

ICT2 83 题目5（不在本实验室评估，但**必须完成并提交**）。

目标：

- **作业1的准备** - 在尝试作业前**必须完成练习/问题1和5**。这两个问题是作业1的一部分。对这些问题的评估在作业中。**问题5需要提交**，以便在作业开始时**获得完整的作业规范**。
- 要学习
 - OO设计，并体会到正确做这个设计的价值
 - 凝聚力和耦合性的实际意义
 - 来做重构和代码维护
 - 做测试
 - 注重细节

先决条件：

实验4必须完成

不落下这些练习是非常重要的。

你应该注意，即使一项练习没有被评估，但不尝试该练习将使你很难理解后续的材料。

如果你想在自己的电脑上工作，先安装graphviz，然后安装doxygen。

阅读清单是为专题5准备的。其中一些章节是以前课题的重复阅读内容。见"[阅读列表 ICT283](#)"。**教科书中关于"重载和模板"的章节特别重要**--是作业所需。

锻炼身体

在你弄清楚所需的确切内容之前，**不要开始编码**。在纸上完成。画一个UML图，说明类是如何连接和使用的。你需要先阅读下面的所有问题（子任务），以弄清楚需要什么，然后计划解决每个子任务的最佳方法。你可能需要多次阅读规范，以了解真正需要的是什么。确保你使用doxygen风格的注释来记录所有代码。

想一想你将用来证明你的程序工作的测试数据。你可以假设输入的数据文件包含正确格式的数据。这意味着你必须正确创建数据文件。

下面的**练习/问题1**要求你按照提供的规范进行练习。当你完成第1题时，你将对C++类的工作原理有所了解，并准备好进行自己的设计和实现。

从练习/问题2开始，你必须进行自己的设计。因此，反思你到目前为止所做的所有工作，重新设计并重新实现所有在之前练习中创建的类。**目的是创建可以在许多情况下重复使用的类，而不需要对类进行重新编码。在你的设计中力求高内聚性和低耦合性。**

仅仅让代码工作是不够的，还需要通过。设计是非常重要的。

1. 在之前的实验练习中，注册类有一个原始的结果数组。这个数组本身并没有被封装在一个类中，以提供对数组的控制访问。

对于这个练习和后面的练习，需要创建一个数据结构来封装一个原始数组。为了我们的目的，我们将这个数据结构称为**Vector**。**Vector**是一个模板化的数据结构，它封装了一个原始**动态数组**（使用 "new" 在堆上创建内存¹）。

由于**Vector**是模板化的，它可以存储任何东西，包括它自己。

最接近的类似物是Java *ArrayList*，它也是模板化的。你会在先修单元中使用*ArrayList*。原始数组是**模板类****Vector**的一个私有成员。你编写**Vector类**，并通过**Vector**的公共方法提供对动态原始数组的控制访问。你的**Vector**类与STL向量不一样，所以请不要试图复制STL向量。你会得到一个臃肿的结果，你的工作会受到惩罚。**Vector**类需要**最小但完整**。

不要试图模仿STL向量（`std::vector`）。方法或操作符不应该有重复的功能。见**规则33 倾向于最小的类而不是**

单元参考书**中的单体类**

: *C++ 编码标准：101条规则*、

准则和最佳实践》，作者是Sutter和Alexandrescu。电子书在图书馆，如果以下链接不起作用。

<http://lib.myilibrary.com.libproxy.murdoch.edu.au/Open.aspx?id=291948>（可能需要登录）

不要使用STL向量类作为你的**Vector**的服务数据结构（即你不能有`#include`

`<向量>`在本实验或作业1的任何部分的代码中）。

你要写的**模板Vector**类是一个线性数据结构的例子。整个（界面/声明和实现）模板**Vector**类被写在一个叫做**Vector.h**的文件中，因为它是一个模板类。教科书中解释了为模板类提供.cxx文件的方案。在这个实验（和作业）中，模板**Vector**类**没有****Vector.cxx**。没有必要通过为模板类提供一个.cxx文件来使问题复杂化。

尽管只有一个文件**Vector.h**存在，你必须把**Vector**的接口和它的实现分开，这是绝对必要的。**不进行这种分离将导致作业和考试中的分数为0。**

Vector.h包含以下内容：

```
#ifndef VECTOR_H
#define VECTOR_H
```

```
// 类的接口/声明必须有doxygen注释--把这些放在
// 遵循modelfile.h中的风格
```

模板<类T>

```
class Vector/ 这不是std::vector。
{
```

公众：

```
// 类的声明
```

```
// 你填入其余部分，并包括doxygen注释
```

私下里：

```
// 封装的动态数组
```

```
// 填写.....。
```

```
};
```

```
// 类的实现，后面有正常的注释
```

```
// 剩下的由你来填
```

