

面向对象程序设计

基础知识

2020 年 春

耿楠

计算机科学系
信息工程学院

西北农林科技大学
NORTHWEST A&F UNIVERSITY
中国 · 杨凌



基本特性

是什么?

为什么?

发展史

目的

程序结构

基本程序

附件下载

1

▶ 对象

- ▶ 目标
- ▶ 恋爱的对象
- ▶ 描写或写实的人或物

▶ Object

- ▶ Something perceptible by one or more of the senses, especially by vision or touch; a material thing
- ▶ The purpose or goal of a specific action
- ▶

42





▶ 对象（现实世界）

- ▶ 现实世界中的某个具体的事物（实物或实体）
- ▶ 一个封装了属性(attribute)与行为(behavior)的实体

▶ 对象（计算机世界）—**数据结构**

- ▶ 数据域
- ▶ 方法
- ▶ 相互作用关系





▶ 属性

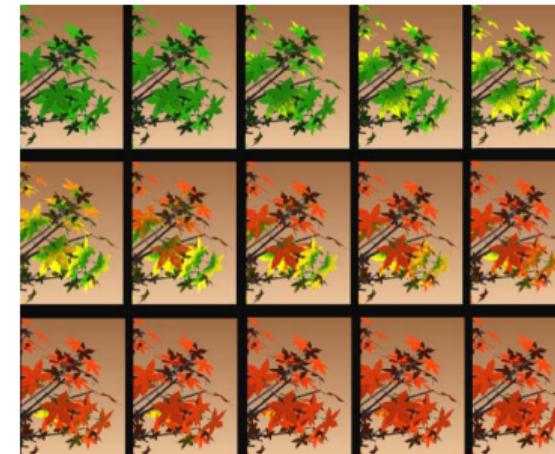
▶ 树干树枝、树叶、树龄、纹理、荷尔蒙……





▶ 方法 (行为、操作)

- ▶ 生长、四季变化、修剪、运动、叶枝摩擦、声音.....



4

42





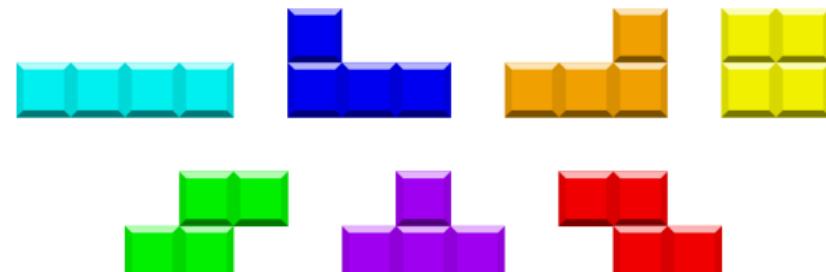
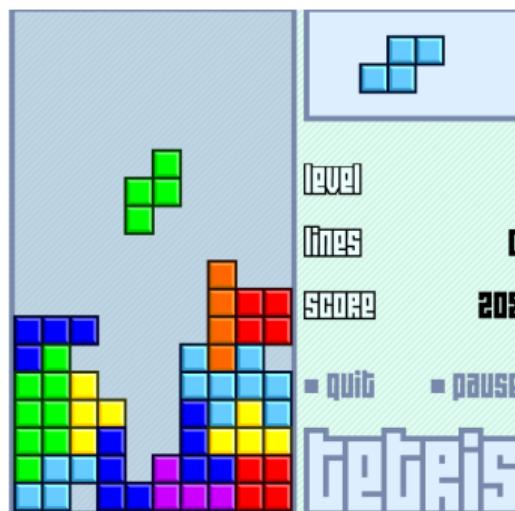
5

▶ 属性

- ▶ 颜色，四个方块的布局，大小，位置，速度，.....

▶ 行为

- ▶ 平移，旋转，加速，碰撞检测，显示，.....





消息 (message)

| 基本特性

OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

6

▶ 类

```
CTetromino{  
    属性: 颜色, 四个方块的布局, 大小, 位置, 速度, ...  
    行为: 平移, 旋转, 加速, 碰撞检测, 显示, ...  
};
```

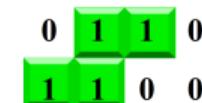
▶ 类对象

```
CTetromino I, J, L, O, S, T, Z;
```

▶ 操作对象

- ▶ 向对象发送消息
- ▶ 通过调用对象的操作实现

```
CTetromino S(绿色, 01101100);  
S. 平移 (-1, 0);
```





消息 (message)

| 基本特性

OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

6

▶ 类

```
CTetromino{  
    属性: 颜色, 四个方块的布局, 大小, 位置, 速度, ...  
    行为: 平移, 旋转, 加速, 碰撞检测, 显示, ...  
};
```

▶ 类对象

```
CTetromino I, J, L, O, S, T, Z;
```

▶ 操作对象

- ▶ 向对象发送消息
- ▶ 通过调用对象的操作实现

```
CTetromino S(绿色, 01101100);  
S. 平移 (-1, 0);
```

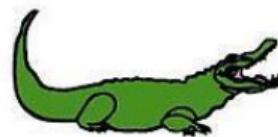
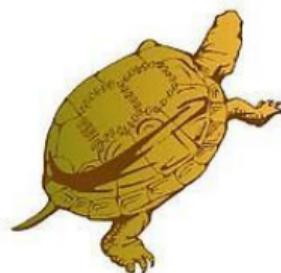




