

类型

不仅定义了数据元素的内容，还定义了可以进行的**运算**

进行的运算是否合法，比如成员运算符：. \rightarrow

自定义类型时考虑要进行的运算

标准输入输出对象

1. `cin` 作为条件时，检测流的状态。（文件结束符及无效输入判断状态为无效）
2. `cout`
3. `cerr` 警告和错误消息，不缓冲
4. `clog` 一般性信息

I/O缓冲区

为什么要设置缓冲区？

减少 `read()` 和 `write()` 的调用次数

--> 系统调用，操作系统从用户态转换到内核态开销大

缓冲模式

1.全缓冲

填满缓冲区自动冲洗，或调用 `fflush()`

2.行缓冲

填满缓冲区自动冲洗，或调用 `fflush()`，或有换行符 `\n`

3.不带缓冲

写入什么立即输出什么， `stderr`

修改缓冲类型

1. `void setbuf(FILE *fp, char buf)`

关闭缓冲，将 `buf` 设置为 `NULL`

2. `int setvbuf(FILE *fp, char buf, int mode, size_t size)`

mode	缓冲方式
_IOFBF	全缓冲
_IOLBF	行缓冲
_IONBF	不带缓冲

