

I1 - OOP概述

亢孟军 武汉大学
mengjunk@whu.edu.cn



授课教师介绍

姓名	亢孟军
Tel	15172448589
Email	mengjunk@whu.edu.cn
办公室	南极电脑城 402室
研究兴趣	空间信息可视化, 地理信息系统开发, machine learning + cartography
答疑时间	提前预约

课程介绍

■ **课程名称:** 面向对象的GIS软件开发

■ **课程性质:** 专业必修 2学分 36学时=24理论+12课间实习

■ **参考教材:** 《C++编程思想》

■ **工具**

✓ QT Creator, MS Visual Studio

■ **语言:** C++

■ **作业:** 1个热身型实习 + 1个持续性实习

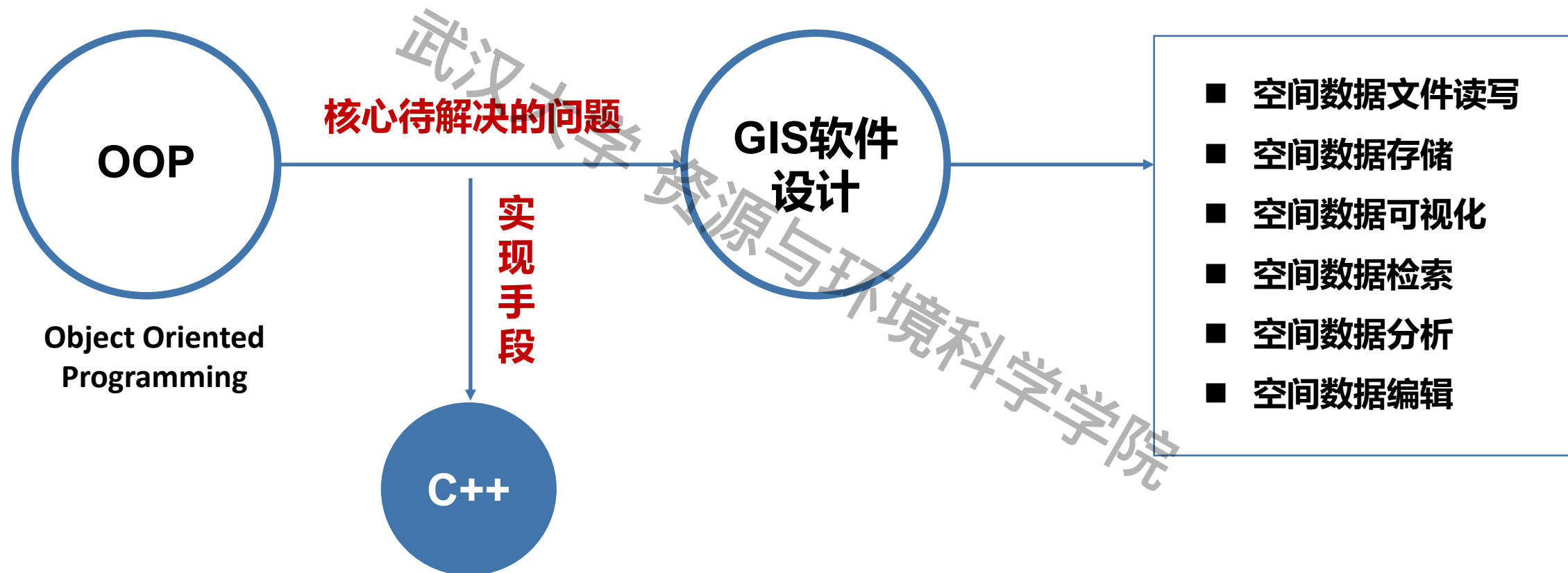
■ **成绩组成**

✓ 期末考试*50% + 实习任务*30% + 讨论/报告*20%



学习方式要求: 课前预习 + 课后实践/复习

课程结构



课程内容结构

章节	类型	课程内容			
L1	基础知识	✓ OOP原理、特征 ✓ 如何应用OOP进行设计 ✓ 案例分析	L3	基础知识	C++基本语法-1 ✓ 执行控制语句 ✓ 运算符及使用 ✓ 数据类型 ✓ 存储空间分配
A1	应用知识	QT Creator 安装配置			
A2	应用知识	Visual Studio + QT Plugin安装与配置			
A3	应用知识	QT Creator调试案例及说明	L4	基础知识	C++基本语法-2 ✓ 作用域 ✓ 指针 ✓ 引用 ✓ 函数及函数地址 ✓ copy-by-value 和 copy-by-reference (举例, 交换两个整数)
L2	基础知识	C++基础入门知识点 (方便理解后续的案例) ✓ 编写第一个C++程序 “hello world” ✓ 编译: 语言的翻译过程 ✓ include ✓ 命名空间 ✓ 输入输出流 ✓ 字符串 ✓ 文件读写 ✓ vector	A4	应用知识	✓ 编码规范 ✓ 类、变量、函数命名的原则 ✓ 注释及注释内容
			L5	基础知识	✓ 复合数据类型 ✓ 数组与指针的关系

