

Program Design

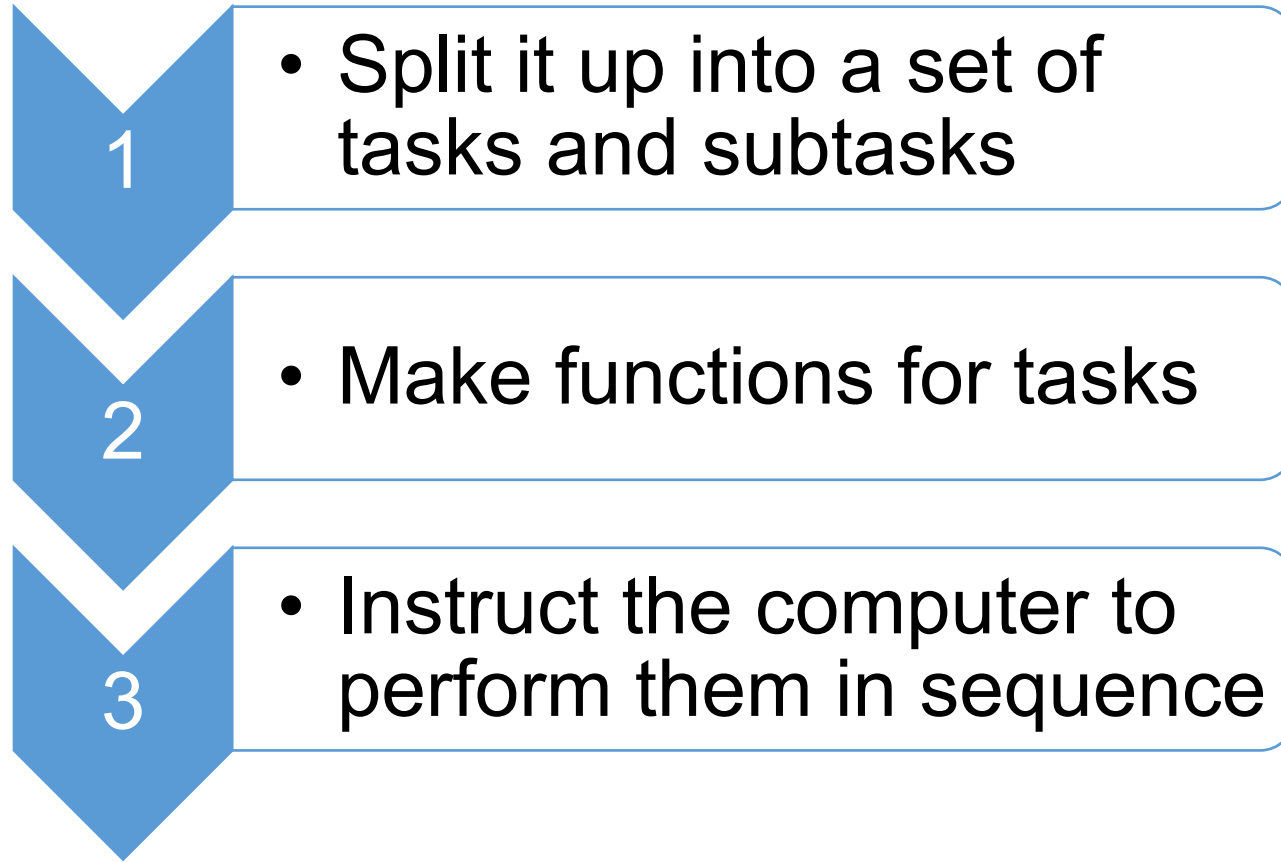
-- Object-oriented programming

Junjie Cao @ DLUT

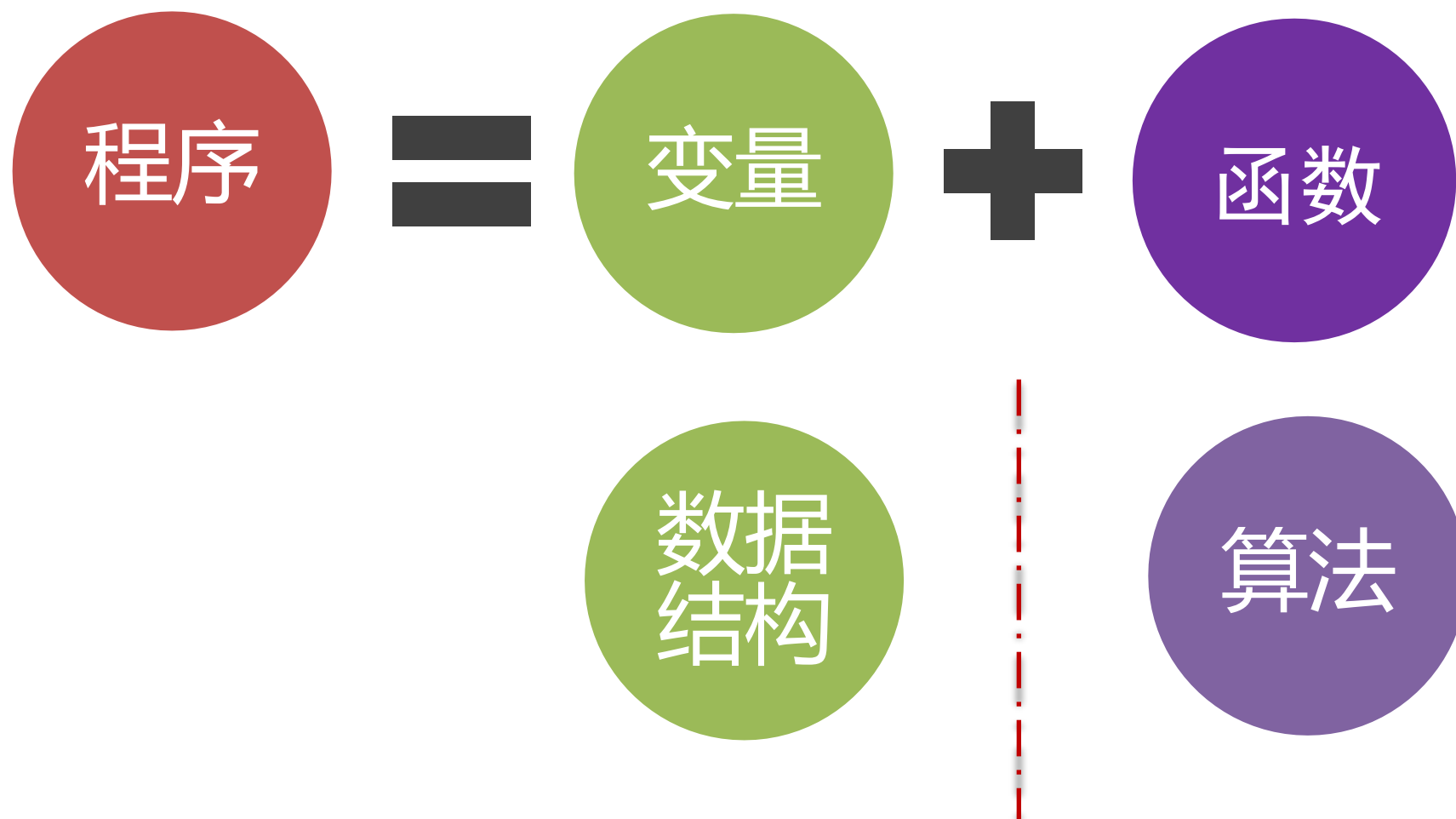
