

中山大学移动信息工程学院本科生实验报告

(2015-2016 学年春季学期)

课程名称: Data structures and algorithms

任课教师: 张子臻、黄淦

年级	15	专业(方向)	软件工程(移动信息工程)
学号	15352408	姓名	张镓伟
电话	13531810182	Email	709075442@qq.com
开始日期	2016. 3. 21	完成日期	2016. 3. 21

1. 实验题目

1000:

完成一个 MyString 类如下, 该类有构造函数、拷贝构造函数、析构函数、成员函数和静态成员函数, 有一个 C 字符串指针变量, 还有一个静态成员变量 numberOfObjects。

1001:

有一个向量类 MyVector, 包括一个点的坐标位置 x、y 和 z, 实现其构造函数和四个友元函数, 完成两个向量的加法(友元函数)、减法(友元函数)、点乘(友元函数)与叉乘(友元函数)运算以及一个输出当前向量的函数。

1002:

对于一个表达式可以写成:1+2-3+4, 设计一个 Number 类, 完成其中的 add 方法和 sub 方法, 这两个方法分别接受一个 int 型的参数, 此外 Number 还有一个构造函数, 接受的参数也是 int 型, 最后你需要实现一个 print 的方法, 用于打印结果。这个类应能像那个表达式一样有类似的链式调用功能, 比如该表达式可以这样写:

```
Number op(1);  
op.add(2).sub(3).add(4)  
op.print();    //在一行上输出最终结果: 4
```

