

拉勾教育

— 互联网人实战大学 —

《前端高手进阶》

朱德龙 前中兴软创主任工程师

— 拉勾教育出品 —

第28讲：你都了解过哪些编程范式？

代码的可读性、可维护性在软件开发中尤为重要

所以我们在编写代码的时候通常会遵循一些编码规范或风格

如果宏观且抽象地来看，编码风格可以上升为**编程范式**

编程范式 (Programming Paradigm)

也称“编程泛型”或“程序设计法”
是对代码编写方式的一种抽象
体现出了开发者对程序执行的看法

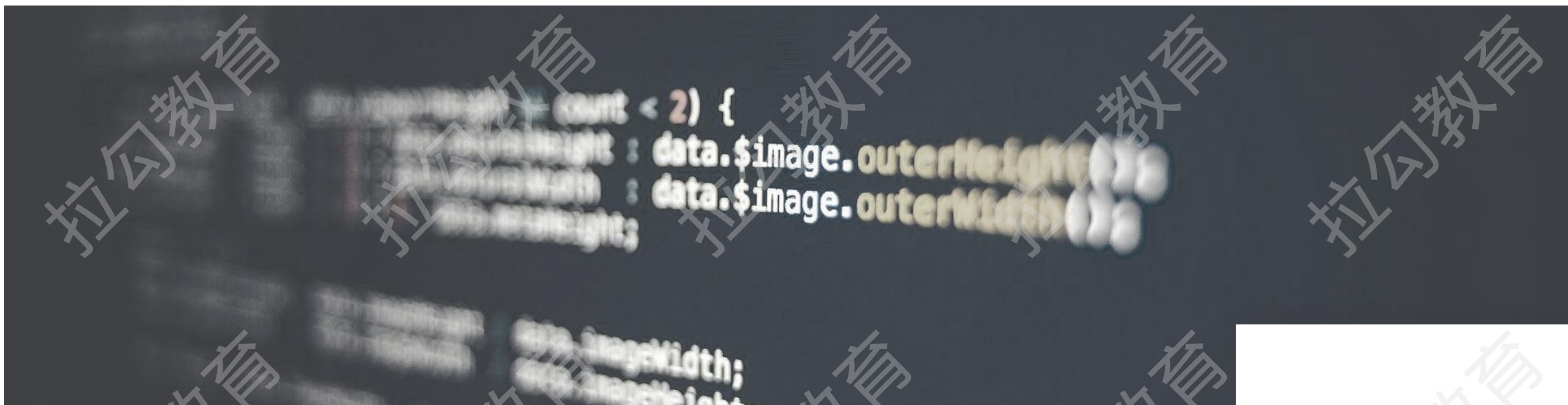


命令式编程 (Imperative Programming)

拉勾教育

— 互联网人实战大学 —

命令式编程是一种古老的编程范式
主要关注点是如何通过具体步骤得到计算结果



面向过程 (Procedural Programming)

拉勾教育

— 互联网人实战大学 —



面向过程

以过程为中心的编程思想

在编程过程中分析出解决问题所需要的步骤

然后再按照执行过程编写代码

面向过程 (Procedural Programming)

拉勾教育

— 互联网人实战大学 —

```
function calc(arr, type) {  
  switch(type) {  
    case 'add':  
      return arr.reduce((acc, cur) => acc + cur, 0)  
      break;  
    case 'multiple':  
      return arr.reduce((acc, cur) => acc * cur, 1)  
      break;  
  }  
}
```

面向对象 (Object-oriented Programming)

拉勾教育

— 互联网人实战大学 —



面向对象

核心是对象，把问题抽象成对象

对象是程序代码中的基本单位

对应代码中的类或类的实例

对象内部封装了数据和方法

面向对象 (Object-oriented Programming)

拉勾教育

— 互联网人实战大学 —

特性

封装、继承、多态



封装 (Encapsulation)

封装

通过限制只有特定类的对象可以访问这一特定类的成员
而它们通常利用接口实现消息的传入传出



拉勾教育

— 互联网人实战大学 —



封装 (Encapsulation)

拉勾教育

— 互联网人实战大学 —

```
// ES5 写法
function Animal() {
  var name = ''
  this.setName = function(a) {
    name = a
  }

  this.getName = function() {
    return name
  }
}
```

封装 (Encapsulation)

拉勾教育

— 互联网人实战大学 —

```
// TypeScript 写法
class Animal {

    private name:string;

    public setName(name:string):void {
        this.name = name
    }

    public getName():string {
        return this.name
    }
}
```