课程内容结构

L6	基础知识	类和对象-1 <ul style="list-style-type: none">✓ 对象和类的定义、形式、组成✓ 一个class的案例，与struct的传承关系✓ 访问控制：private, protected and public✓ 头文件的作用与应用	L9	基础知识回顾	内存管理与指针应用（作为回顾课，对内存管理的要点进行复习）
L7	基础知识	<ul style="list-style-type: none">✓ 函数重载与默认参数✓ 内联函数；✓ 常量的应用；✓ const与指针✓ 常量在函数参数描述的应用✓ 类中的const	L10	应用知识	<ul style="list-style-type: none">✓ 简单要素模型SFS✓ GML✓ GeoJSON✓ KML✓ WKT & WKB
L8	基础知识	类和对象-2 <ul style="list-style-type: none">✓ 构造函数✓ 析构函数✓ 避免浅拷贝✓ 拷贝构造函数的各种情况	L11	应用知识	<ul style="list-style-type: none">✓ 空间数据文件格式✓ GDAL空间对象结构组织✓ QGIS空间对象结构组织
			L12	基础知识	功能性内容： <ul style="list-style-type: none">✓ 命名空间✓ 静态成员的应用&全局变量✓ 静态成员初始化

课程内容结构

L13	基础知识	<div><div>✓</div> STL通用容器</div> <div><div>✓</div> QGIS中STL应用及案例（空间数据组织）</div>
L14	应用知识	<div><div>✓</div> VCT格式</div> <div><div>✓</div> VCT格式解析案例</div> <div><div>✓</div> 格式转换，与shp</div>
L15	应用知识	<div><div>✓</div> 图形设备接口介绍</div> <div><div>✓</div> OpenGL介绍及案例（三维地球）</div> <div><div>✓</div> Opengl环境配置</div> <div><div>✓</div> 矢量数据绘制</div>
L16	基础知识	<div>内存管理基础</div> <div><div>✓</div> 动态对象的创建</div> <div><div>✓</div> 以QGIS包为例介绍相关内存使用案例</div>
L17	应用知识	<div><div>✓</div> QT UI介绍及示例、操作案例</div> <div><div>✓</div> 消息传递机制</div> <div><div>✓</div> QGIS UI案例分析</div>

L18	应用知识	<div><div>✓</div> QGIS MapControl控件事件介绍</div> <div><div>✓</div> 如何设计并开发一个地图控件，案例介绍</div>
L19	基础知识	<div><div>✓</div> 继承</div> <div><div>✓</div> 组合</div> <div><div>✓</div> 继承与组合的关系，以及选择</div> <div><div>✓</div> 名字自动隐藏</div> <div><div>✓</div> 继承二义性</div> <div><div>✓</div> 向上类型转换</div>
L20	基础知识	<div><div>✓</div> 多态性</div> <div><div>✓</div> 虚函数及实现机制</div> <div><div>✓</div> C++如何实现晚绑定</div> <div><div>✓</div> 抽象基类和纯虚函数</div>
L21	基础知识	<div><div>✓</div> 重载和重新定义</div> <div><div>✓</div> 虚函数和构造函数</div> <div><div>✓</div> 析构函数和虚析构函数</div> <div><div>✓</div> 向下类型转换</div>

课程安排

L22	应用知识	1. 空间检索、空间索引 2. 网格索引设计实现 3. 图属、属图查询的设计实现案例
L23	基础知识	1. 异常处理 2. 日志管理
L24	基础知识	1. 设计模式 2. QGIS中设计模式的案例分析
L25	应用知识	1. 空间数据编辑的原理 2. QGIS空间数据编辑案例介绍

L26	基础知识	✓ 并发 ✓ 多线程原理及开发
L27	应用知识	✓ 图形渲染的效率优化问题 ✓ 双缓存 ✓ OpenGL的渲染管线
L28	应用知识	✓ 核密度分析原理 ✓ 核密度分析实现

课程目标



从一个测试开始

- IDE(集成开发环境)是什么，你用过哪些？
- 什么是变量？深入的讲下自己的理解
- 用什么类型数据存储你存储在银行里面的钱数？
- 请写一个计算两个整数和的函数sum？
- 请写一个交换两个整数数值的函数swap()，并能运行之？

答案

■ 使用的IDE(集成开发环境)是什么?

