输入属性之后，对该组件进行准备
  - 只调用一次
- ngDoCheck ( )
  - 检测被 Angular 忽略的更改

生命周期钩子

- ngAfterContentInit( )
  - 在投影（映射）内容初始化之后调用，只会调用一次
- ngAfterContentChecked( )
  - 在投影（映射）内容变更检测之后调用
- ngAfterViewInit( )
  - 在 Angular 完全初始化了组件的视图后调用
- ngAfterViewChecked( )
  - 在默认的变更检测器完成了对组件视图的变更检测之后调用

生命周期钩子

- ngOnDestroy( )
  - 执行组件注销时的清理工作

生命周期钩子

- ng-content（内容映射）
  - 在组件中嵌入模板代码的**占位符**，告诉 angular 在何处插入模板代码
  - 方便定制可复用的组件
  - 当有多个映射时，模板代码取类名进行区分，ng-content通过select属性进行绑定
    - 父组件中： <app-component>
      ……
      </app-component>
    - 子组件中：<ng-content select= “.header” ></ng-content>

Thank You

河北師范大学软件学院
Software College of Hebei Normal University