



# 类模板

苏州大学计算机科学与技术学院  
面向对象与C++程序设计课程组

# 模板函数

- ❖ 可以增强一个函数的通用性，减少函数的编程

∞ Typename

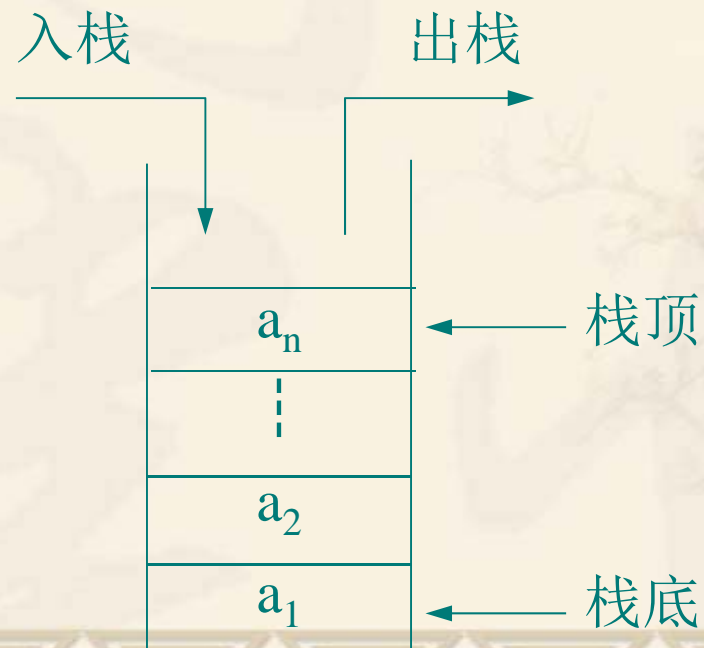
∞ Class: 更加通用

# 模板类

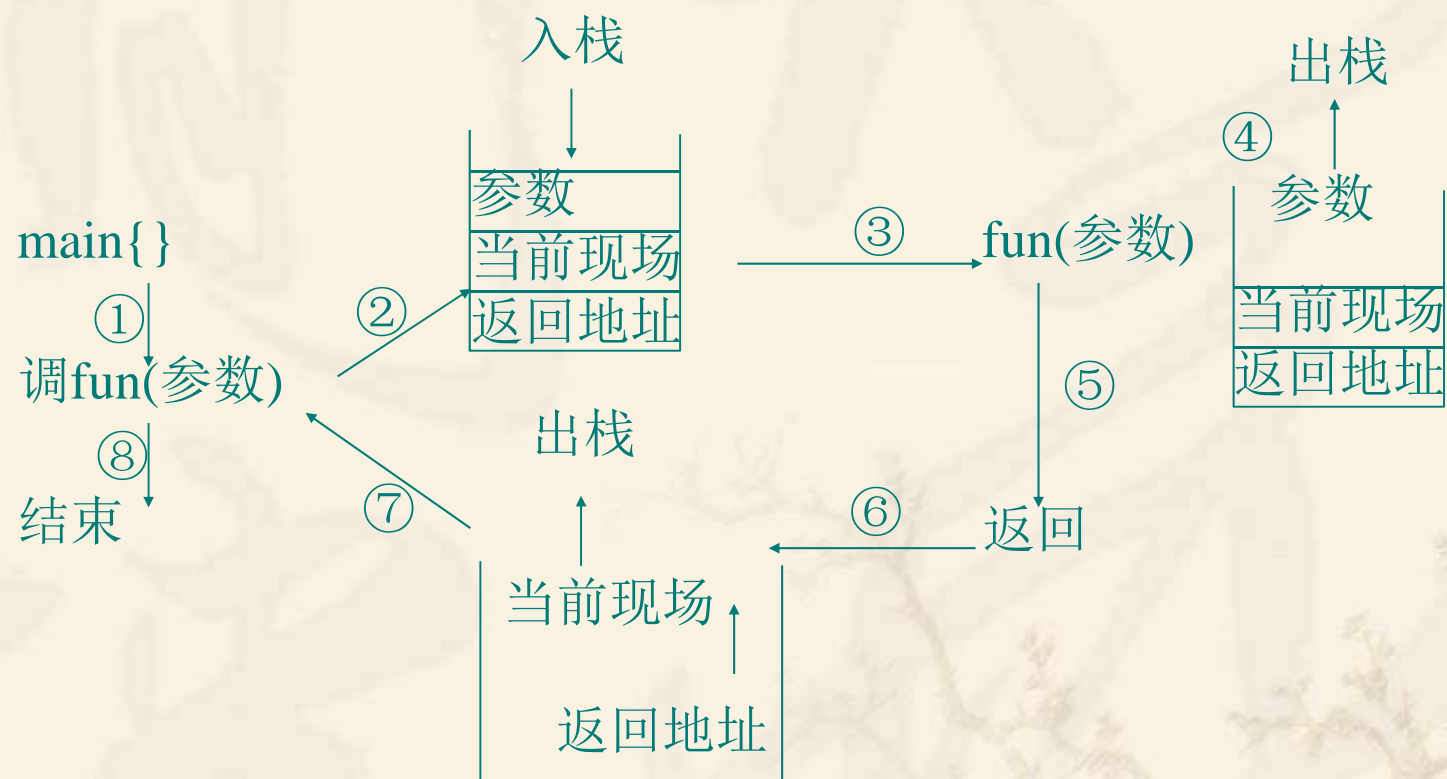
- ❖ 用模板技术同样可以消除相似的类
- ❖ 例子：
- ❖ 栈
  - ❧ 后进先出表
  - ❧ 一个字符堆栈类
  - ❧ 一个整型数据的堆栈类
  - ❧ 如何实现通用堆栈类？