Summer 2022

<https://github.com/jjcao-school/c>

Procedural programming 结构化程序设计



Procedural programming



没有紧密/直观的联系

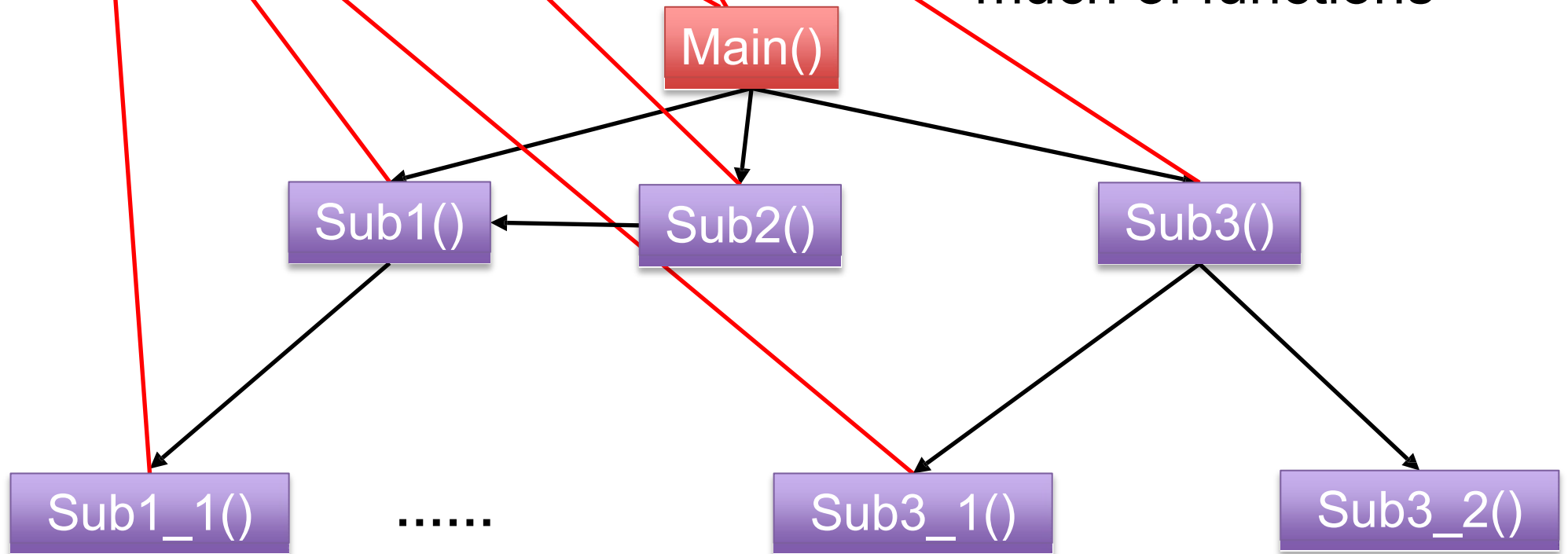
结构化程序设计的程序模式

变量:



- Sequence is obvious but hierarchy is very ambiguity.
- It is hard to organize too much of functions

函数:



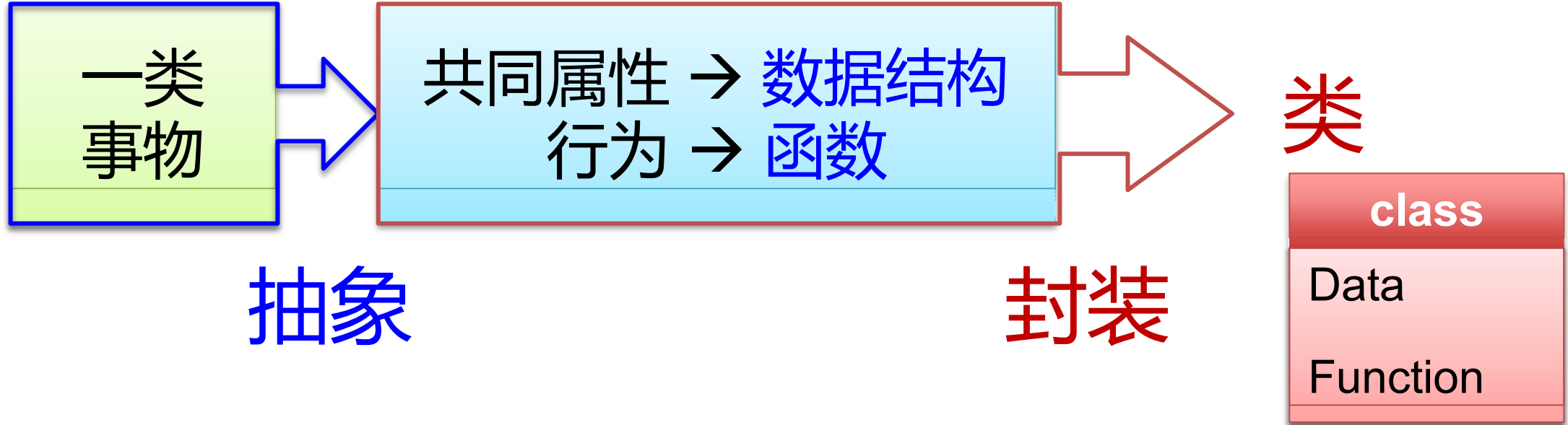
业界需要面向对象

- 软件设计的目标：



Procedural vs Object-Oriented programming

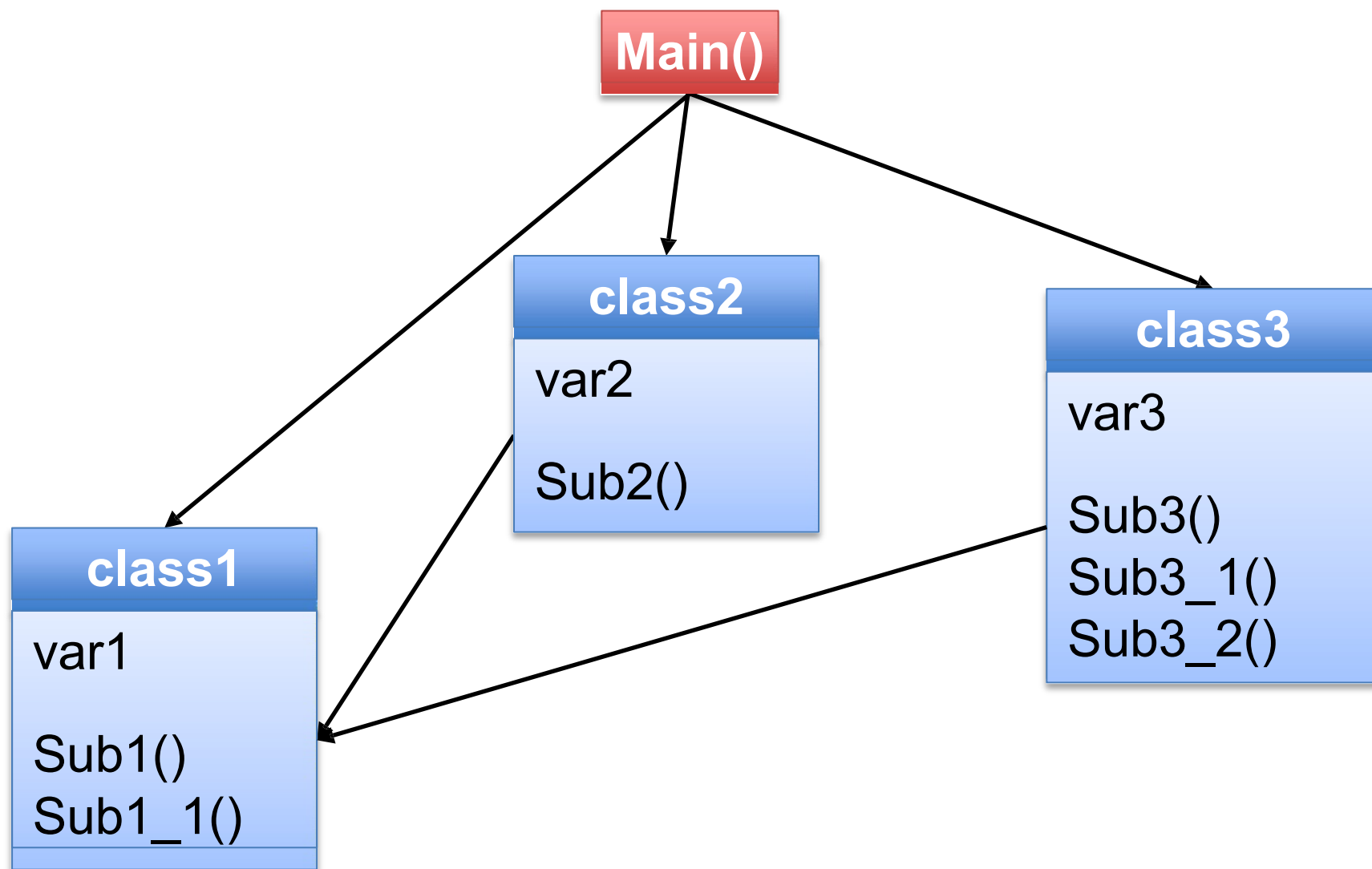
- People think of the world in terms of interacting **objects**
 - **Properties + behaviors** (they are inseparable)



