

C++ 课前热身练习实验报告

袁雨 PB20151804

一、实验目的

- 学习使用 CMake 来搭建项目
- 学习使用 Visual Studio 2019 进行编程，学会其 debug 工具来调试代码
- 学习面向对象 C++ 编程，特别是类（`class`）的封装特性及构造函数、析构函数、函数重载、运算符重载等
- 熟悉 C++ 指针、动态内存分配、预编译头机制等
- 学习模板 `template`
- 学习 STL 的 `vector`、`list`、`map` 等
- 学习静态库 `lib`，动态库 `dll` 的编写

二、实验内容

小练习 1. 基础的动态数组

详细说明见于 [documents/1_BasicDArray](#).

完成 [src/executables/1_BasicDArray](#).

小练习 2. 高效的动态数组

详细说明文档见于 [documents/2_EfficientDArray](#).

完成 [src/executables/2_EfficientDArray](#).

小练习 3. 模板动态数组

详细说明见于 [documents/3_TemplateDArray](#).

仿照小练习 1，在文件夹 [src/executables/](#) 中添加文件夹 `3_TemplateDArray`，并在其内

- 添加文件 `TemplatedArray.h`
- 添加文件 `main.cpp`
- 添加文件 `CMakeLists.txt`，同于 [src/executables/1_BasicDArray/CMakeLists.txt](#)

重新 CMake 后得到新子项目 `3_TemplateDArray`

小练习 4. 基于 `list` 的多项式类

详细说明见于 [documents/4_list_Polynomial](#)

这里将 `PolynomialList` 编写成了动态库，具体查看 [src/libraries/shared](#)，编译后会生成 [lib/CppPractices_libraries_shared\(d\).dll](#)

你需要补充完成 [src/libraries/shared/PolynomialList.cpp](#)

[4 list Polynomial](#) 会测试该动态库

小练习 5. 基于 `map` 的多项式类

详细说明见于 [documents/5_map Polynomial](#)

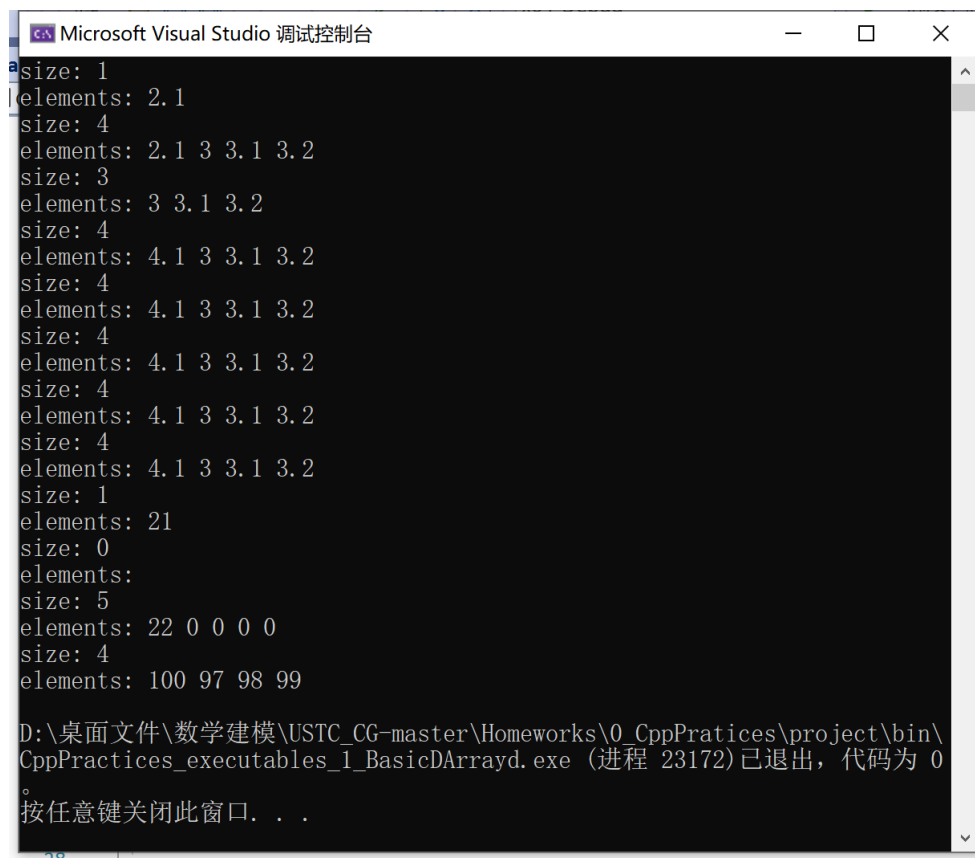
这里将 PolynomialMap 编写成了静态库，具体查看 [src/libraries/static](#)，编译后会生成 [lib/CppPractices_libraries_static\(d\).dll](#)