¹ 随着内存在堆上的创建，一些方法成为强制性的。这些方法之一是析构器。见Lect- 13.ppt

² 见主题1。声明/接口与实现的分离是强制性的

#endif

向量要有多大？一旦你用 "new" 在堆上创建了内部数组，其大小就是固定的。如果有更多的数据要添加，会发生什么？Vector 一旦满了，就需要调整自己的大小（增长）。Vector 的用户不应该被这个细节所困扰，因为 Vector 会在幕后处理这个问题。当然，作为 Vector 的设计者，你必须编写代码来完成调整大小的工作。调整大小的数量通常是当前大小的 1.5 到 2 倍。

所需的 Vector 类是一个容器类。它必须不违反德~~克萨斯~~基百科有一个可以理解的解释。在实验 4 中，你是否遵守或违反了这一定律？

在使用这个 Vector 之前，你需要对它进行单元测试。与你的导师讨论你的想法。
任务的模板 Vector 类。

你将需要这个

在前面的实验中，你使用的是一个 Result 对象的数组。假设你在 Registration.h 中有以下内容：

结果 [MaxResults]; // 你可能已经使用了不同的名称来表示数组和最大尺寸的
数组，所以要相应地修改这个例子。

将上述内容改为：

```
Vector<Result> results (MaxResults); // 为 Vector 对象使用与你的实际数组名称相同的名称。
```

你需要在 Registration.h 中包含 "Vector.h"。建立你的 Registration 程序。如果你没有在 Vector 中提供所需的操作符[]方法，请提供这些方法。

在做了上述改变之后，是否需要改变其他的代码？是否应该有必要改变其他的东西？

你是否让你的 Vector 类依赖于应用程序的任何部分？注册数据的任何部分是 Vector 的成员数据的一部分吗？如果是，Vector 就不能正确地用于本实验的第 5 题和作业：你的 Vector 做得不对。

Vector 必须不依赖于应用程序的任何部分。Vector 只是一个封装好的数组，可以存储任何其他的数据类型，包括 Vector 本身。为了能够拥有这种灵活性，Vector 必须不知道它所存储的数据类型。

你的 Vector 类可以用在你提交的 Lab 4 的练习 7 中，以进入这个主题吗？在你继续之前，你需要工作实验室 4 练习 7 的代码。你在实验室 4 的练习 7 中的声明应该是：

```
float *dataArray ;
```

在实验 4 的练习 7 中，将上面的一行改为以下内容：

```
Vector<float> dataArray (N); // 你需要在练习 7 的文件顶部 #include "Vector.h"，以便能够使用 Vector 类。  
N 是你在原练习 7 中使用 new 创建动态数组时使用的值。
```

该程序。程序的建立和运行是否正确？需要修改吗？如果是的话，重新设计 Vector。在注册实验练习中再次用结果向量检查向量。

2. 考虑到前几周Registration实验工作中的要求变化：每个单位都有一个单位协调员，单位协调员的详细信息包括：协调员的姓名、房间号、电话号码和电子邮件。

思考一下你需要做的重新设计，以满足注册代码库的新要求。你将如何“面向未来”³你的设计，以便你编写的任何新代码将引入额外的功能来满足新的需求，而不是不断地修改现有的代码？(开放-封闭原则)注意，在过去，你得到了一个程序应该做什么输出的例子。如果输出格式被改变了呢？你是否要回去修改现有的类？

理想情况下，改变输出格式应该只需要改变“主程序”（客户端代码），而**不**需要改变数据类。那么，输入格式的改变呢？例如，你将如何满足文件末尾所示的输入格式？

其余的ict283。

你将使用第一个版本的输入格式，用于

你可以自由地重新设计注册代码库中的所有类，试图使它们适应未来。以后，当你发现如果你没有很好的设计和谨慎的实施所选择的设计，你必须做大量的重新设计和编码来满足扩展/变化的需求时，你将从你的成功和错误中学习。

在设计上花很少的时间或不花时间，而花大量的时间试图让代码变得更更有价值，这是一个常见的错误。“工作”。例如，作业2利用了先前的工作。如果先前的工作没有设计（和实施）对，任务2需要更多的工作（和时间）来完成，当你有非常有限的可用的时间。不考虑设计问题会导致不得不在两周的时间内完成所有的作业2。这将是很有压力的。

你应该使用UML进行设计。

对于这个特定的练习/问题，你是否必须改变问题1中的Vector类，以便它能在问题2中的新设计中工作？如果你这样做了，那么你在问题1中设计的Vector类是错误的。

3. 实现你对注册代码库的新设计。如问题2所示，新的设计有与Unit相关的额外信息。展示你的设计是多么灵活，以便通过编写不同的主程序来满足不同的输出格式。注意，仅仅通过这种方式获得不同的输出格式并不足以说明是好的设计--但这只是一个开始。