继承 (Inheritance)

继承

在创建子类的时候
会默认获得父类的一些非私有属性和方法



拉勾教育

— 互联网人实战大学 —



继承 (Inheritance)

拉勾教育

— 互联网人实战大学 —

// ES5 写法

```
function Dog() {  
  this.bark = function() {  
    return 'wang wang wang!!!'  
  }  
}
```

```
Dog.prototype = new Animal()
```

// TypeScript 写法

```
class Dog extends Animal {  
  public bark(): string {  
    return 'wang wang wang!!!'  
  }  
}
```


多态 (Polymorphism)

多态

指由继承而产生的相关的不同的类
其对象对同一消息会作出不同的响应



拉勾教育

— 互联网人实战大学 —



多态 (Polymorphism)

拉勾教育

— 互联网人实战大学 —

```
function Cat() {  
  this.bark = function(sound) {  
    if(undefined !== sound && null !== sound) {  
      return sound  
    } else {  
      return '...'  
    }  
  }  
}
```

声明式编程 (Declarative Programming)

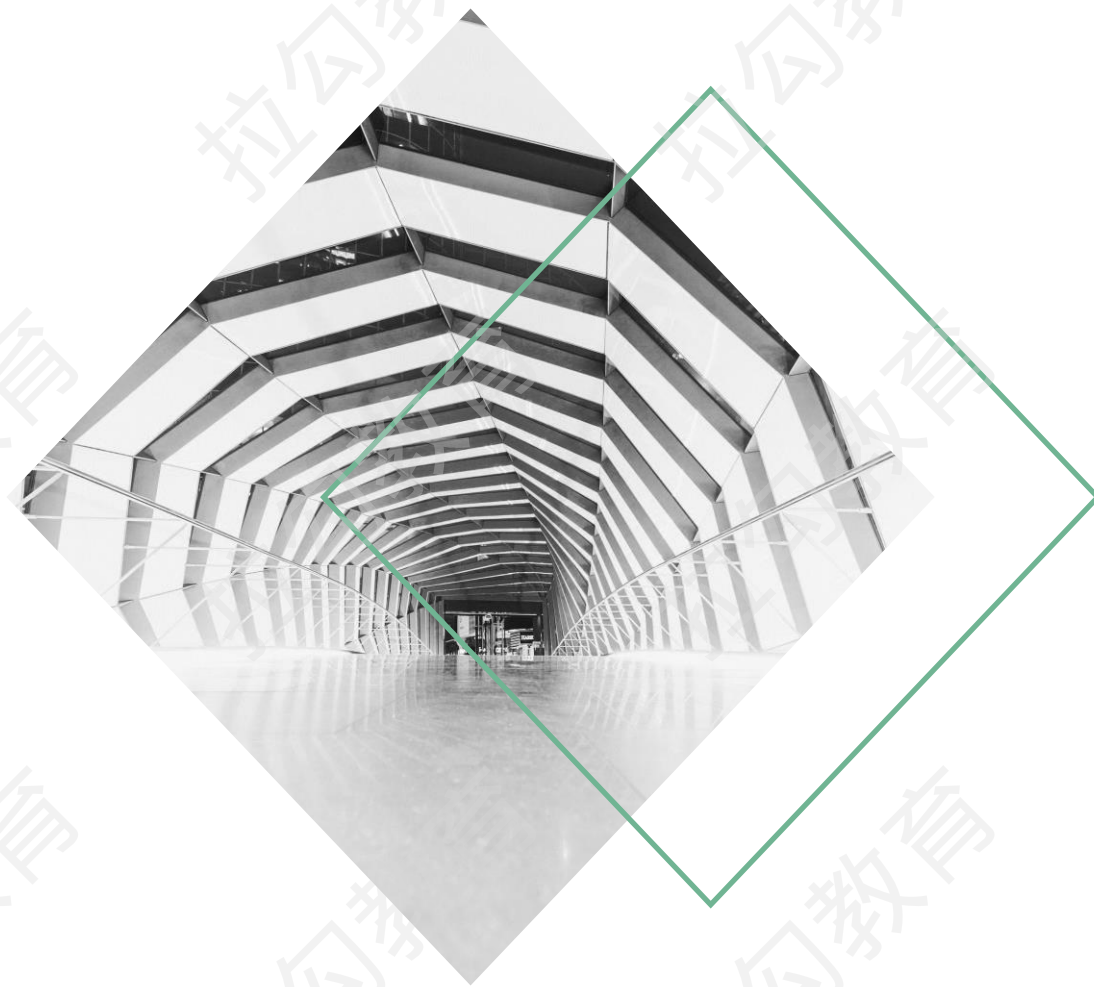
拉勾教育

— 互联网人实战大学 —

声明式编程是一种构建程序的样式

该表达式表示计算逻辑而无须谈论其控制流程

重点是需要完成的工作，而不是应该如何完成



逻辑式编程 (Logic Programming)