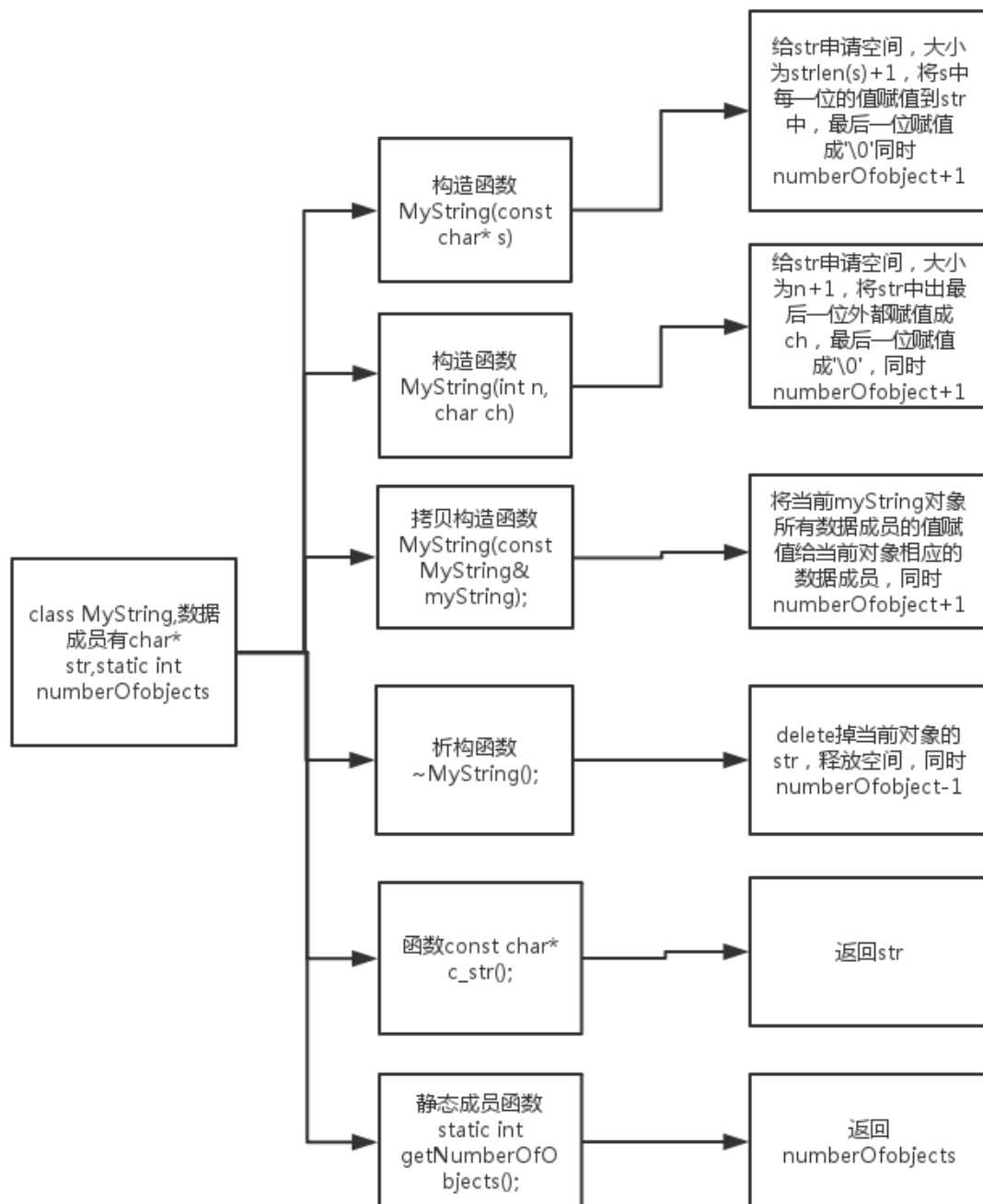
2. 实验目的

- 熟悉 C++ 类的各种函数和数据成员的编写与应用。
- 熟悉类的静态数据成员的使用

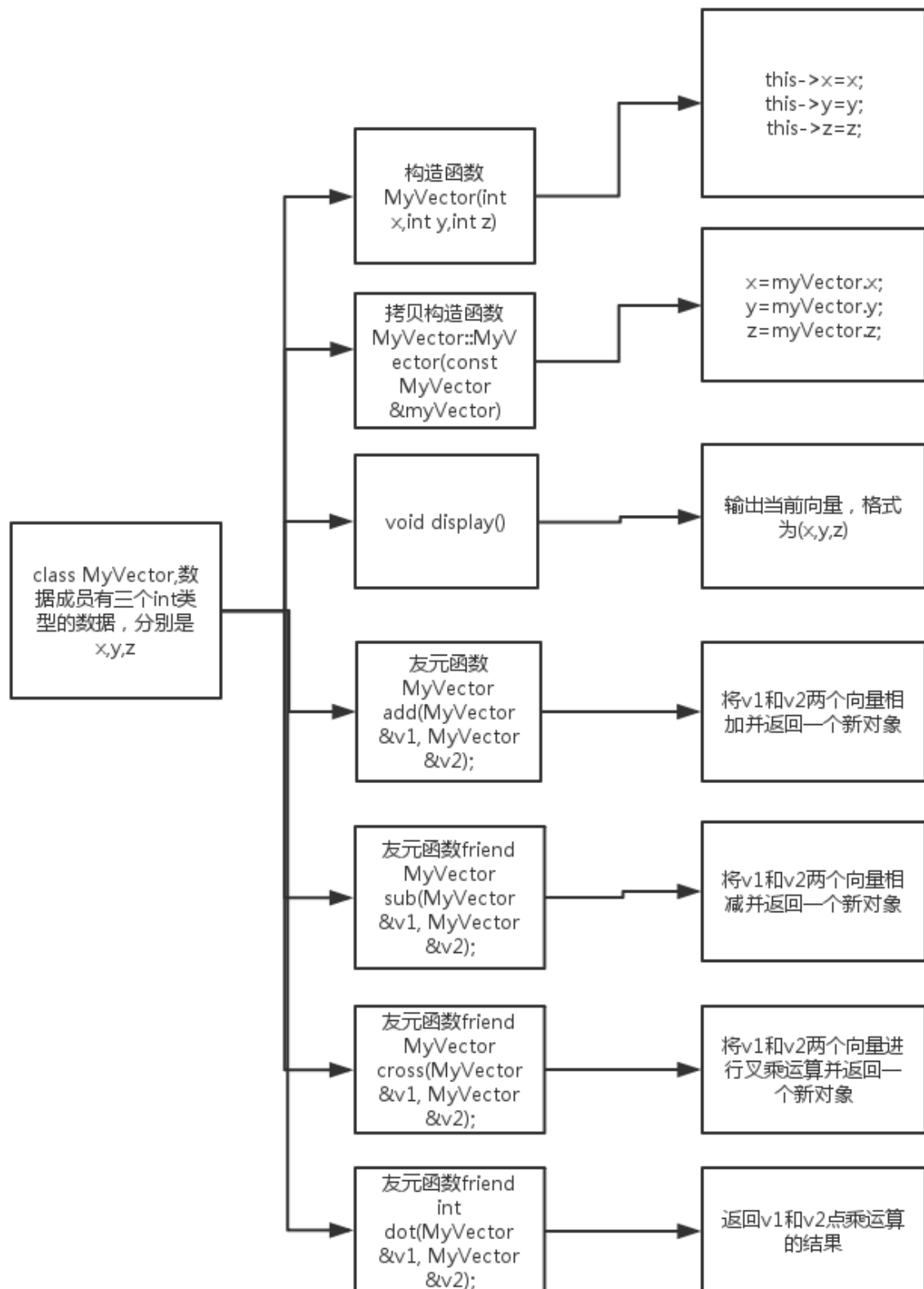
- C. 熟悉友元函数的使用，明确友元函数和其他类的成员函数有什么不同
D. 知道如何使用串联函数，知道这种写法有什么限制。