# 特殊的线性群体——栈

栈是只能从一端访问的线性群体，可以访问的这一端称栈顶，另一端称栈底。

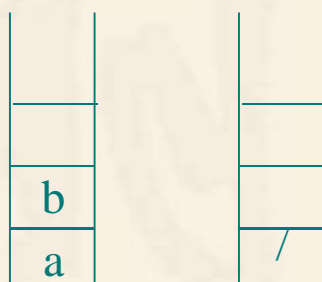


# 栈的应用举例——函数调用



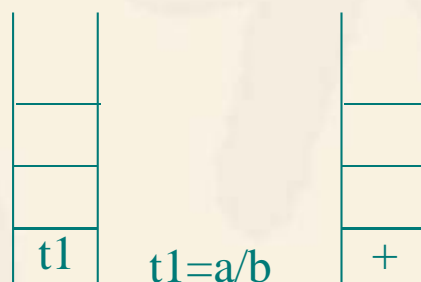


# 栈的应用举例——表达式处理



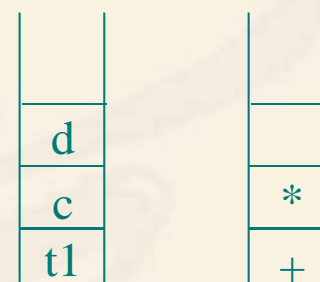
$a/b+c*d$

(a)



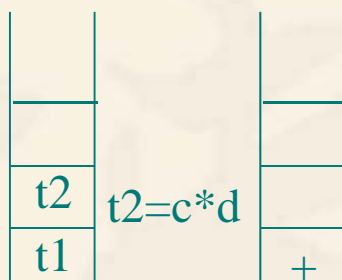
$a/b+c*d$

(b)



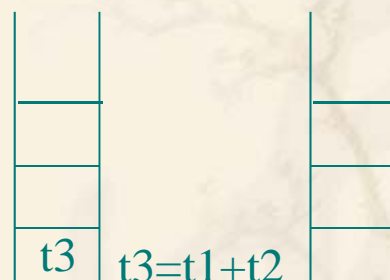
$a/b+c*d$

(c)



$a/b+c*d$

(d)



$a/b+c*d$

(e)