Procedural vs Object-Oriented programming

- With procedural programming, the properties (data) and behaviors (functions) are **separated**
 - does not provide a very intuitive representation of reality
 - It's up to the programmer to manage and connect the properties to the behaviors in an appropriate manner.
- OOP provides us with the ability to design an “object”: tool to manage complexity when needed
 - easier to write and understand
 - a higher degree of code-reusability

面向对象的程序模式



面向对象的程序设计

- 如果你是Gru, 面对如此千姿百态的Minion, 怎么办?



面向对象的程序设计



Minion类

class Minion

Name;
Hair Color;
Eye Number;
Clothes Type;
.....

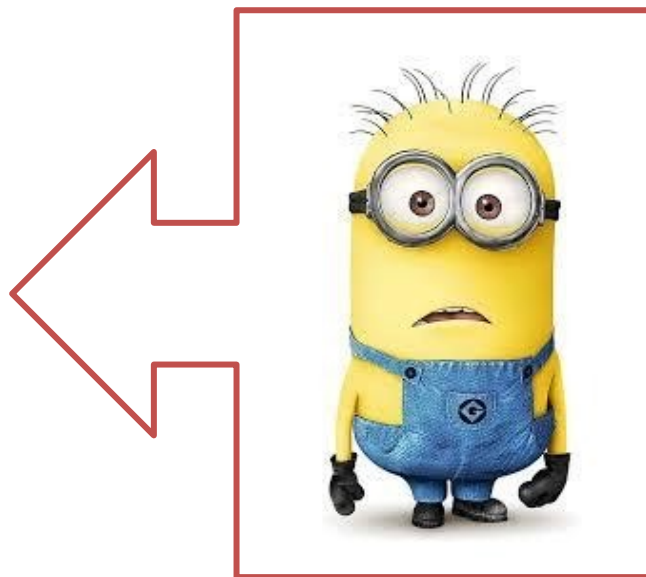
GetMinionName();
CountEyeNum();
SetHairColor();
SelectClothType();
.....

面向对象的程序设计



不同的Minion对象

对象/实例: object/instance



Minion类

```
class Minion
```

```
Name;  
Hair Color;  
Eye Number;  
Clothes Type;  
.....
```

```
GetMinionName();  
CountEyeNum();  
SetHairColor();  
SelectClothType();  
.....
```

类的定义

```
class 类名
{
    访问范围说明符:
        成员变量1
        成员变量2
        ...
        成员函数声明1
        成员函数声明2
    访问范围说明符:
        更多成员变量
        更多成员函数声明
        ...
};
```

结构化 **vs** 面向对象

- 都是分而治之
- 分解问题的出发点不同：功能 vs 对象
- 思维模式不同
- 结构化方法将数据结构和过程分开考虑，面向对象的方法组合数据和过程于对象之中。

Once you've been properly familiarized with OOP, you'll likely never want to go back to pure procedural programming again.