3. 程序设计

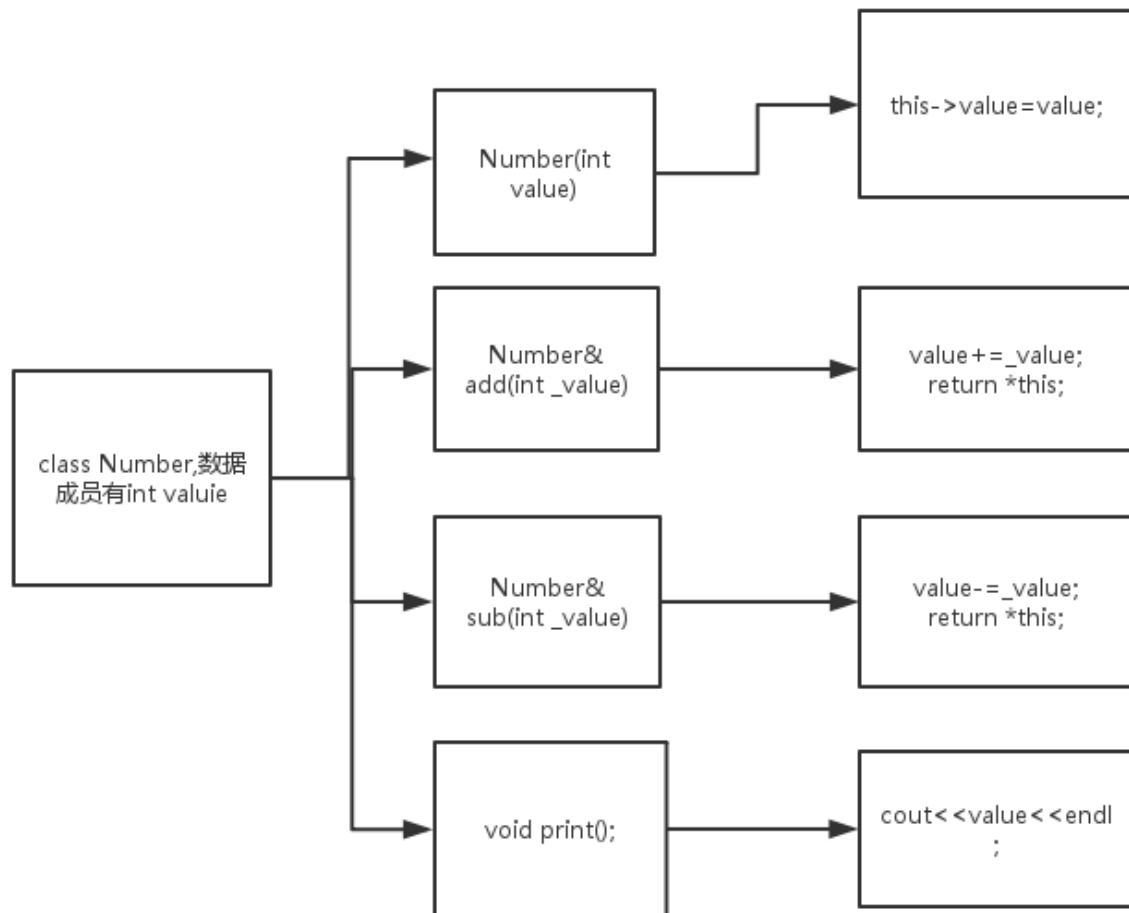
1000:



1001:



1002:



三题完整可测试代码详见附件 1000.cpp、1001.cpp 和 1002.cpp

4.程序运行与测试

1000:

按照如下主函数运行：

```

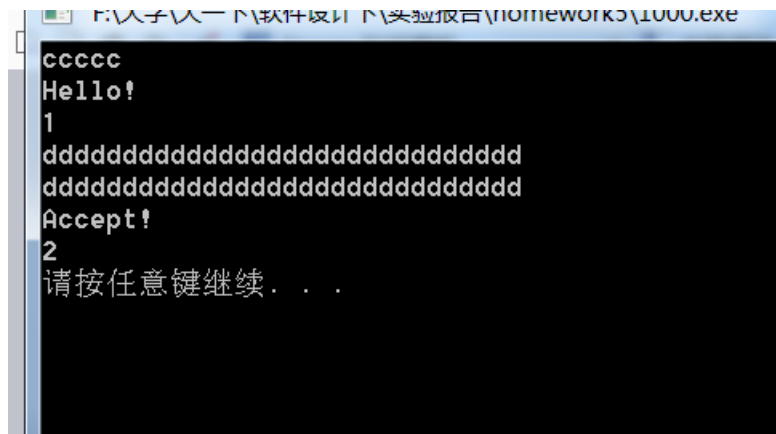
int main(){
    MyString ms1(5, 'c');
    cout << ms1.c_str() << endl;
    MyString *ptr_ms2 = new MyString("Hello!");
    {
        MyString ms3(ms1);
    }
    cout << ptr_ms2->c_str() << endl;
    delete ptr_ms2;
    cout << MyString::getNumberOfObjects() << endl;

    MyString ms4(30, 'd');
    cout << ms4.c_str() << endl;
    MyString *ptr_ms5 = new MyString("Accept!");
    {
        MyString ms6(ms4);
        cout<<ms6.c_str()<<endl;
    }
    cout << ptr_ms5->c_str() << endl;
    delete ptr_ms5;
    cout << MyString::getNumberOfObjects() << endl;

    return 0;
}