- ✓ **答案:** Integrated Development Environment Software, 是进行程序开发的应用程序, 一般包括代码编辑器、编译器、调试器、图形用户界面工具以及其他可装载的辅助插件

■ 什么是变量?

- ✓ **答案:** 变量来源于数学, 是计算机语言中能储存计算结果或能表示值抽象概念。变量可以通过变量名访问

■ 用什么类型数据存储你存储在银行里面的钱数?

- ✓ **答案:** 高精度浮点型, double in C++ / C

答案

■ 请写一个计算两个整数和的函数Sum

✓ 答案:

```
#include <stdio.h>

int Sum(int,int);

int _tmain(int argc, _TCHAR* argv[])
{
    printf("3 + 5 = %d", Sum(3,5));
    int c=0;
    scanf("%d",&c);
    return 0;
}

/*
 * 求两个整数之和
 */
int Sum(int a,int b)
{
    return a+b;
}
```

答案

■ 请写一个交换两个整数数值的函数，并能运行之

✓ 答案:

```
void Swap1(int& a,int& b)
{
    int temp=a;
    a=b;
    b=temp;
}

void Swap2(int* a,int* b)
{
    int temp=*a;
    *a=*b;
    *b=temp;
}
```

```
int _tmain(int argc, _TCHAR* argv[])
{
    int v1=10,v2=20;
    Swap1(v1,v2);
    printf("v1=%d,v2=%d\n",v1,v2);
    Swap2(&v1,&v2);
    printf("v1=%d,v2=%d\n",v1,v2);

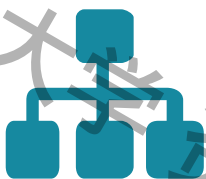
    int c=0;
    scanf("%d",c);
    return 0;
}
```

本节课内容



OOP定义和特征

什么是OOP



OOP的应用

如何应用OOP



程序语言的本质

程序语言到底是什么？



案例分析

从一个案例讲起



OOP的定义和特征

- OOP定义 OOP内涵

资源与环境科学学院

OOP的定义

■ OO: Object-Oriented, 面向对象

- ✓ 是一种对现实世界理解和抽象的方法
- ✓ 是一种对世界**离散的认识**
- ✓ **核心**: 把相关的**数据**和**方法**组织为一个整体来看待, 从更高的层次来进行系统建模, 更贴近事物的自然运行模式
- ✓ 相对于**面向过程**而言

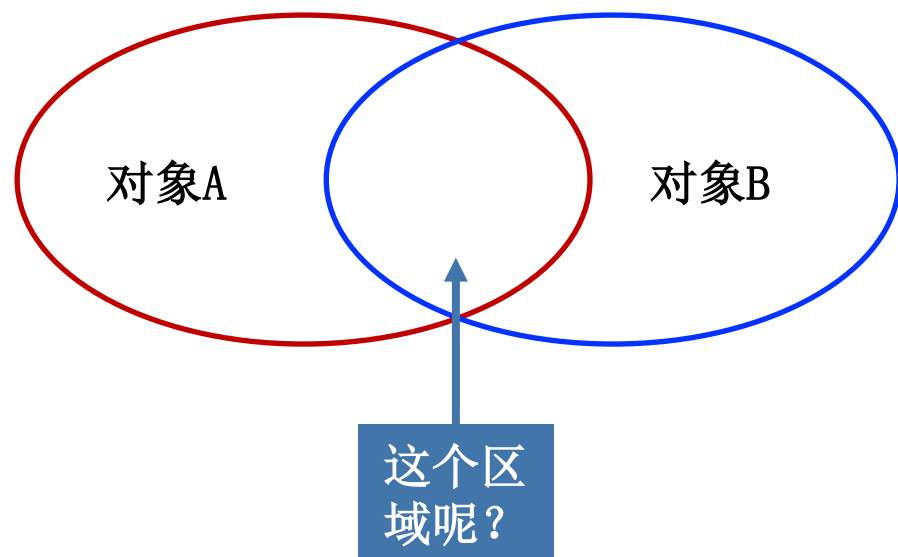
■ OOP: Object-Oriented Programming, 面向对象程序设计

