

## 测试 1

请阅读以下问题说明，并回答后面的 7 个问题。您可以将答案写在试卷的正反面。请在答案上标注清楚。

所有类都必须是最小的和完整的。采用模型-视图-控制器（MVC）模式，并适当使用 SOLID 原则。

如果需要更多空间，可在每页背面作答。确定答案并将答案与要回答的问题匹配。

图书的一些特征包括书名、作者、出版商、国际标准书号、价格和出版年份。

- *bookType* 类的每个对象都可以保存一本书的以下信息：书名（至少包含一位作者）、出版商、国际标准书号（ISBN）、价格和库存数量。同一本书可以有多个作者。
- 包含对*图书类型*对象执行各种操作的成员函数以及所需的构造函数或析构函数。
- 根据需要加入辅助功能。

模板*向量*类用于需要数组的地方。*Vector* 类提供了与数组相同的功能，但数组被封装在 *Vector* 类中。因此，可以通过 *Vector* 类的方法控制对数组的访问。*Vector* 类的大小可根据需要增大。

不允许使用 STL 容器和算法。您可以使用 std 命名空间中的 cin、cout、string 或 stringstream。

1. 为上述所有类绘制详细的 UML 类图。

[15分]

2. 编写 *bookType* 类的 C++ 规范。这是 *bookType.h* 中的代码

[15分]

3. 编写 *bookType* 类的 C++ 实现。下面是 *bookType.cpp*.

[20分]

4. 编写模板/向量类的 C++ 规范。这是 *Vector.h* 中的代码

[10分]

5. 解释矢量类中每个方法背后的原理，以及任何一个类的辅助函数。

[5个标记]

6. 提供 *bookType* 类的测试计划。用表格显示。

[15分]

7. 编写一个 C++ 测试程序来测试你的 *bookType* 类。测试程序从文本数据文件中读取数据。书籍的数据来自名为 *books.csv* 的文件。每本书的数据使用逗号分隔值存储为一行（记录）。您可以选择数据行中字段的顺序。由于可能有多个作者，因此请考虑如何在每条记录中使用不同数量的作者格式。

解释您的解决方案是如何设计以适应模型-视图-控制器 (MVC) 模式的。您的解释需要明确指出哪些类、子程序（函数、过程）适合 MVC 的哪个部分。您还应解释为什么它们适合 MVC 组件。

使用代码注释，确定解决方案中包含的 **SOLID** 原则。

[20分]

\*\*\*\*问题结束。后面有 2 页空白\*\*\*\*