▶ 抽象

- ▶ 抽取共性
- ▶ 问题空间的事物 ⇒ 解空间的面向对象的概念

▶ 示例 1：两栖动物



▶ 示例 2：斑马





8

- ▶ 对象的属性和服务结合成一个独立的系统单位
- ▶ 隐蔽对象的内部细节
- ▶ 保留有限的对外接口
- ▶ 代码安全性
- ▶ 示例：





▶ 类 (三栏矩形)

- ▶ 类名
- ▶ 属性
- ▶ 操作
- ▶ 对外可见性
 - ▶ “+”—公有
 - ▶ “-”—私有

类名
- 属性 1
- 属性 2
+ 操作 1()
+ 操作 2()

▶ 对象 (矩形)

- ▶ 对象名称
- ▶ 类型
- ▶ 属性值

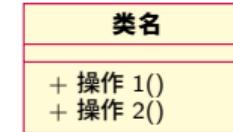
Circle
- radius : double
- center_x : int
- center_y : int
+ area() : double
+ perimeter() : double
+ move(in newx : int, in newy : int) : void
+ scale(in factor : double) : void





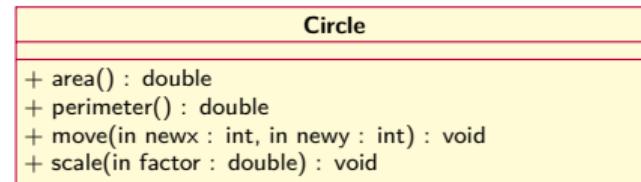
▶ 类 (三栏矩形)

- ▶ 类名
- ▶ 属性
- ▶ 操作
- ▶ 对外可见性
 - ▶ “+”—公有
 - ▶ “-”—私有



▶ 对象 (矩形)

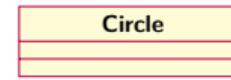
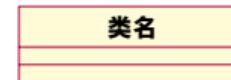
- ▶ 对象名称
- ▶ 类型
- ▶ 属性值





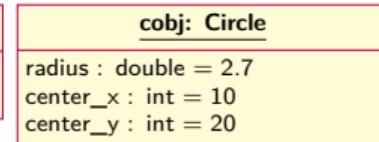
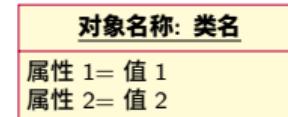
▶ 类 (三栏矩形)

- ▶ 类名
- ▶ 属性
- ▶ 操作
- ▶ 对外可见性
 - ▶ “+”—公有
 - ▶ “-”—私有



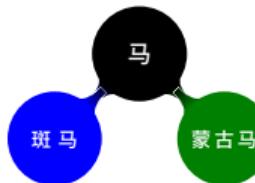
▶ 对象 (矩形)

- ▶ 对象名称
- ▶ 类型
- ▶ 属性值





- ▶ 一般类 ⇒ 特殊类 (具备全部属性与行为)
- ▶ 代码重用
- ▶ 示例 1:





► 示例 2(伪代码):

```
Animal 类 {  
    Animal 类 (const string& name) : name(name) {}  
    virtual string talk() = 0;  
    const string name;  
};
```

```
Cat 类 : Animal 类 {  
    Cat 类 (const string& name): Animal 类 (name) {}  
    string talk() { return "喵喵!"; }  
};
```

```
Dog 类 : Animal 类 {  
    Dog 类 (const string& name): Animal 类 (name) {}  
    string talk() { return "汪汪!"; }  
};
```





继承性 (Inheritance)

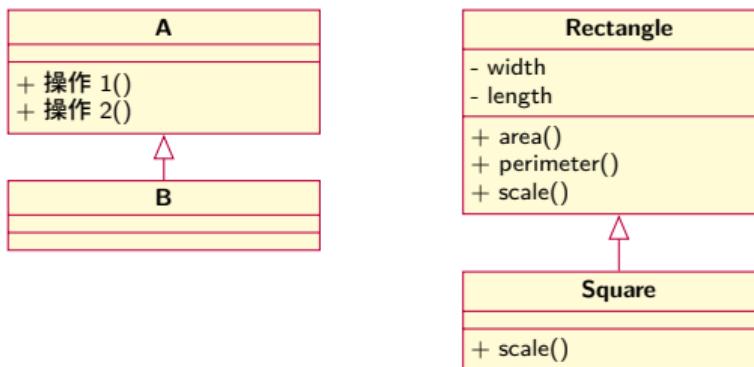
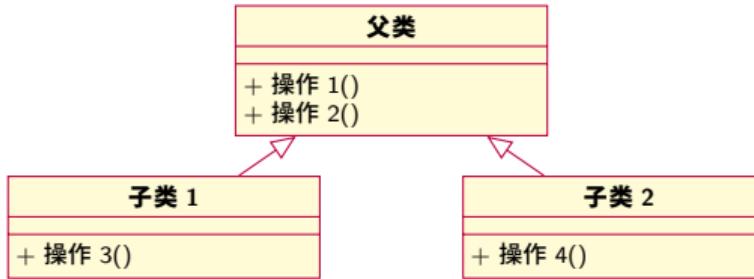
| 基本特性

OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

14

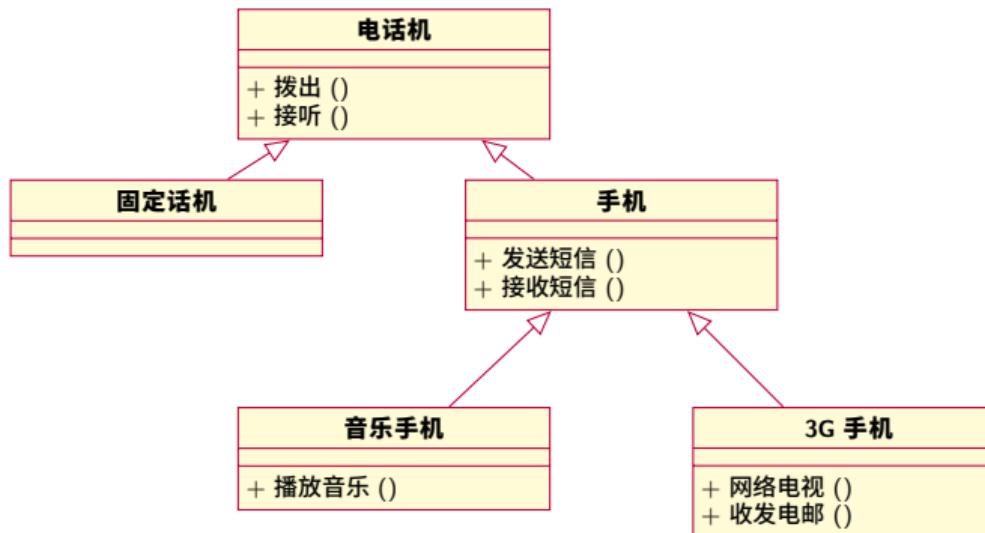
▶ 类的继承



42



▶ 类的继承

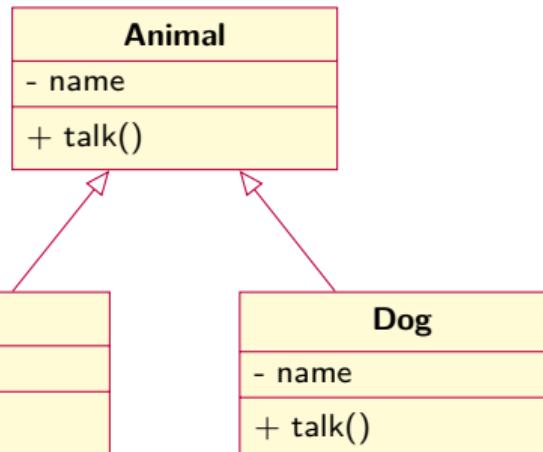




- 在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为

```
Animal 类 *animalCat = new Cat 类 ("爱丽丝");
Animal 类 *animalDog = new Dog 类 ("旺财");

cout << animalCat->name << ":" << animalCat->talk() << endl;
cout << animalDog->name << ":" << animalDog->talk() << endl;
```





多态性 (Polymorphism)