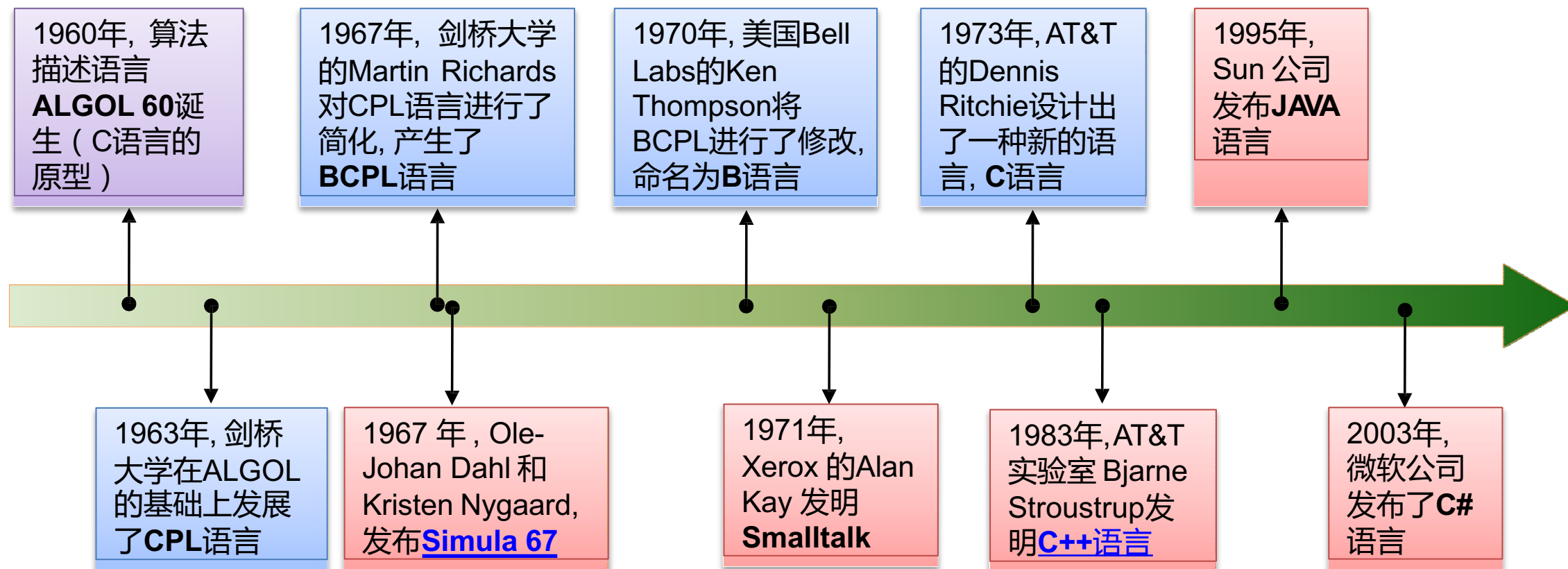
面向对象程序设计

语言的发展历程



早期程序设计语言的历史

■ C语言发展历程 ■ 面向对象程序设计语言的发展历程





C++语言的历史



- 1979年, AT&T实验室[Dr. Bjarne Stroustrup](#)开始发明
 - 开始的版本称为: **C with Classes**
- 1983年8月, 第一个C++实现投入使用
- 1983年10月, **正式命名为C++ (CPlusPlus)**
- 1985年10月, Bjarne完成了经典巨著[The C++ Programming Language](#) (第一版)