- ✓ 是面向对象思想在软件设计上的应用

OOP 的内涵

■ 一种思维方式

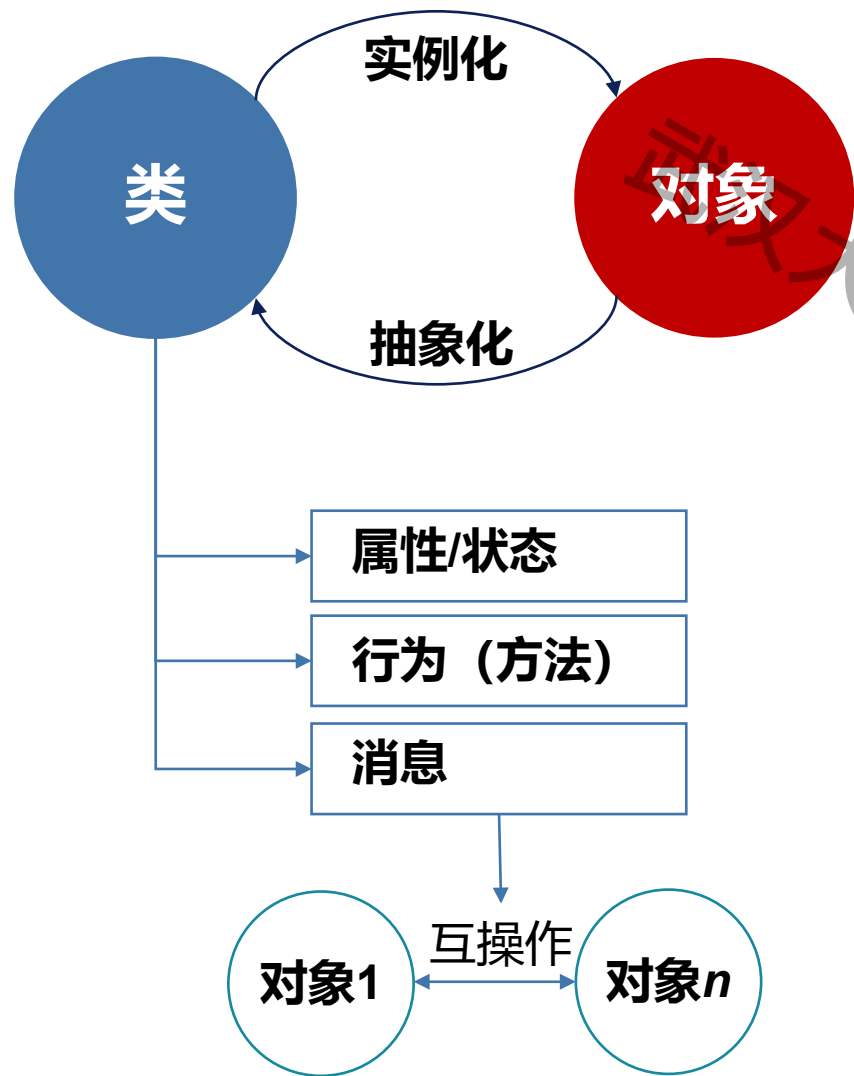
- ✓ 万物皆对象
- ✓ 任何系统可看作是对对象的集合
- ✓ 系统的触发和运行是由系统内外对象相互作用引起的



■ 一种设计范式

- ✓ 对象是程序的基本组成单元
- ✓ 每个对象都属于某一种类型
- ✓ 每个对象都有独立的存储区, 该存储区可包含其他对象存储区
- ✓ 抽象, 对象是现实世界的抽象
- ✓ 相同类型对象可接收相同的消息
- ✓ **归纳**: 对象 -> 类; **演绎**: 类 -> 对象
- ✓ **自上而下**: 父类 -> 子类; **自下而上**: 子类 -> 父类