请参阅本问题清单末尾的输入格式样本。本问题使用第一个版本的输入格式（CSV）。

4. 在你的代码库上运行doxygen，检查输出结果，看你的初始设计是否与doxygen中显示的设计相符。
5. 这个练习是作业1的一部分。你需要先完成这个练习，然后再尝试任务1的其他部分。要进入作业1，需要在LMS中提交这个练习。

使用文本编辑器Notepad++（<https://notepad-plus-plus.org/>）来检查数据文件data/MetData-31-3.csv。这个数据是从<http://www.met.murdoch.edu.au/>。这个在线气象站最初是由我们默多克大学计算机科学毕业班的一个项目组创建的。你可以查看该网站，如果现任政府在网上有。

用Microsoft Excel这样的电子表格打开同一个文件（data/MetData-31-3.csv）。比较Excel和notepad++中显示的视图。这就是将用于作业1的数据类型。Notepad++中的

³ 可能不可能得到一个完美的设计来满足所有未来的要求。你可能希望的是尽量减少对现有代码的修改，以满足新的/变化的要求。

要看你的程序会看到什么：在notepad++中，选择查看菜单/显示符号/显示所有字符。

第一行显示传感器代码。其含义见文件SensorCodes.rtf。

第一个字段同时包含了日期和时间。你将需要把日期和时间值分开。日期可以存储在之前练习中的Date类中。你需要创建并单元测试一个Time类来存储时间值。

对于这个问题，我们只对第一个标有"**WAST**"的字段和标有"**S**"的字段感兴趣。S字段中的数据是风速。你应该假定数据文件中记录的速度是以m/s为单位的。你可以下载你自己的数据文件来检查单位。单位显示在你下载的原始数据文件中。

对于实验5，你将使用以下数据结构：

```
// #包括其他你需要的东西
#include "Date.h"// 你在以前的实验中的Date类
#include "Time.h"// 你在本实验室的时间类
#include "Vector.h"// 你的模板Vector类来自本实验室

typedef struct {
    Date d;
    时间t; 浮动速
    度;
}WindLogType;

Vector<WindLogType> windlog; // Vector<WindlogType>是一个实现的向量。
// 这不能被声明为全局性的，见讲义。UML图
```

会是什么样子的？

*Date类已经创建，并在实验室4中进行了单元测试。时间类和矢量类已经创建，并在第5实验室（本实验室）中进行了单元测试。请注意，当你创建类时，这些类会在以后的其他环境中被重复使用，所以要确保你的设计是为了重复使用。记住要保持类的简单。你对类的添加越多，超出了最低限度，它们的可重用性就会下降。应用低耦合和高内聚的原则。参见规则33在本单元的参考书中，**优先选择最小的类而不是单一类***

: C++

编码标准：101条规则、指南和最佳实践》，作者Alexandrescu等人，电子书在图书馆中，如果以下链接不起作用，

<http://lib.myilibrary.com/libproxy.murdoch.edu.au/Open.aspx?id=291948>。

编写一个C++程序，从文件data/MetData-31-3.csv中读取数据到windlog。

一旦数据被读入windlog，你的程序将使用windlog计算出平均风速（公里/小时）和样本标准偏差（SD）。这些数字会被打印到屏幕上的新一行，并有适当的文字信息来显示这些数字所代表的内容。

该程序将通过 *风记录*，当发现一个速度值与打印出来的平均速度相匹配时，这个速度的日期和时间（在 *风记录* 中）将被打印在一个新的行中。

一些需要考虑的事情：

- 是否有可能平均速度不会出现在 *风控记录* 中？
- 当你需要 "匹配" 而被匹配的值是浮点值时，这是什么意思？
- 如果一个传感器处于离线状态，数据文件中会有什么值？如果，比如说，一个离线的传感器有一个 N/A（或一些其他）的值，你的程序会发生什么。
- 任何离线的传感器值都应该参加平均数和 SD 的计算吗？

一旦你有了正确的结果，在更大的数据文件上检查你的程序，`data/MetData_Mar01-2014- Mar01-2015-ALL.csv`。这个数据文件被用于作业1。