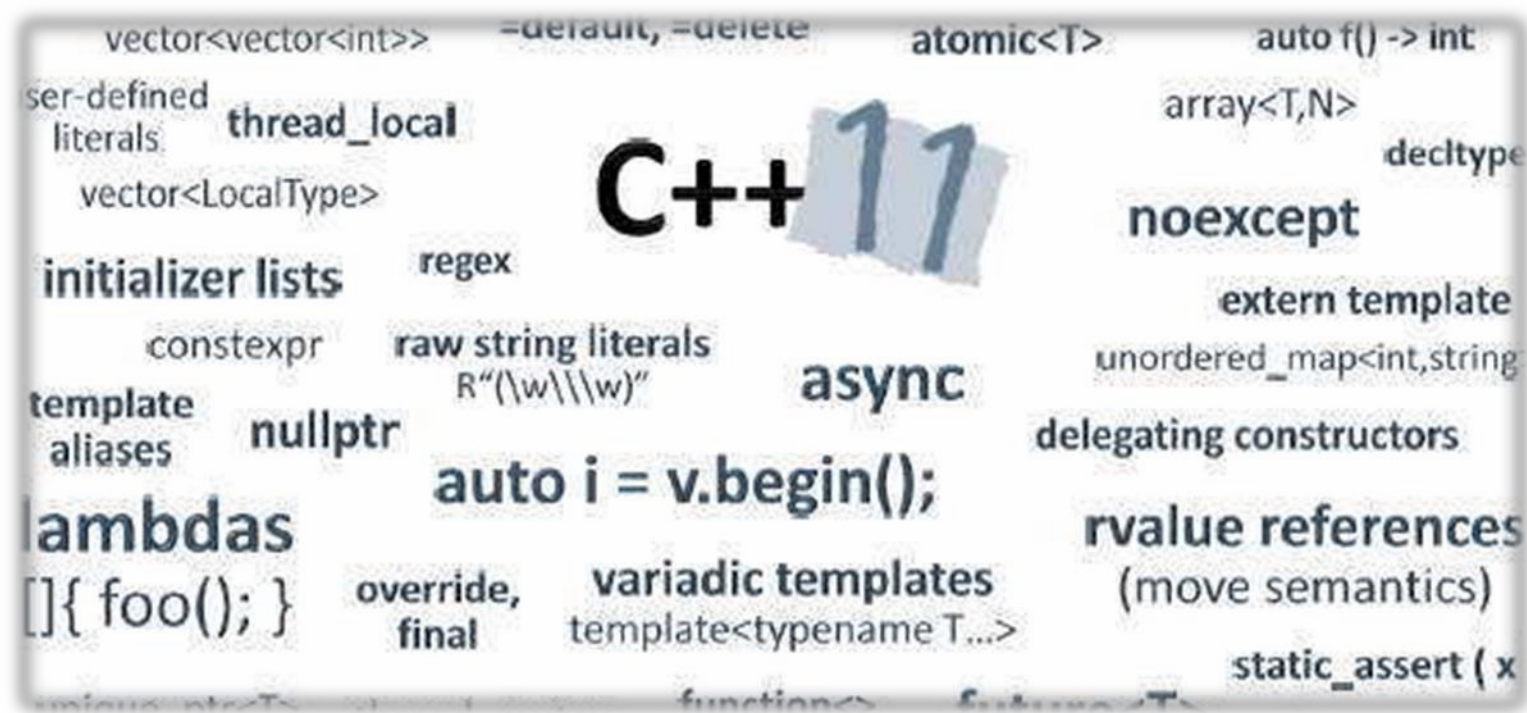


C++标准的发展

- ▲ 1989年, C++2.0版发布
- ▲ 1994年, ANSI C++标准发布
- ▲ 1998年, ANSI和ISO联合发布 “C++98” 标准
 - 加入STL (Standard Template Library) – 泛型设计
- ▲ 2003年, ISO的C++标准委员会发布 “C++03” 标准
- ▲ 2005年, “Library Technical Report 1” (TR1) 技术报告发布

C++11标准

- 2011年9月, ISO标准委员会通过了新的C++标准 -- C++11





常用的C++编译器

- ▲ GCC
- ▲ Visual C++ 10.0
- ▲ Dev C++
- ▲ Eclipse
- ▲ Borland C++Builder

第一个面向对象语言：Simula

Simula

1967年, Ole-Johan Dahl
和Kristen Nygaard

发布Simula 67

- 提出了类(class)和子类(subclass)的概念
- 程序设计语言发展史上的一个重要里程碑



C++之父

■ Bjarne Stroustrup

- <http://www2.research.att.com/~bs/>
- AT&T Fellow, AT&T Labs - Research
- Distinguished Professor
- Holder of the College of Engineering Chair
Computer Science at Texas A&M University
- C++ Bible
 - The C++ Programming Language (4th Edition)
 - Addison-Wesley ISBN 978-0321563842. May 2013.