OOP的基本组成要素



■ **对象**：是由描述其**属性**的数据以及对这些数据施加的**一组操作**封装在一起构成的统一体

■ **类**：是一组具有相同属性和行为的对象的抽象

■ **属性**：又称状态，反映对象的某些静态特征

- 在C++，**属性=成员变量**

■ **行为**：又称操作、功能，反映对象可执行的动作

- 在C++，**行为=成员函数**

■ **消息**：对象间发出的互操作的请求。

- 调用对象的成员函数就是向该对象发送消息
- OO系统通过消息传递进行激活

OOP的基本特征

■ 抽象

- ✓ 构建问题空间中的事务和它们在解空间中的表示的过程
- ✓ 对机器建模 -> 对问题建模

■ 封装

- ✓ 把数据和函数捆绑在一起的能力，可以用于创建新的数据类型

■ 隐藏

- ✓ 设定边界，确定不同角色可访问与不可访问
- ✓ 角色：**类创建者**和**客户程序员**

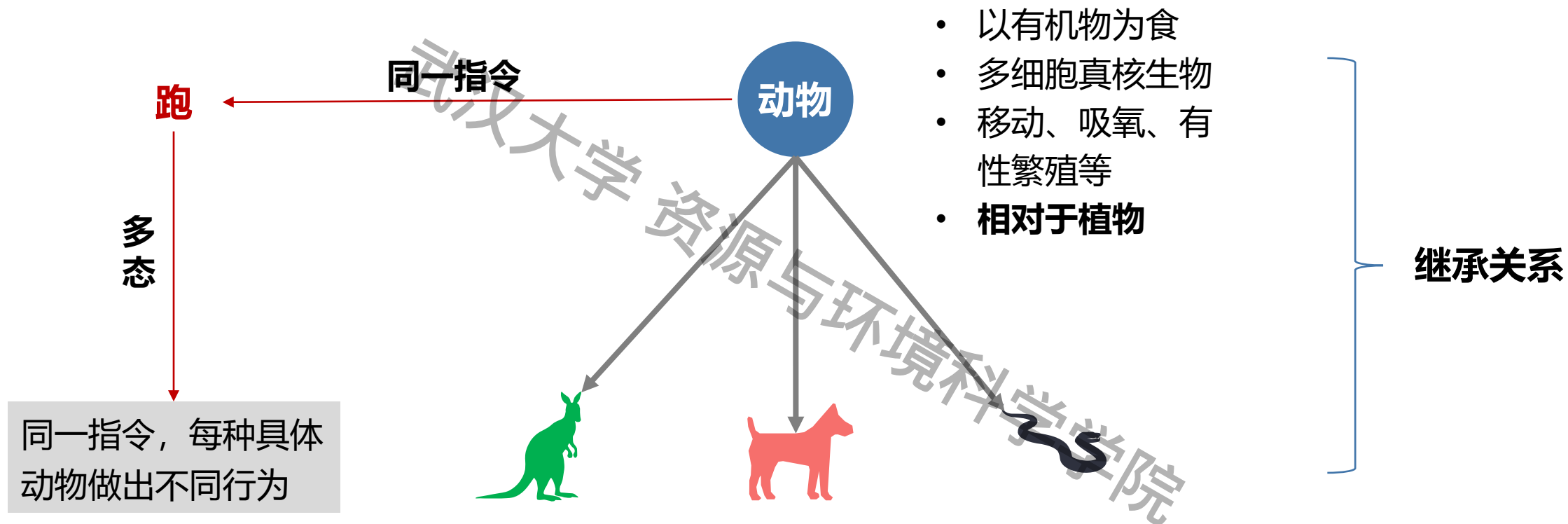
■ 继承

- ✓ 从基类克隆并扩展其数据和功能的能力

■ 多态

- ✓ 相同的操作指令执行不同的具体操作实质的能力
- ✓ 是泛化的继承

OOP的基本特征



C++中的OOP实现

C++实现

class

class

public
protected
private

类继承

重载、模板、