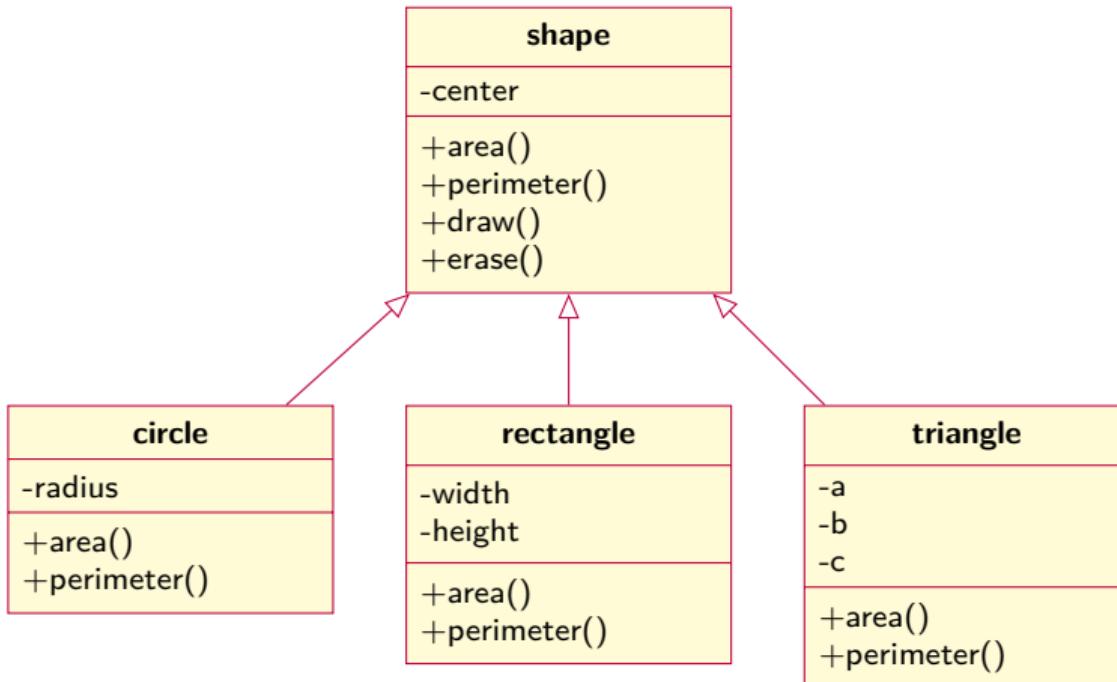
| 基本特性

OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

17

▶ 形状类



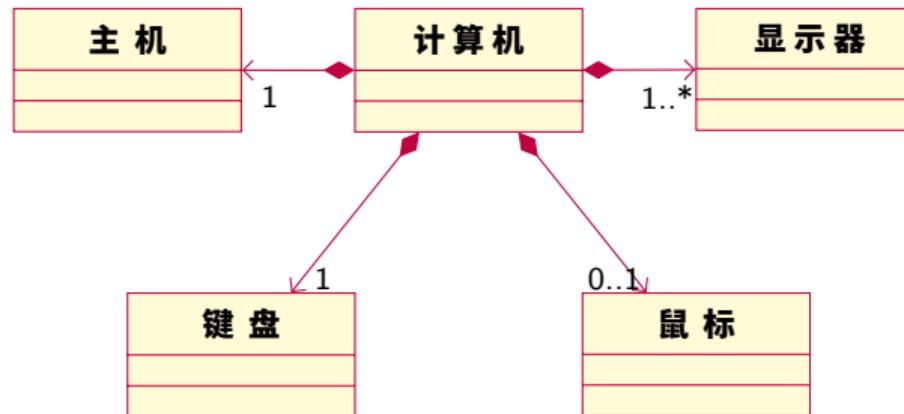
42





▶ 组合

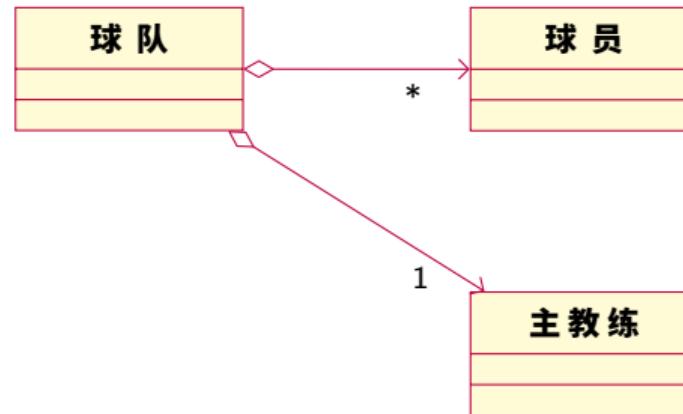
- ▶ 整体和部分关系
- ▶ 同时存在
- ▶ 同时销毁





▶ 聚合

- ▶ 包含关系
- ▶ 共享成员
- ▶ 组合的泛化

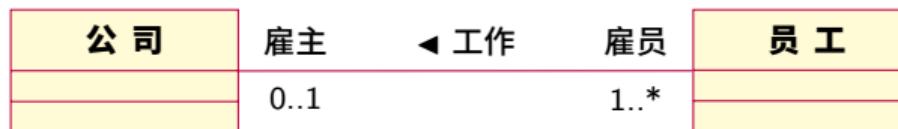




20

▶ 关联

- ▶ 消息传递链
- ▶ 聚合的泛化



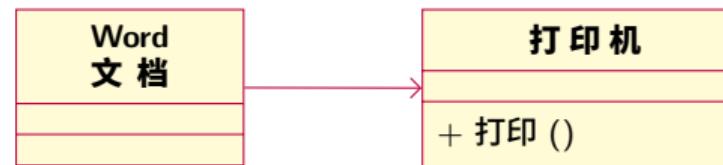
42





▶ 依赖

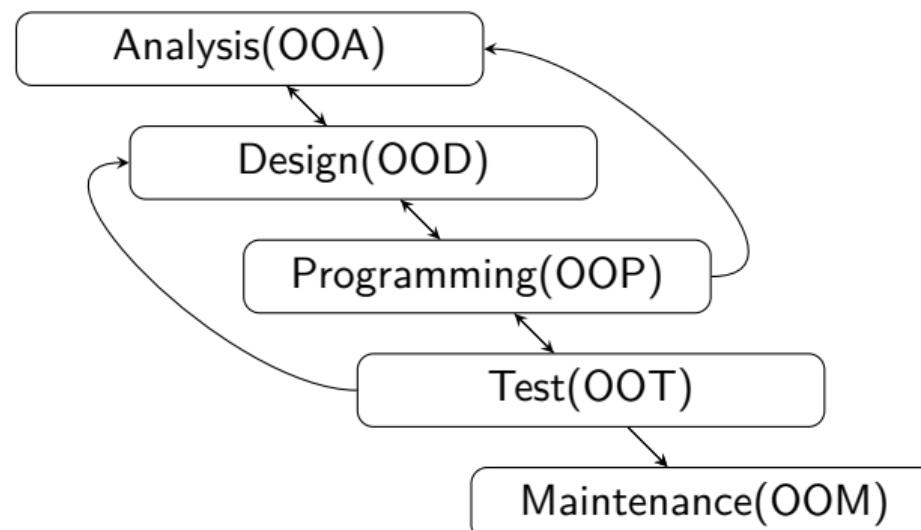
- ▶ 短时使用关系
- ▶ 非永久关系





▶ 使用**对象**去设计程序编写代码的一种方法

- ▶ 数据抽象
- ▶ 信息隐藏
- ▶ 代码重用





▶ 过程式程序设计

- ▶ 数据和过程分离，可维护性与可重用性差
- ▶ 不适于大型程序开发

▶ 面向对象程序设计

- ▶ 现实世界的直接模拟
- ▶ 可重用性好，可维护性好
- ▶ 适合大型程序开发





基本特性

是什么？

为什么？

发展史

目的

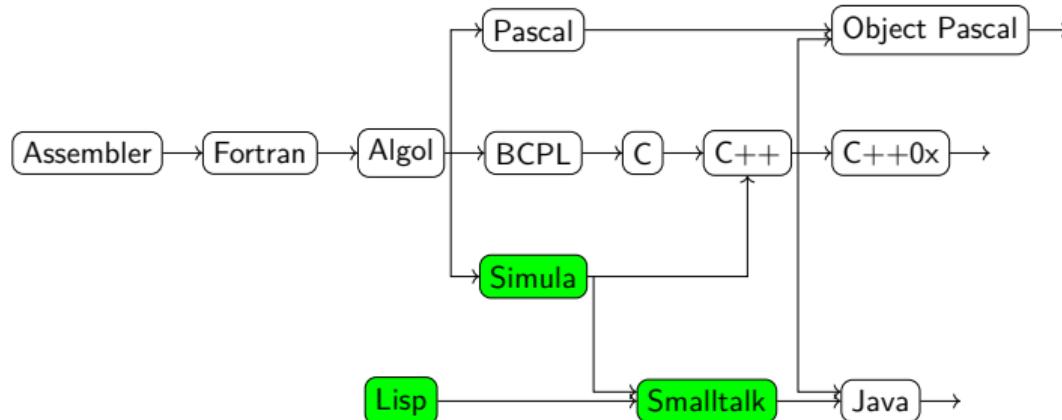
程序结构

基本程序

附件下载

24

▶ 族谱 (Family tree)





OBJECT
ORIENTED
PROGRAMMING—
OOP

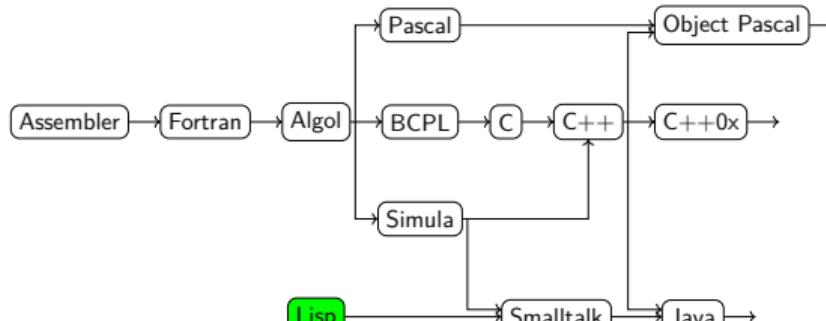
基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

25

► Lisp

- ▶ 1958
- ▶ 人工智能语言
- ▶ 引入了对象的概念 (Identified items with attributes)
- ▶ 语法特点

```
( let( (a 6) (b 4) (+ a b) ) )
