

作业练习3：类的继承

面向对象程序设计 (C++)

WHUT-CS 2022 Spring

I. 作业目的

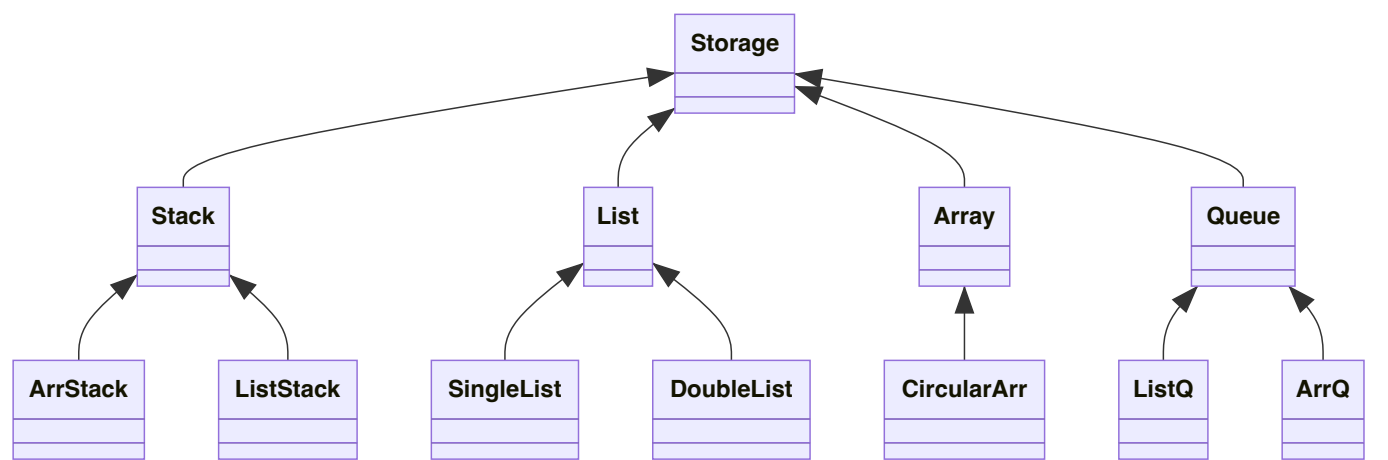
本次实验主要在于学习使用C++类继承机制实现程序功能。C++中的类继承机制能够用于表示类之间的“is-a”关联，合理使用能够有效减少重复代码，并进而实现多态行为。

本次作业将基于类继承体系设计并实现具有数据容器功能的一系列类，通过对这些类的开发过程学习掌握C++中的类继承机制。

注意：本次练习所涉及的部分类的功能在作业练习2中已经进行了实现，但并未基于类继承机制进行设计。本次练习应充分使用继承机制对相关类的功能进行重新改进和完善。

II. 作业要求

1: 类继承体系



在如上图所示的类继承体系中包含多个常见的数据容器类。这些类以Storage为其根类，通过父类与子类之间的继承关系形成了相互之间的关联。

- **Storage**：该类表示所有数据容器类的基本功能特性，即数据元素能够放入容器并从中取出。其他所有类通过继承机制作为该类的直接或间接子类，这意味着所有这些类都具备数据容器的基本功能。**Storage**类可以被声明为一个抽象类（abstract class），在其中通过必要的纯虚函数定义了所有子类应具备的公共函数签名。
- 上图中所有位于叶节点处的类都应被声明为具体类（concreate class），它们可以被用于创建对象实例。
- **Array**：该类用于表示数组容器。在本例中，**Array**类应被实现为具体类，并且作为**CircularArr**的直接父类用于展示两个具体类之间的继承关系。

- 上图中的其他类全部声明为抽象类，它们分别包含若干纯虚函数供子类实现。

2: 代码框架

本次作业包含三个代码文件，说明如下

- **storage.h**: 所有类的声明代码
- **storage.cpp**: 所有类的功能实现代码
- **hmk3.cpp**: 主函数文件，用于编写功能测试代码

编写代码时应通过注释对代码内容进行必要且详细的描述，如果你觉得样例代码框架中存在错误，也可以对代码进行改正并在注释中描述你认为的错误原因。

样例代码中仅提供了最基础的代码框架，你可以对各个类的实现提供更多你认为必要的功能。

III. 作业内容

本次作业包含如下任务：

- 在**storage.cpp**文件中补齐所有缺失的代码，提供必要的函数实现；
- 在 **storage** 类中增加一个函数 **bool find(Item n)**，如果容器中包含指定的数据元素时，该函数返回**true**，否则返回**false**。在 **storage** 类的所有子类中提供该函数的实现；
- 在**hmk3.cpp**中的**main**函数中编写代码，用于调用并测试所有容器类中的公共函数；
- 针对 **queue** 及其子类，编写代码测试其**enqueue**和**dequeue**功能的多态特性；
- 针对 **stack** 及其子类，编写代码测试其**push**和**pop**功能的多态特性；
- 针对 **array** 及其子类，编写代码测试其[]操作符的多态特性；
- 编译并运行你编写的程序；

IV. 作业提交

- 所有同学请独立完成本次作业，不允许共同完成和抄袭
- 通过武汉理工大学“理工智课”平台中的本课程页面提交作业结果
 - 课程编号：69274（面向对象程序设计B）
- 截止日期: 2022 Dec. 31 23:00

VI. 参考文献

[1] C++ Primer Plus (edition 6), Stephen Prata, ISBN: 978-0321-77640-2, Pearson

[2] C++语言程序设计 (第5版), 郑莉, 董渊, ISBN: 978-7302-56691-5, 清华大学出版社