

C++ 高级程序设计-经验分享

熊丘桓 软件学院

eaglebear@smail.nju.edu.cn

2023 年 12 月 31 日





目录

分享目的

设计理念

设计的结果

三条主要脉络

不同人眼里的 C++

归纳的考试重点

例题



免责声明分享目的

- 咨询任课老师和助教以获取考试内容和考试题型。
- 分享个人学习经验和工程经验，供参考之用。



设计理念

- ① 效率
- ② 实用性优于艺术性严谨性
- ③ 允许一个有用的特征比防止各种错误使用更重要（相信程序员）

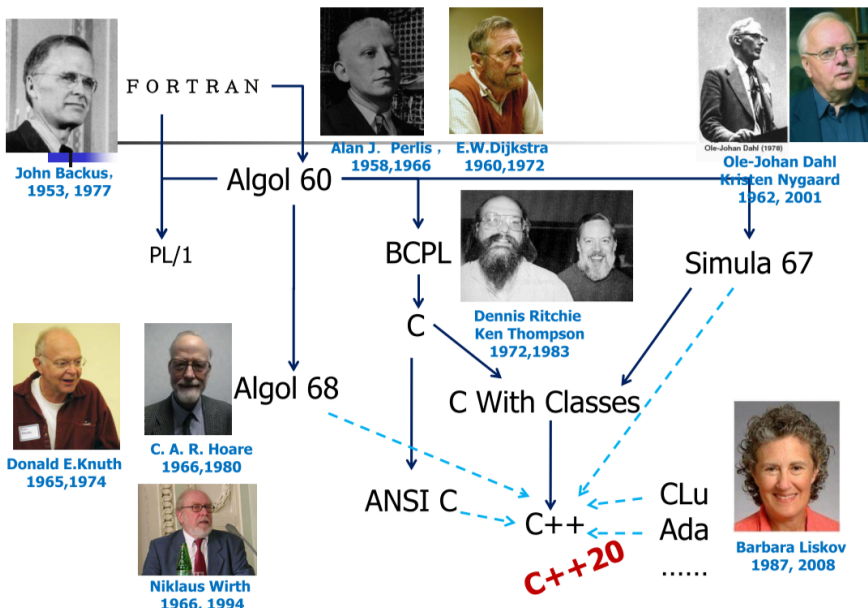


设计的结果

- 相比于 Java 并不优美的 OO 语法
- 相比于 C 增加了更反直觉且难以理解的执行逻辑
- 开发端最紧缺的岗位之一：C++ 工程师

但如果你理解了 C++ 的理念和目的，那么以上问题都能迎刃而解





三条主要脉络

- Algo60: 结构化编程, 拒绝意大利面条!
- BCPL & C: 系统编程, 程序员可以摸到 CPU 和内存
- Simula67: OO 编程, 梦开始的地方



不同人眼里的 C++

- 算法竞赛选手: $C++ = C + STL$
- Java 选手: $C++ = C + OO$
- 面经魔怔人: $C++ = \&...*\$ \# @$
- 软院选手: $C++ =$ 比 CPL 更简单的机试题, 面向往年卷复习
- 苏州校区选手: 往年卷? 什么往年卷?



我猜的根据软院往年卷归纳的考试重点

- ① 多态：重载（overload）与重写（override）
- ② 函数的运行机制：传值和传引用
C 语言：你猜我为什么不支持重载？
- ③ 宏：上古时期的奇技淫巧魔法
- ④ 常量指针和指针常量：CPL 笔试的漏网之鱼
- ⑤ C++ 为什么比 Java 难：五三原则、虚函数、多继承
- ⑥ 面向对象编程十大问题