```



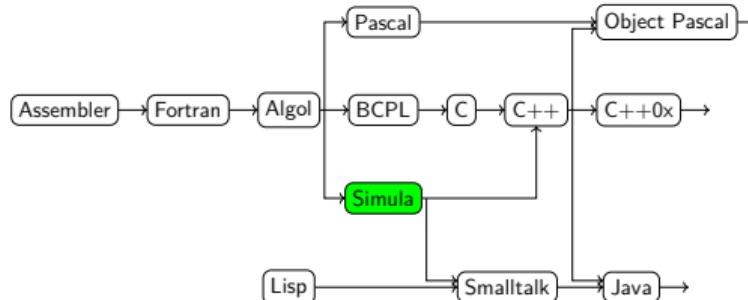
John McCarthy



▶ Simula

- ▶ 1967
- ▶ 被认为是第一个面向对象程序设计语言
- ▶ 第一次引入了对象、数据抽象和类的定义及继承机制
- ▶ 语法特点

```
Begin
  Class Char(c);
  Character c;
  Begin
    Procedure print;
    OutChar(c);
  End;
End;
```

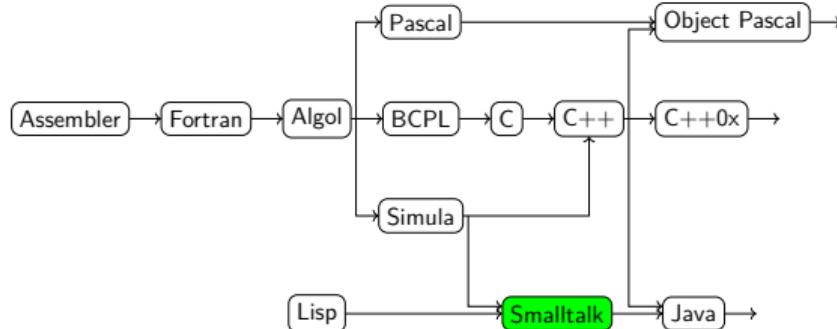


Ole-Johan Dahl and
Kristen Nygaard



► Smalltalk

- 1972
- 第一个纯 (Pure) 面向对象程序设计语言
- 语法特点
 - Primitives such as character and punctuation is treated as an object
例如：3+4
Sends the message “+” to the receiver 3 with argument 4



Alan Kay





C

发展史

OBJECT
ORIENTED
PROGRAMMING—
OOP基本特性
是什么？

为什么？

发展史

目的

程序结构

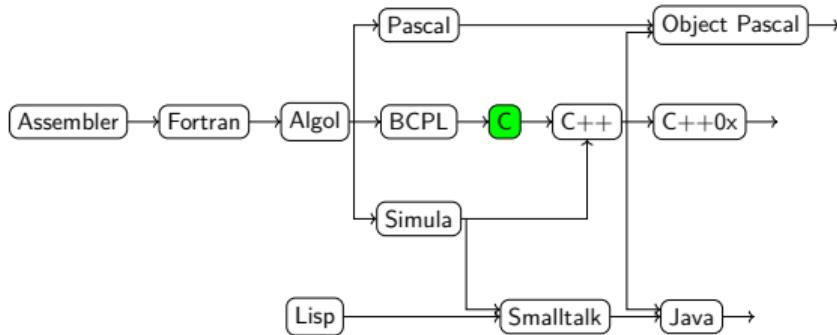
基本程序

附件下载

28

▶ C

- ▶ 1972
- ▶ C is quirky, flawed, and an enormous success. While accidents of history surely helped, it evidently satisfied a need for a system implementation language **efficient enough to displace assembly language**, yet **sufficiently abstract and fluent** to describe algorithms and interactions in a wide variety of environments.