```

结果:



```

F:\大学\大一\软件设计\实验报告\homework3\1000.exe
cccccc
Hello!
1
dddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddd
Accept!
2
请按任意键继续. . .

```

1001:

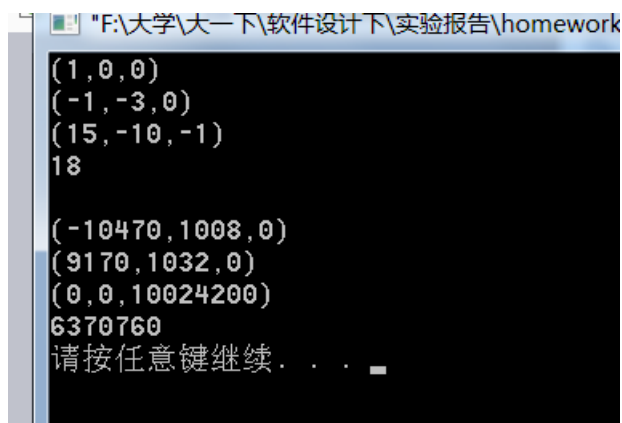
按照如下主函数运行:

```

1 int main() {
2     MyVector a;
3     MyVector b(1);
4     MyVector c(2, 3);
5     MyVector d(3, 4, 5);
6     MyVector e(c);
7
8     MyVector x = add(a, b);
9     x.display();
10
11    MyVector y = sub(b, c);
12    y.display();
13
14    MyVector z = cross(c, d);
15    z.display();
16
17    int w = dot(d, e);
18    cout << w << endl;
19
20    cout<<endl;
21    MyVector f(-650, 1020);
22    MyVector g(-9820, -12);
23    MyVector k=add(f,g);
24    k.display();
25
26    k=sub(f,g);
27    k.display();
28
29    k=cross(f,g);
30    k.display();
31
32    cout<<dot(f,g)<<endl;
33    return 0;
34 }

```

结果:



```

(1,0,0)
(-1,-3,0)
(15,-10,-1)
18

(-10470,1008,0)
(9170,1032,0)
(0,0,10024200)
6370760
请按任意键继续. . .

```

1003:

按照如下主函数运行:

```

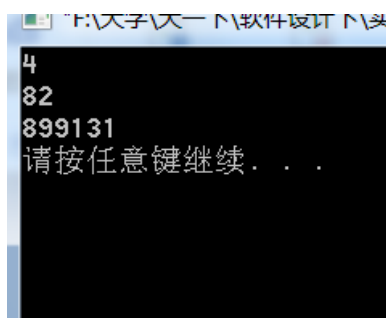
int main() {
    //1+2+3+4;
    Number n(1);
    n.add(2).sub(3).add(4);
    n.print();

    //1-2+9-9+120+45-89+7
    Number m(1);
    m.sub(2).add(9).sub(9).add(120).add(45).sub(89).add(7);
    m.print();

    //10000-3467+999999-88765+1234+3+8-6751+2345-5557
    Number k(10000);
    m.sub(3467).add(999999).sub(88765).add(1234).add(3).add(8).sub(6751).add(2345).sub(5557);
    m.print();
    return 0;
}

```

结果:



5. 实验总结与心得

- A. 对类的编写和应用有了更深一层的了解，对类更加熟悉了。
- B. 静态成员函数在类外编写时前面不用加 `static` 关键字。
- C. 静态成员可以调用静态成员，静态成员只能在类外初始化。
- D. 友元函数是类外的函数，所以它的声明可以放在类的私有段或公有段且没有区别。我们可以直接调用友元函数，不需要通过对象或指针。
- E. 虽然友元函数能提高效率，表达更简单、清晰，但是友元函数会破坏类的封装性，一般在我们需要进行运算符重载或者两个类要共享数据时才使用，其他情况尽量不使用。
- F. 串联函数其实就是当前函数的返回值是某个类的对象，所以能满足继续调用函数的条件。其他返回值的函数并不能这样做。

附录、提交文件清单

实验报告一份：实验报告.pdf

代码三份：1000.cpp

1001.cpp

1002.cpp