拉勾教育

— 互联网人实战大学 —

逻辑式编程通过设置答案须符合的规则来解决问题
而非设置步骤来解决问题





当我们需要解答一个新的问题时

先提出一个新的假设

然后再证明它跟现在的理论无冲突

”

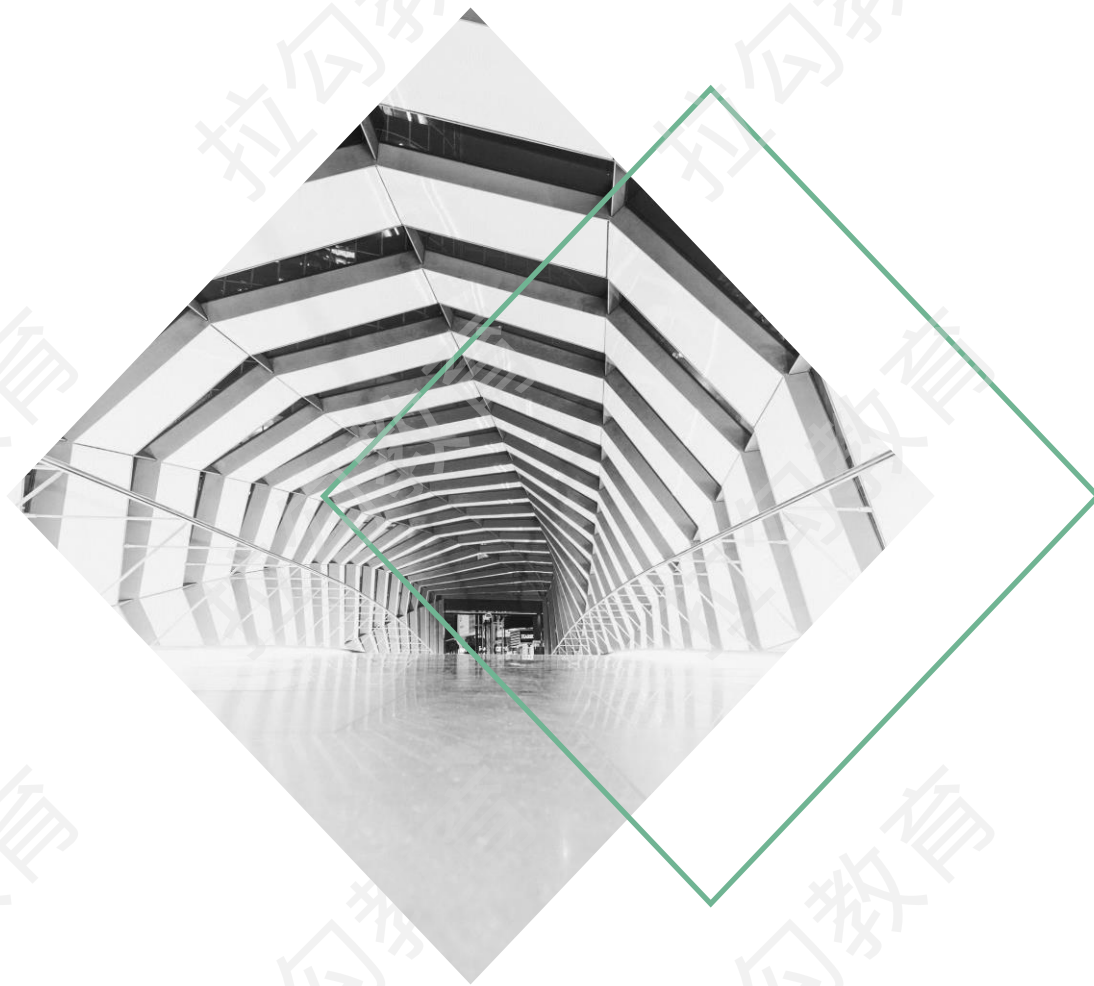
数据驱动编程 (Data-driven Programming)

拉勾教育

— 互联网人实战大学 —

这种编程方法基于数据

程序语句由数据定义而非执行步骤



数据驱动编程 (Data-driven Programming)