Dennis Ritchie

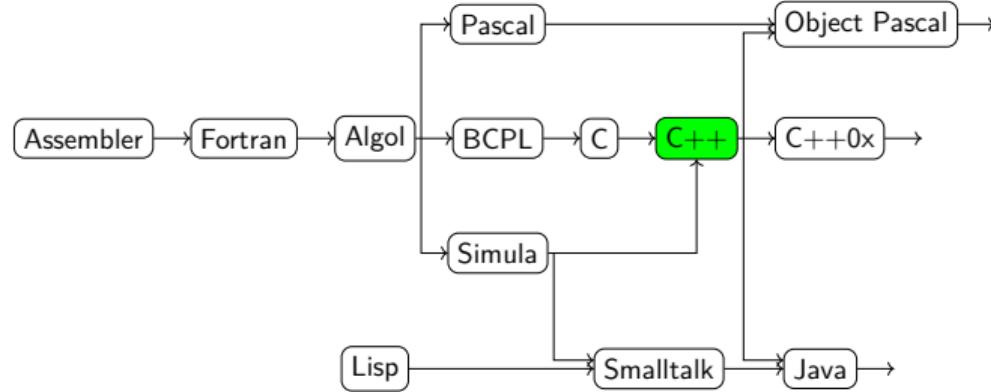




29

▶ C++

▶ 1985



Bjarne Stroustrup
(本贾尼·斯特劳斯特卢普)



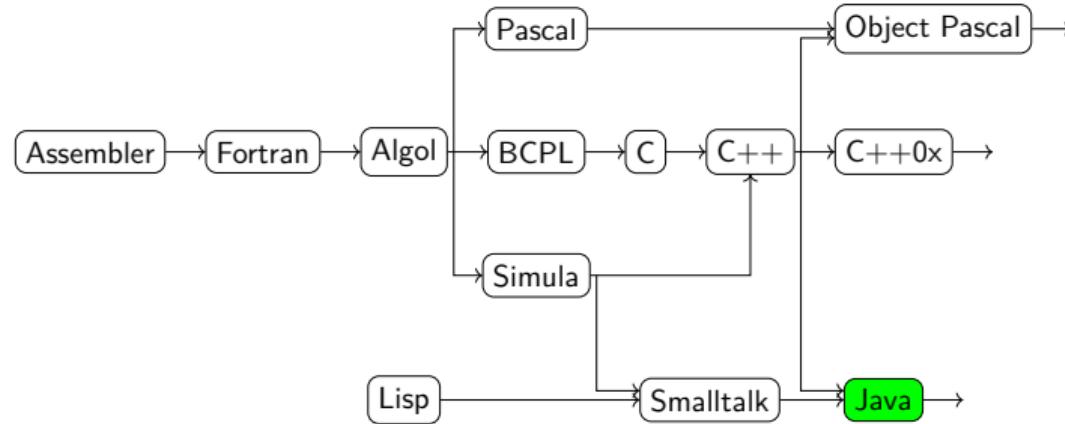


OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

30

► JAVA
► 1995



James Gosling





- ▶ Java
网络计算 (Network computing)
- ▶ C++
系统程序设计 (Systems programming)



Bjarne Stroustrup

<http://www2.research.att.com/~bs/>





- ▶ Adobe 系统: Photoshop, Illustrator 等
- ▶ Alias|Wavefront: Maya, Autodesk 等
- ▶ Apple OS X 重要的部分
- ▶ 游戏: 星际争霸, 暗黑破坏神, 魔兽争霸等
- ▶ Telecommunications
- ▶ Google (网络搜索引擎等)
- ▶ Microsoft applications and GUIs
- ▶ Linux tools and GUIs
- ▶ Amazon.com: 大型电子商务应用软件
- ▶



