1. 函数对象（仿函数）
   1. 重载 () 所以函数的对象 使用（）像函数调用
   2. 是类 而不是普通的函数
   3. 内部记录状态
   4. 作为类型 与模板进行配合使用
2. 谓词
   1. 普通函数或者仿函数返回值 bool类型
   2. 一元 一个参数 二元 两个参数
   3. 一元 查找 大于20的数字 find\_if 返回迭代器
   4. 二元 排序
3. 内建函数对象
   1. 取反
   2. 加法
   3. 大于 greater<int>()
4. 适配器
   1. 函数适配器
   2. 0~9 加起始值 进行输出 用户提供起始值
   3. bind2nd 绑定
   4. 继承 binary\_function<参数类型1，参数类型2，返回值类型>
   5. const修饰 operator()
   6. 取反适配器
      1. not1 一元 找出小于5
      2. not2 二元 排序 not2( less<int>() ) 从大到小 相当于 greater<int>()
   7. 普通函数指针适配
      1. ptr\_fun
   8. 成员函数适配
      1. //如果容器存放的是对象指针， 那么用mem\_fun
      2. //如果容器中存放的是对象实体，那么用mem\_fun\_ref
5. 常用遍历算法
   1. for\_each 可有有返回值
   2. 可以绑定参数进行输出
   3. transform 将容器中的数据进行搬运到另一个容器中
   4. 注意：目标容器需要开辟空间
6. 常用查找算法
   1. find 按值查找 Person
   2. find\_if 按条件查找 Person\*
   3. adjacent\_find算法 查找相邻重复元素 返回第一个重复元素的迭代器位置
   4. binary\_search算法 二分查找法 必须容器是有序序列
   5. count 和count\_if
7. 常用排序算法
   1. merge算法 容器元素合并，并存储到另一容器中，两容器要有序，并且顺序一致
   2. sort 排序
   3. random\_shuffle 洗牌 自己提供随机种子
   4. reverse反转
8. 常用的拷贝和替换算法
   1. copy复制
   2. replace replace\_if 替换
   3. swap 交换
9. 常用算数生成算法
   1. 头文件 numeric
   2. accumulate 累加
   3. fill 填充
10. 常用集合算法
    1. 交集 set\_intersection
    2. 并集 set\_union
    3. 差集 set\_difference