你需要补充完成 [src/libraries/static/PolynomialMap.cpp](#)

[5_map Polynomial](#) 会测试该静态库，另外该子项目还用到了小练习 4 的动态库 PolynomialList，其中会测试小练习 4 和小练习 5 的性能差异

三、实验结果

小练习 1. 基础的动态数组



```
Microsoft Visual Studio 调试控制台
size: 1
elements: 2.1
size: 4
elements: 2.1 3 3.1 3.2
size: 3
elements: 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 1
elements: 21
size: 0
elements:
size: 5
elements: 22 0 0 0 0
size: 4
elements: 100 97 98 99

D:\桌面文件\数学建模\USTC_CG-master\Homeworks\0_CppPractices\project\bin\
CppPractices_executables_1_BasicArrayd.exe (进程 23172) 已退出，代码为 0
按任意键关闭此窗口. . .
```

小练习 2. 高效的动态数组

```
Microsoft Visual Studio 调试控制台

size: 4
elements: 2.1 3 3.1 3.2
size: 3
elements: 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 1
elements: 21
size: 0
elements:
size: 5
elements: 22 0 0 0 0
size: 4
elements: 100 97 98 99

D:\桌面文件\数学建模\USTC_CG-master\Homeworks\0_CppPractices\project\bin\
CppPractices_executables_2_EfficientDArrayd.exe (进程 27324)已退出, 代码
为 0。
要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试
停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

小练习 3. 模板动态数组

```
Microsoft Visual Studio 调试控制台

elements: 2.1
size: 4
elements: 2.1 3 3.1 3.2
size: 3
elements: 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 4
elements: 4.1 3 3.1 3.2
size: 1
elements: 21
size: 0
elements:
size: 5
elements: 22 0 0 0 0
size: 4
elements: d a b c

D:\桌面文件\数学建模\USTC_CG-master\Homeworks\0_CppPractices\project\bin\
CppPractices_executables_3_TemplateDArrayd.exe (进程 2252)已退出, 代码为
0。
按任意键关闭此窗口. . .
```

小练习 4. 基于 `list` 的多项式类

小练习 5. 基于 `map` 的多项式类

四、实验分析与心得体会

通过小练习1, 了解和掌握了 C++ 的基本语法。初步了解类（对象）的编写, 了解构造函数、析构函数、函数重载等; 熟悉和巩固了指针、动态内存分配的机制与操作。

通过小练习2, 学会了使用预先多分配一些内存的方法来提高动态数组的效率, 理解对象的 `public` 接口的重要性。

通过小练习3, 学会了使用 `template` 来处理不同类型的动态数组类, 初步了解和使用 STL 的 `vector`。

通过小练习4, 学会了使用 `vector` 和 `list`、了解生成和使用动态库。

通过小练习5, 学会了使用 `map` 来实现 `polynomial` 类, 了解静态库的编写与使用;

对比发现, `list<T>` 容器是双向链表, 因此可以有效的在任何位置添加和删除。列表的缺点是不能随机访问内容, 要想访问内容必须在列表的内部从头开始便利内容, 或者从尾部开始。`map<K, T>` 映射容器: K表示键, T表示对象, 根据特定的键映射到对象, 可以进行快速的检索。`map` 的效率更高, 二者的时间复杂度不同。