基本特性

是什么？

为什么？

发展史

目的

程序结构

基本程序

附件下载

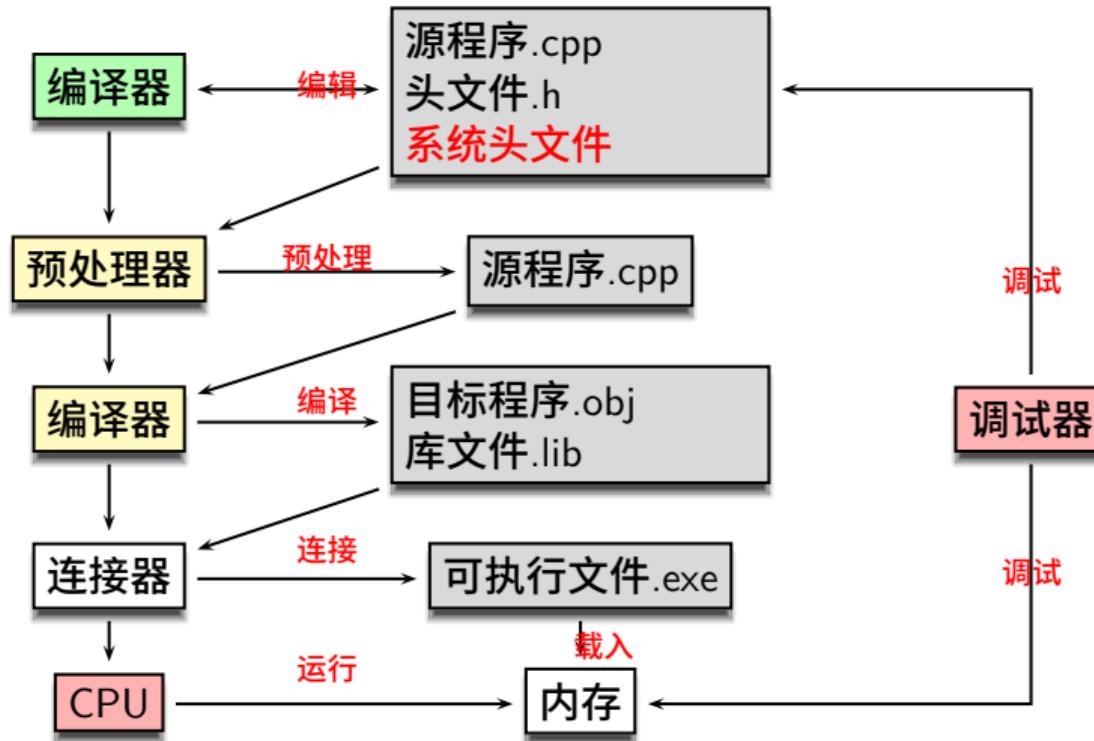
33

- ▶ Java、C# 和 Objective C
- ▶ 计算机图形学及计算机动画
- ▶ 图形图像处理
- ▶ 虚拟现实技术
- ▶





▶ 开发流程





- ▶ 辅助程序
- ▶ 工具
 - ▶ 编辑器
 - ▶ 编译器／解释器
 - ▶ 自动建立工具
 - ▶ 调试器
 - ▶ 版本控制器
 - ▶ GUI 设计器
 - ▶





▶ DEV C++

基本特性

是什么？

为什么？

发展史

目的

程序结构

基本程序

附件下载

36

The screenshot shows the Dev-C++ 4.9.9.2 IDE interface. The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar has icons for New, Insert, Toggle, and Goto. The Project tab is selected, showing main.cpp, rectangle.h, and rectangle.cpp. The code editor displays the following C++ code:

```
#include <cstdlib>
#include <iostream>
#include "rectangle.h"

using namespace std;

int main(int argc, char *argv[])
{
    double l, w;

    cout << "Please input the length and the width of a rectangle";
    cout << "Length=";
    cin >> l;
    cout << "Width=";
    cin >> w;

    cout << "The perimeter of the rectangle is " << GetPerimeter(l, w);
    cout << "The Area of the rectangle is " << GetArea(l, w) << endl;
```

<http://sourceforge.net/projects/orwelldevcpp/>



OBJECT
ORIENTED
PROGRAMMING—
OOP

▶ Visual Studio 2005

基本特性

是什么?

为什么?

发展史

目的

程序结构

基本程序

附件下载

37

The screenshot shows the Microsoft Visual Studio 2005 IDE interface. The title bar reads "VSTest - Microsoft Visual Studio". The menu bar includes File, Edit, View, Project, Build, Debug, Tools, Window, Help. The toolbar has icons for New, Open, Save, Cut, Copy, Paste, Find, Replace, and others. The Solution Explorer on the left shows a project named "VSTest" with files: stdafx.h, stdafx.cpp, VSTest.cpp, and ReadMe.txt. The Properties and Resource View tabs are also visible. The main code editor window displays the following C++ code:

```
// VSTest.cpp : 定义控制台应用程序的入口点。
//

#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Hello world!" << endl;

    return 0;
}
```

The status bar at the bottom shows "生成成功" (Build Succeeded), "行 12 列 1 Ch 2 Ins", and page numbers "42" and "43" on either side.

<http://www.visualstudio.com/>





OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性

是什么?

为什么?