虚函数



OOP特征

抽象

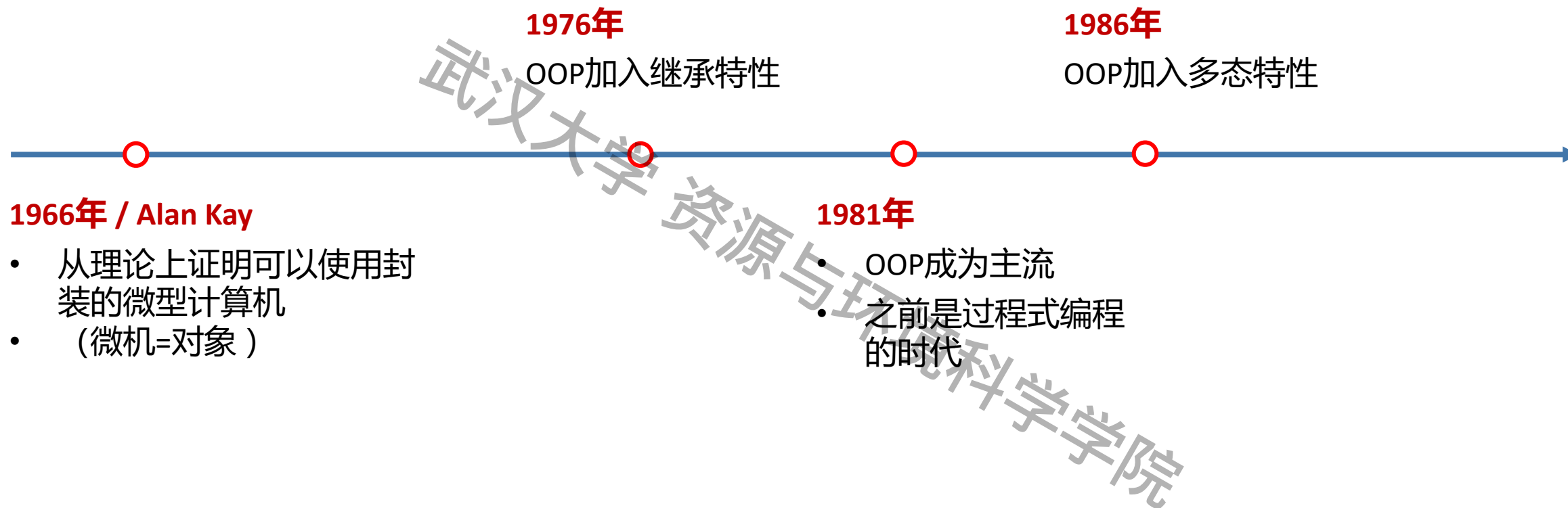
封装

信息隐藏

继承

多态

OOP 的发展过程



OOP VS POP

■ **POP**: Procedure-Oriented Programming, 面向过程程序设计, 以C语言为代表

■ **本质**: 另外一种思维和设计范式

武汉大学 资源与环境科学学院

OOP VS POP

条目	POP	OOP
区别	函数 是程序的基本组成	对象 是程序的基本组成
	函数重要性大于数据 系统是函数的顺序执行序列	数据最重要，是现实世界属性和行为的抽象
	自上而下的设计模式	自下而上的设计模式
数据传递	函数间便于传递数据	通过成员函数传递函数
扩展	不易扩展	易扩展
数据获取	全局变量应用较多，为了函数在任意位置访问	通过访问权限管理数据： public, private, protected
数据封装	无特定机制进行数据封装	支持数据封装，更安全
	不允许重载	支持函数重载和符号重载
支持语言	C, VB, FORTRAN, Pascal.	JAVA, VB.NET, C#.NET.

OOP – 优点

■ 转换代价小

- 兼容C语言
- 同时又提高了生产效率

■ 较好的运算效率和更优的生产效率

- 显著提高了生产效率
- 损失的运算效率有限 ($\pm 10\%$ 以内)