练习/问题5和作业1需要一次只处理1个数据文件。

虽然你的程序只需要处理一个文件，但我们的程序需要处理不同的列排列/顺序的数据。列/字段名称没有变化，如WAST、S、T等。列/字段的数量也保持不变。

数据文件夹中的示例数据文件有不同的安排。其他安排也是可能的，而且

赋值标记将使用他们自己安排的列/字段来测试你的程序。

需要考虑的一些基本事项：

- 一些不感兴趣的字段将在问题5中被读取和丢弃。其他字段将在以后使用（环境空气温度T，太阳辐射SR）。未来的工作，包括任务不会有相同顺序的列。字段名（列名）将是相似的，但列的排列可以是不同的顺序。你将如何保证你的读取程序的未来性？
- 你怎么知道数据被正确读入？提示：在你试图将数据读入数据结构之前，你需要读取、分割数据，并将其打印在屏幕上或文件中以便调试。然后手动检查输出。你的程序不需要对整个数据文件进行这样的检查--只需检查前几行就足够了。一旦这个有限的手动检查成功，检查你是否可以正确地读取整个文件。你通过读取`MetData-31-3.csv`并将日期、时间和风速打印到一个输出文件（`testoutput.csv`）来做这个检查。逗号分隔日期、时间和风速，每行有3个数据值。

在文本编辑器和电子表格中打开`MetData-31-3.csv`和`testoutput.csv`，然后手动检查`testoutput.csv`中的数值是否正确。一旦你能直观地确认数值正确，你就开始做这个问题，并使用`windlog`数据结构。

通过将`windlog`中的所有数据打印到输出文件called `testoutput.csv`中来测试你的程序。输出文件中的三个值必须用逗号分开。做一次目视检查，看看`testoutput.csv`中的数据是否正确。一旦你能确认输出是正确的，在屏幕上做平均速度计算和输出日期/时间。确保你通过查看电子表格（应用测试）中的输入数据文件来检查屏幕输出是否正确。使用电子表格的平均公式计算来验证你的程序的计算。对样本的标准偏差重复这个过程。

- 你能想出任何好的理由，为什么`WindLogType`应该是一个类而不是一个结构？你什么时候会使用一个类，什么时候会使用一个结构？请仔细思考这个问题。本课题和以前的课题中的阅读内容将为您提供指导。作业确实要求（必修⁴）你为你采取的数据结构方法提供理由（原因）。

问题的输入格式样本：

有两种格式可供试验。第一种比第二种容易。只有在你能完成第一个版本的情况下，才能尝试第二个版本。

下面的第一个版本需要用于问题5（作业1的准备）。

让你的设计和程序与实验室4的注册练习中使用的信息相同。一旦你有了可行的设计和程序，想想如何将更多的细节（如上面的问题2）纳入输入数据中。你会使用一个单独的数据文件来存储单位协调员的信息吗？

第一个版本的输入数据格式（csv）：（所有作业都将使用这种输入格式）

⁴ 任务必须满足强制性要求。因此，对细节的关注至关重要。

这种格式通常被称为CSV（逗号分隔的值）格式。你会在ICT159的先决条件/修订练习中使用这种格式。这种格式通常在一行中出现的字数固定时使用。数据可能会丢失。在上述问题5和作业中使用这种格式。

所有的数据都用逗号隔开，直到行末。每一行(行)被称为一个记录。每条数据都被称为一个字段。

ID、学期、单位ID、单位名称、学分、分数、日期

对于问题5，数据将来自已经提供的数据文件。

第二个版本的数据格式：（不用于作业--用于额外的练习）。

只有当你不能让第一个版本工作，已经完成了上面的问题5，并且正在寻找更有"挑战性"的东西时，才能尝试这个版本。作业1不需要这种格式。

一个学生ID的所有数据都在一行。主要的分隔符是分号（；），次要的分隔符是逗号（，）。每一行被称为一条记录。

ID ; 学期, 单位ID, 单位名称, 学分, 分数, 日期, 单位ID, 单位名称, 学分, 分数, 日期 ; 学期, ...等。

从上面可以看出，在一个学期内，可以有不止一个单位。任何在分隔符之间的项目都被称为"字段"。因此，单位名称是一个字段，ID、学期、学分.....等也是。

日期是指分数被输入系统的时间。学期是指学习该单元的时间。

请注意，记录（行）是由学生的ID识别的。因此，每个学生在一行中都有他/她的所有信息。在学生编号之后，分号将根据学期完成的单位分开。

一旦你有了一个使用上述数据格式的工作设计和程序，想一想如何将额外的细节纳入输入数据。

注意事项：

从上面来看，你会注意到一条记录包含一个或多个字段，每条记录出现在一行上。

想一想你可能会如何处理另一种日期格式，即像问题5的数据文件那样，将日期和时间放在一个字段中。这意味着日期和时间只由一个或多个空格分隔。

在考虑如何解决上述日期问题时，你会让日期类处理所有这些可能性吗？如果是这样，当输入格式再次被改变为另一种规格时，会发生什么？

你现在需要完成练习5。一旦你完成了准备性的实验，作业1就不那么复杂了。这种方法也适用于作业2。作业主要是"将实验室工作打包成一个更大的应用"。

要进入作业1，请提交练习5的解决方案。