拉勾教育

— 互联网人实战大学 —

数据库程序是业务信息系统的核心

并提供文件创建、数据输入、更新、查询和报告功能



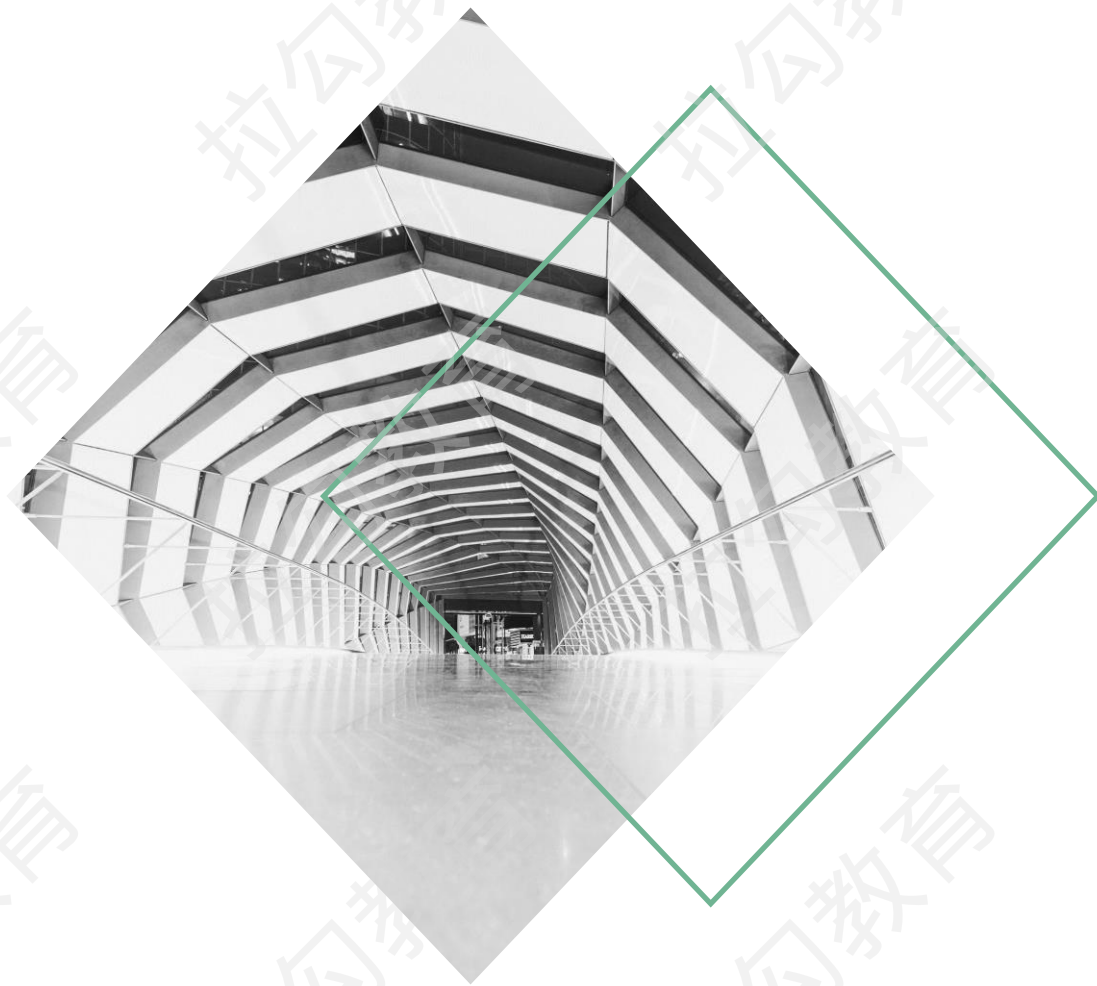
函数式编程 (Functional Programming)

拉勾教育

— 互联网人实战大学 —

函数式编程范式来源于数学而非编程语言

它的关键原理是通过执行一系列的数学函数来得到结果



```
// 非幂等
function rand() {
  return Math.random()
}

// 幂等
function zero() {
  return 0
}
```

// 无副作用

```
function add(a, b){  
  return a + b  
}
```

// 有副作用

```
function setAddition(a, b) {  
  localStorage.setItem('sum', a + b)  
}
```