■ 系统更易于表达和理解

- 用问题空间属于描述问题的解
- 易于维护

■ 尽量使用库

■ 基于模板 (C++ Template) 的代码重用

■ 完备的异常处理机制

■ 更适合大型程序设计

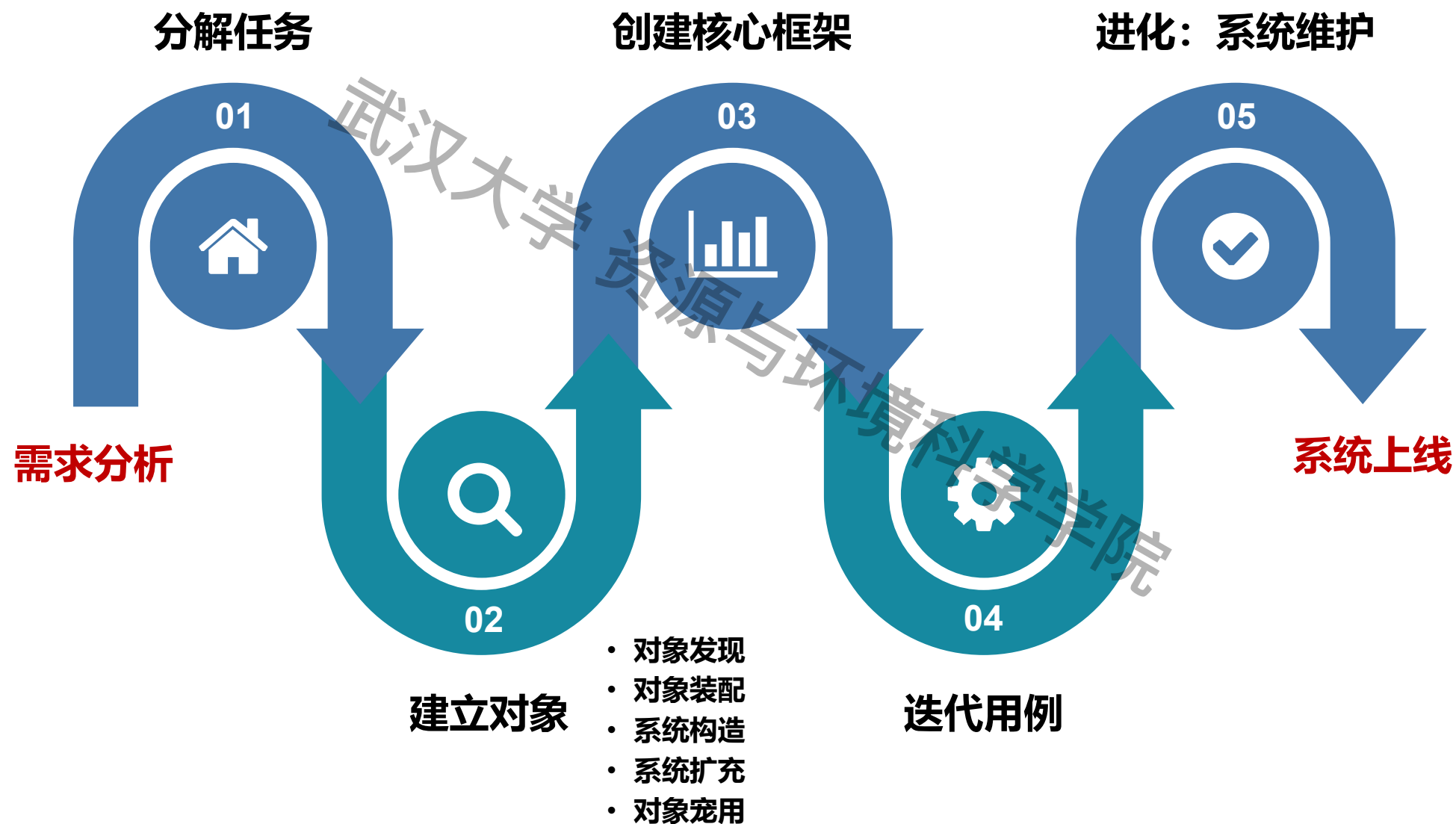


如何应用OOP进行设计

- 如何设计一个应用程序，利用OOP思想

资源与环境科学学院

OOP设计的过程



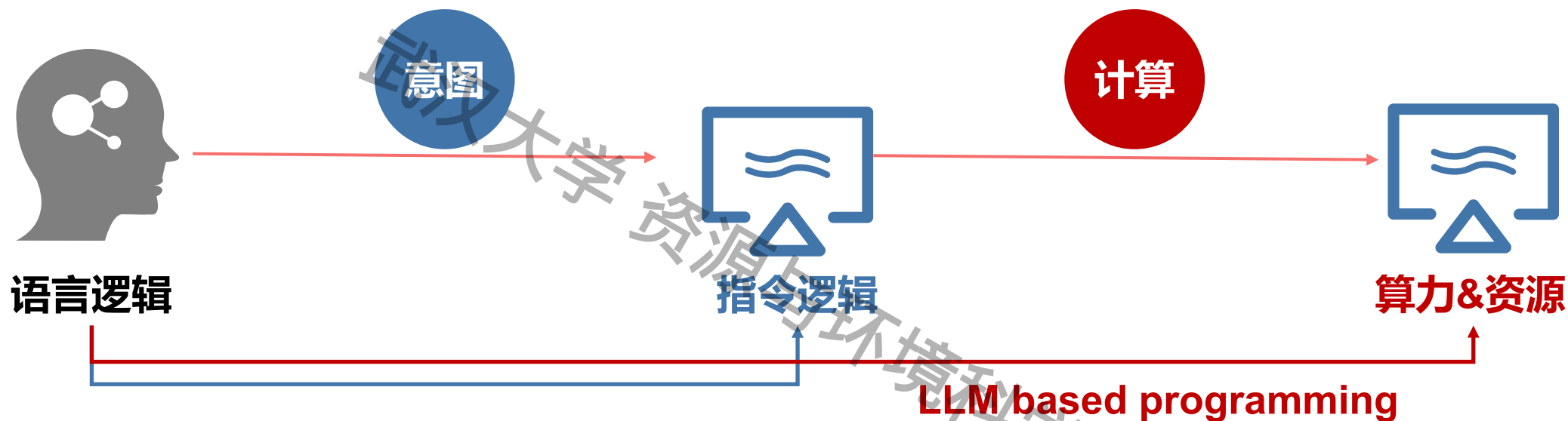


程序语言的本质

- 程序语言的作用是什么？
- 为什么要如此设计？

资源与环境科学学院

程序语言的本质



程序语言

- 人类语言的语义
- 指令语言的规则
- 问题建模
- 一种交互

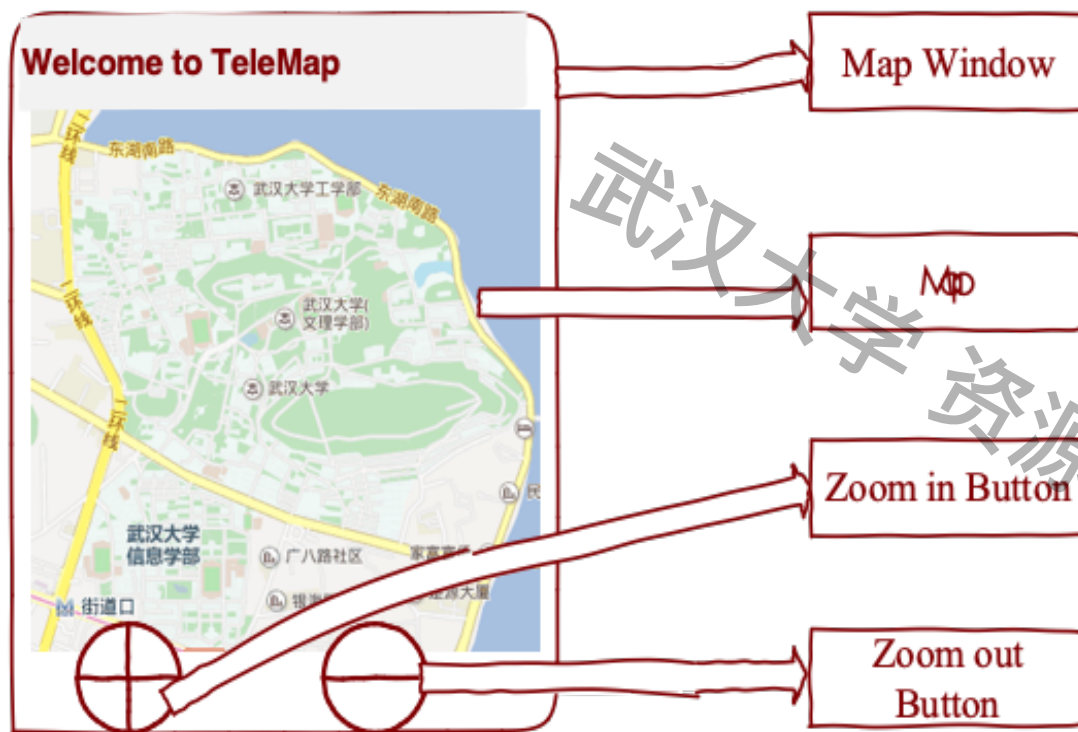


案例分析

- 简单的小案例，对OOP有个基本了解

资源与环境科学学院

OOP设计流程 – 案例



```
class MapWindow{  
    Button btn_zoomin;  
    Button btn_zoomout;  
    Map map;  
    //member function  
    void buttonClick();  
}
```

```
class Map{  
    void cmd_zoomout();  
    void cmd_zoomin();  
}
```

```
void buttonClick(){  
    if(btn_zoomin)  
        map.cmd_zoomin();  
    else if(btn_zoomout)  
        map.cmd_zoomout();  
}
```

```
class Button{  
    int id;  
    void onclick(int eventID);  
}
```

Thanks.

亢孟军 武汉大学
mengjunk@whu.edu.cn