发展史

目的

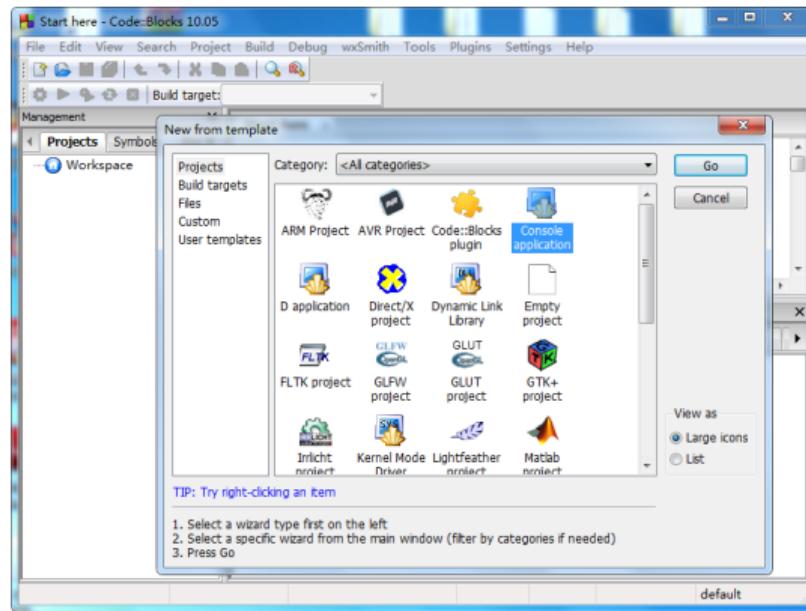
程序结构

基本程序

附件下载

38

▶ Code::Blocks



<http://www.codeblocks.org/>





OBJECT
ORIENTED
PROGRAMMING—
OOP

► QT

基本特性

是什么？

为什么？

发展史

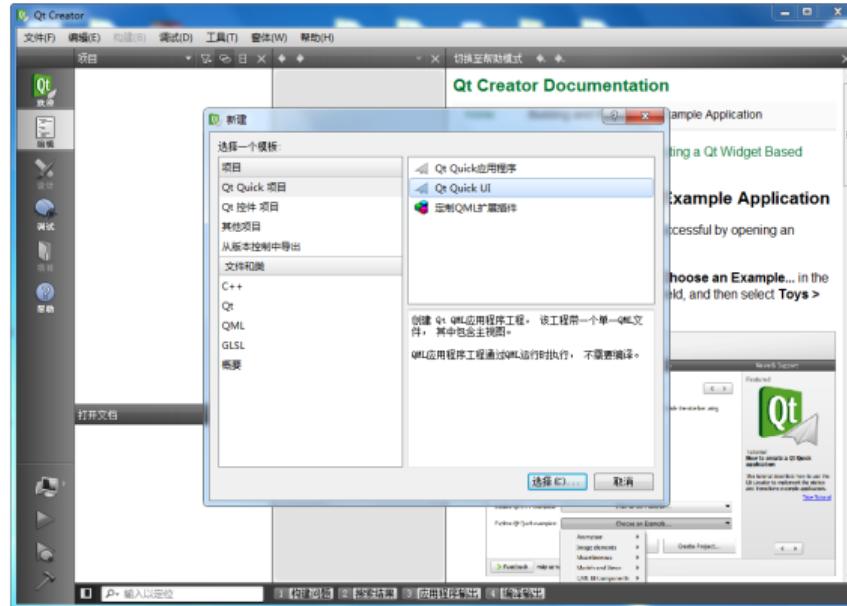
目的

程序结构

基本程序

附件下载

39



<http://qt-project.org/>





```
// 例 01-01: ex01-01.cpp
#include <iostream>
using namespace std;
int main()
{
    char strName[32];
    cin >> strName;
    cout << "Hello, " << strName << "!" << endl;
    return 0;
}
```

▶ 名字空间 (namespace)

```
using namespace std;
```

▶ 输入/输出

▶ cin 对象

```
cin >> 对象 1 >> 对象 2 >> ... >> 对象 n;
```

▶ cout 对象

```
cout << 对象 1 << 对象 2 << ... << 对象 n;
```





▶ 两个整数的加法

```
// 例 01-02: ex01-02.cpp
// 计算两个整数的和
#include <iostream> // 输入输出（流）
// main 函数是程序的入口
int main()
{
    // 变量声明
    int number1 = 0; // 第 1 个整数（初始化为 0）
    int number2 = 0; // 第 2 个整数（初始化为 0）
    int sum = 0; // 和（初始化为 0）

    std::cout << "Enter first integer: "; // 提示输入数据
    std::cin >> number1; // 读入数据到 number1

    std::cout << "Enter second integer: "; // 提示输入数据
    std::cin >> number2; // 读入数据到 number2

    sum = number1 + number2; // 加，并将结果存入 sum

    std::cout << "Sum is " << sum << std::endl; // 显示 sum; 并显示 end line
    return 0;
}
```





本讲附件

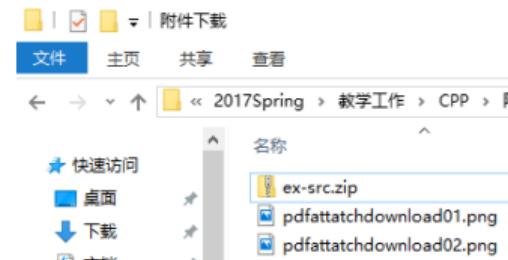
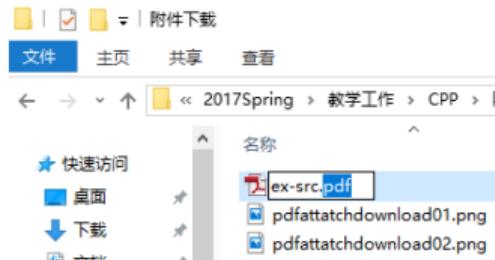
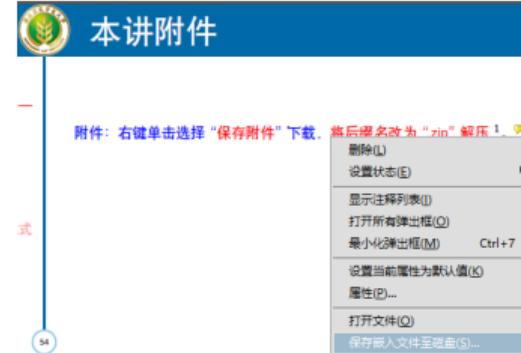
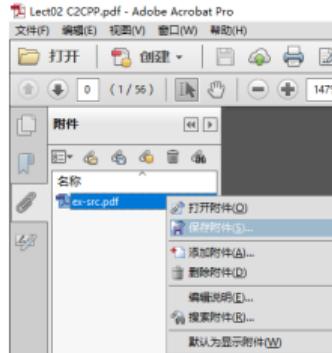
| 附件

OBJECT
ORIENTED
PROGRAMMING—
OOP

基本特性
是什么?
为什么?
发展史
目的
程序结构
基本程序
附件下载

42

附件：右键单击该链接，选择“保存附件”下载，将后缀名改为“.zip”解压^{1 2}。



¹请退出全屏模式后点击该链接。

²以 Adobe Acrobat Reader 为例。

本讲结束，谢谢！
欢迎多提宝贵意见和建议