恭喜你！你已经精通 C++ 啦
下面来做几道例题吧



```
1 // 例题 1: 函数重载
2 void bar(int i) { cout << "bar(1)" << endl; }
3 void bar(const char c) { cout << "bar(2)" << endl; }
4 void func(int a) { cout << "func(1)" << endl; }
5 void func(char c) { cout << "func(2)" << endl; }
6 void func(long long ll) { cout << "func(3)" << endl; }
7 void hum(int i, ...) { cout << "hum(1)" << endl; }
8 void hum(int i, int j) { cout << "hum(2)" << endl; }
9 int main() {
10     char c = 'A';
11     bar(c);
12     short s = 1;
13     func(s);
14     hum(12, 5);
15     hum(10, 12, 1);
16     system("pause");
17 }
```

```
1 // 例题 1: 函数重载
2 void bar(int i) { cout << "bar(1)" << endl; }
3 void bar(const char c) { cout << "bar(2)" << endl; }
4 void func(int a) { cout << "func(1)" << endl; }
5 void func(char c) { cout << "func(2)" << endl; }
6 void func(long long ll) { cout << "func(3)" << endl; }
7 void hum(int i, ...) { cout << "hum(1)" << endl; }
8 void hum(int i, int j) { cout << "hum(2)" << endl; }
9 int main() {
10     char c = 'A';
11     bar(c); // bar(2)
12     short s = 1;
13     func(s); // func(1)
14     hum(12, 5); // hum(2)
15     hum(10, 12, 1); // hum(1)
16     system("pause");
17 }
```

1 // 例题 2: 常量和指针

2 int main() {

3 const int c = 128;

4 int* q = const_cast<int*>(&c); // 强制类型转换

5 *q = 111; // 企图通过变量指针修改常量

6 cout << " c " << &c << c << endl;

7 cout << " q " << &q << q << endl;

8 cout << "*q " << q << *q << endl;

9 }

```
1 // 例题 2: 常量和指针
2 int main() {
3     const int c = 128;
4     int* q = const_cast<int*>(&c); // 强制类型转换
5     *q = 111; // 企图通过变量指针修改常量
6     cout << " c " << &c << c << endl;
7     // c 是符号常量, 在编译时符号常量已经变为 128, 相当于 define
8     // 被编译器当作: cout << " c " << &c << 128 << endl;
9     cout << " q " << &q << q << endl;
10    cout << "*q " << q << *q << endl;
11    //Name Addr Value
12    // c 0012FF74 128
13    // q 0012FF70 0012FF74
14    // *q 0012FF74 111
15    //对于同一个地址 0x0012FF74, 输出了不同的值
16 }
```

```
1 // 例题 3: 右值引用
2 class A {
3     int val;
4     void setVal(int v) {
5         val = v;
6     }
7 };
8
9 A getA() {
10     return A();
11 }
12
13 // 知道风险, 并且想要改变新对象, 就使用右值引用 &&
14 int main() {
15     int a = 1;
16     int &ra = a;
17     const A &cra = getA();
18     A &&aa = getA();
19     A &ab = getA();
20 }
```

```
1 // 例题 3: 右值引用
2 class A {
3     int val;
4     void setVal(int v) {
5         val = v;
6     }
7 };
8
9 A getA() {
10     return A();
11 }
12
13 // 知道风险, 并且想要改变新对象, 就使用右值引用 &&
14 int main() {
15     int a = 1;
16     int &ra = a; // OK, 非 const 引用绑定左值
17     const A &cra = getA(); // OK, const 引用绑定右值
18     A &&aa = getA(); // OK, 右值引用绑定右值
19     A &ab = getA(); // ERROR, 引用不能绑定右值
20 }
```



```
1 // 例题 4: const 的含义
2 const void show(const A* const this) const {
3     // const 分别是什么含义？
4 }
```

1 // 例题 4: `const` 的含义

2 `const void show(const A* const this) const {`

3 // 第一个 `const` 修饰指针, 表示 `this` 指针不可修改;

4 // 第二个 `const` 修饰 `this`, 表示 `this` 指向的对象不可修改;

5 // 函数签名当中的 `const` 相当于参数当中第二个 `const`

6 // 返回值当中的 `const`: 自己去试一试呢?

7 }

```
1 // 例题 5: 构造函数
2 class Computer {
3     private:
4         const string name;
5         Keyboard& keyboard;
6
7     public:
8         Computer() {
9             // 想一想, 这么写有什么问题?
10            name = "EagleBear2002's PC";
11            keyboard = new Keyboard();
12        }
13 }
```

```
1 // 例题 5: 构造函数
2 class Computer {
3 private:
4     const string name;
5     Keyboard& keyboard;
6
7 public:
8     Computer(): name("EagleBear2002's PC"), key(new Keyboard()) {
9         // 成员初始化表: 你猜我是干嘛的?
10    }
11 }
```

新年快乐，期末大吉！
让我看看是谁元旦还要复习考试啊

