3.31.2025

C++进阶：

模板，创建一个通用函数，返回值类型和参数类型用虚拟类型替代

template<typename T> //声明创建模板

调用模板函数的方法有两种：

1. 自动类型推导
2. 显式指定类型

注意事项：

1. 自动类型推导，必须推导出一致的数据类型T才可以使用
2. 模板必须确定出T的数据类型才可以使用

示例：template.cpp

实现选择排序函数模板：Select\_sort\_template.cpp

普通函数与模板函数的区别：

1. 普通函数调用时会发生隐式数据类型转换
2. 函数模板调用时，如果使用自动类型推导，则不会发生隐式类型转换
3. 使用显示指定类型调用时，会发生隐式数据类型转换

普通函数与模板函数重载时，调用规则：

1. 如果普通函数和模板函数都可以调用，优先使用普通函数
2. 可以通过空模板参数列表，强制调用模板函数
3. 模板函数可以发生重载
4. 如果模板函数可以更好的匹配，优先调用模板函数

模板通用性不是万能的，遇到特定的数据类型，可以针对提供具体化模板

例：模板为

template<class T>

bool myCompare(T a,T b)

{省略号}

重载模板为为特定person类重载模板

template<> bool myCompare(Person a,Person b)