无状态

高度的可测试性

01

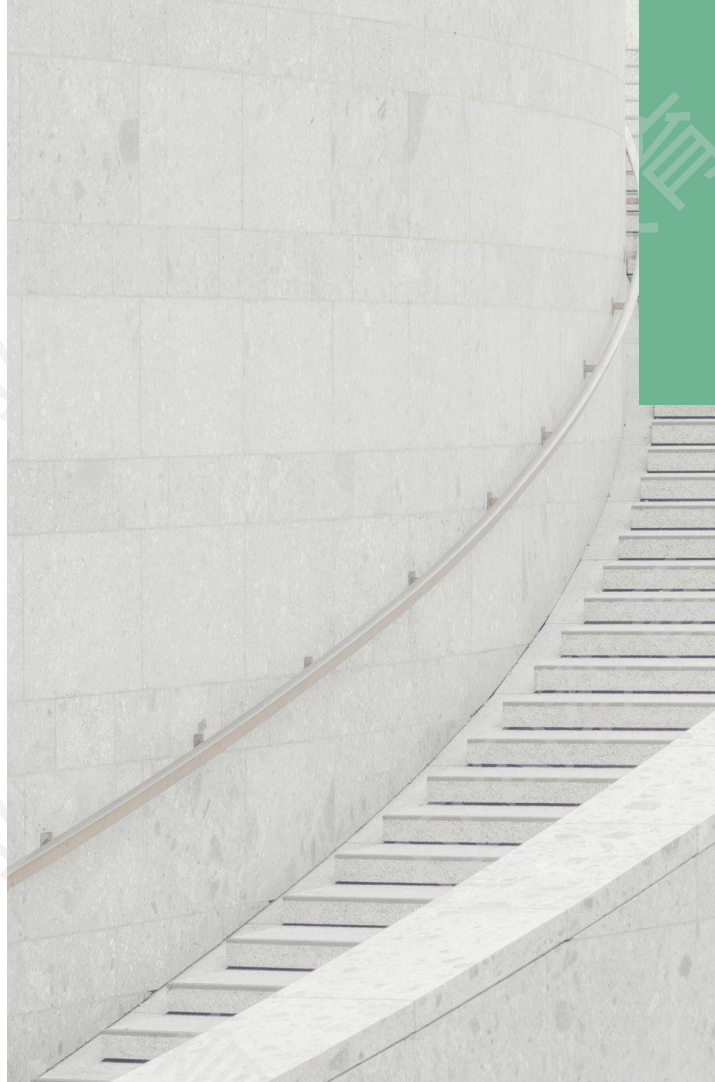
02

高阶函数

指接收一个函数作为参数

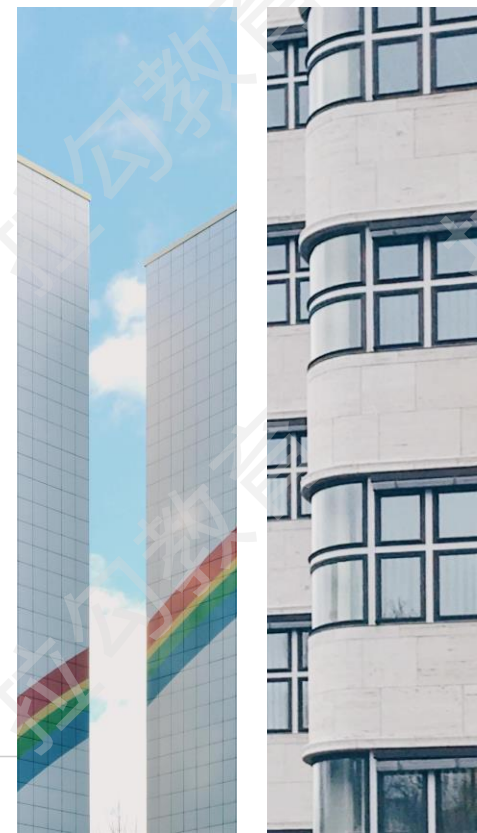
然后返回另一个函数的函数

体现的是一种高级的抽象思维
使用场景也比较多



拉勾教育

— 互联网人实战大学 —



卡瑞化或加里化

把接受多个参数的函数
变换成接受一个单一参数的函数
并且返回接受余下的参数且返回结果的新函数

卡瑞化或加里化

在一个函数中预先填充几个参数
这个函数返回另一个函数
这个返回的新函数将其参数和预先填充的参数
进行合并，再执行函数逻辑

编程范式指的是编程风格，使用合理的编程范式能提升代码的可维护性

编程范式可分为声明式和命令式



你还用过哪些编程范式 ?



Next：第29讲：《框架到底用了哪些设计模式？》

拉勾教育

— 互联网人实战大学 —



下载「拉勾教育App」
获取更多内容