C++基础

> 刚开始比较零散就是记录一下各种关键字的用法等等

- 头文件iostream ，包含在了std的命名空间中
- namespace 创建命名空间
- << 流插入 可以自动识别类型 >> 流提取
- cout 输出，cin 插入（使用时需要包含头文件）
- endl 就相当于c中的换行\n
- 缺省参数有全缺省和半缺省（必须从右往左缺省，必须连续缺省）
  - 函数的声明和定义不能同时出现缺省参数，要么在声明要么在定义，最好在声明中写。
- 函数重载

  1. ```
     void f();
     void f(int a=0);
     //这构成重载但是会报错（不知道调用哪个）
     ```

  2. 函数重载只与参数个数，参数类型和参数顺序有关
  3. 函数重载的本质就是函数名修饰规则与c不同
     - C语言不支持重载函数，因为编译的时候，两个重载函数的函数名相同，在func.o符号表中存在歧义和冲突
     - C++的目标文件符号表中不是直接用函数名来标识和查找函数：他引入了函数名修饰规则，不同的编译器规则不同
- auto 在C++中变成了自动推导类型，auto创建的变量必须初始化，而且auto自动推导的类型会丢失常属性
- auto不能作为形参，也不能定义数组
- 范围for
  - ```
    int array[]={1,2,3,4,5};
    //自动取数组里面的元素赋值给e
    //依次取，并且自动结束
    //也可以不用auto 也可以自己写清楚类型，但是auto更方便
    //范围for必须是数组名
    for(auto e : array)
    {
        cout<< e <<endl;
    }
    ```
  - ```
    //改变数组中的元素需要用到引用（指针不行）
    for(auto &e :array)
    {
        e++;
    }
    ```
- 指针空值nullptr（c++11）
  - 在C++98的时候NULL就相当于0，在一些极端情况下会出问题
    ```

```
void f(int)
{
    cout<<int<<endl;
}
void f(int*)
{
    cout<<int*<<endl;
}
int main()
{
    f(0);
    f(NULL);
}
//都会走第一个void f(int)
```

- C++11之后用nullptr来表示空指针