# 栈的基本状态

## ❖ 栈空

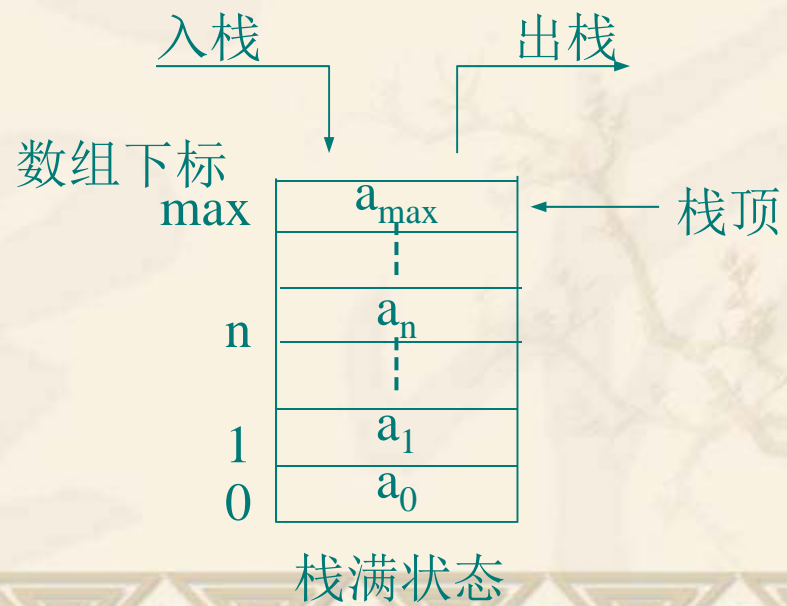
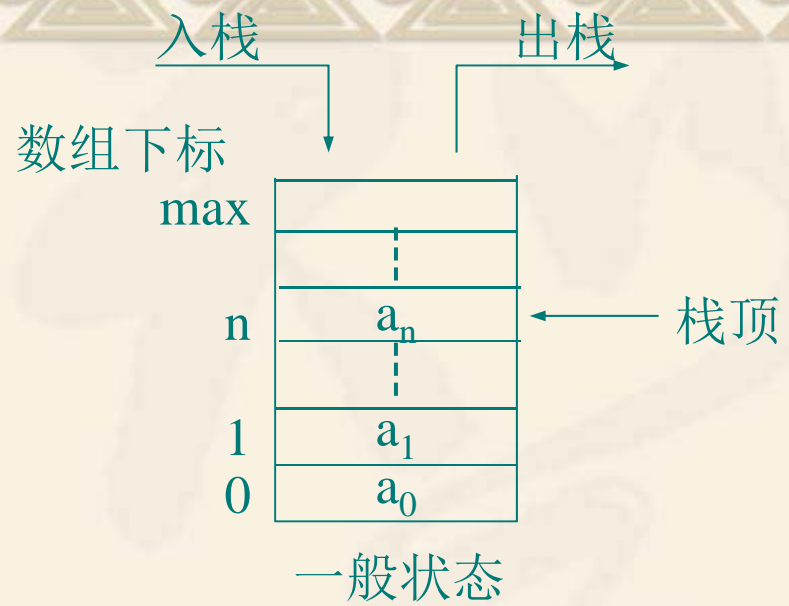
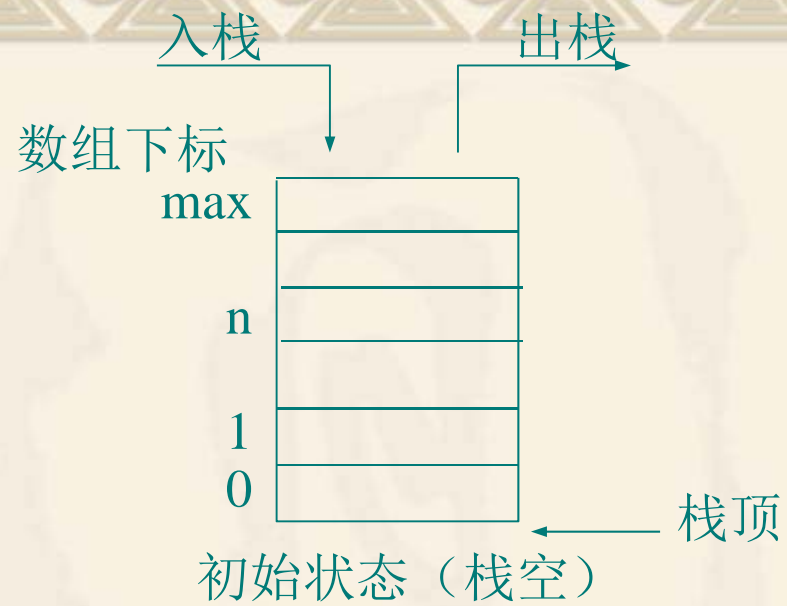
∞ 栈中没有元素

## ❖ 栈满

∞ 栈中元素个数达到上限

## ❖ 一般状态

∞ 栈中有元素，但未达到栈满状态





# 栈的基本操作

- ❖ 初始化
- ❖ 入栈
- ❖ 出栈
- ❖ 清空栈
- ❖ 访问栈顶元素
- ❖ 检测栈的状态（满、空）

## 例 栈类模板

//9-8.h

```
#ifndef STACK_CLASS
```

```
#define STACK_CLASS
```

```
#include <iostream.h>
```

```
#include <stdlib.h>
```

```
const int MaxStackSize = 50;
```

```
template <class T>
class Stack
{ private:
    T stacklist[MaxStackSize];
    int top;
public:
    Stack (void);
    void Push (const T& item);
    T Pop (void);
    void ClearStack(void);
    T Peek (void) const;
    int StackEmpty(void) const;
    int StackFull(void) const;
};
```

//类的实现略

# 标准模板库（STL）

- ❖ STL的目的是标准化组件，这样你就不用重新开发它们了。
- ❖ STL现在是C++的一部分，因此不用额外安装什麼。它被内建在你的编译器之内
- ❖ vector、list、map、set等