

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	张稼伟
电话	13531810182	Email	709075442@qq.com
开始日期	2016. 3. 14	完成日期	2016. 3. 14

1. 实验题目

1000:

设计一个 MyInteger 类, 包括一个 int 数据成员 value, 初始化为 value 的构造函数, 判断值是否为奇数, 偶数, 质数, 相等的成员函数, 这些成员函数分别有无参的, 有一个 int 值为参数的, 有一个 MyInteger 类的对象为参数的, 带参数的函数都是静态的, 除此之外还有一个静态函数将字符串转换成 int 值并赋值给 value。

1001:

定义一个类控制机器人的移动, 机器人在二维平面坐标上移动, 接受指令: 'N', 向上移动; 'E', 向右移动; 'S', 向下移动; 'W' 向左移动。维护机器人当前位置坐标。

2. 实验目的

- A. 熟练掌握 C++ 小程序的编写调试。
- B. 熟练掌握类的编写及使用。
- C. 体会面向对象编程的特点。
- D. 体会 static 关键字的用法

3. 程序设计

1000: 按照相关函数的功能编写代码, 其中值得注意的几点有:

- 1. 静态成员在类外编写时前面不用加 static 关键字。
- 2. 静态成员只能调用静态成员, 不能再类内初始化。
- 3. 静态成员之间可互相访问, 但静态成员函数必须通过对象名来访问

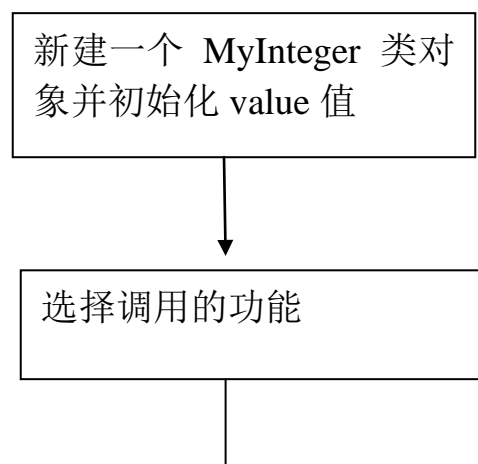
问非静态数据成员。

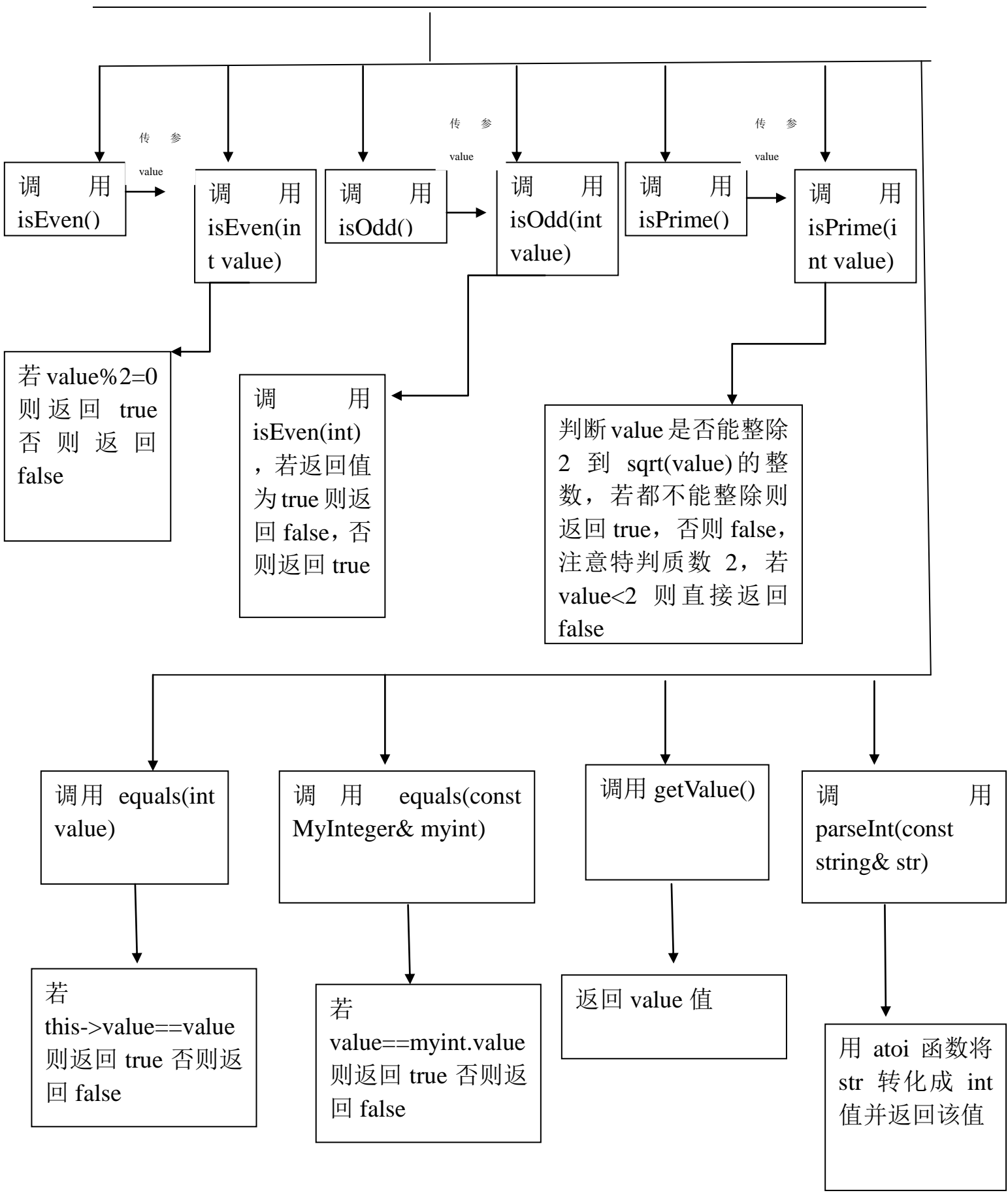
4. 判断奇数的时候注意要先取反，因为负数%2 还是负数，也可以通过调用判断偶数的函数来判断。
5. 判断质数时采用 $O(\sqrt{n})$ 的方法。

类的定义如下：

```
class MyInteger
{
private:
    int value;
public:
    MyInteger(int);
    int getValue()const;
    bool isEven()const;
    bool isOdd()const;
    bool isPrime()const;
    static bool isEven(int);
    static bool isOdd(int);
    static bool isPrime(int);
    static bool isEven(const MyInteger&);
    static bool isOdd(const MyInteger&);
    static bool isPrime(const MyInteger&);
    bool equals(int)const;
    bool equals(const MyInteger&)const;
    static int parseInt(const string&);
};
```

流程图如下：

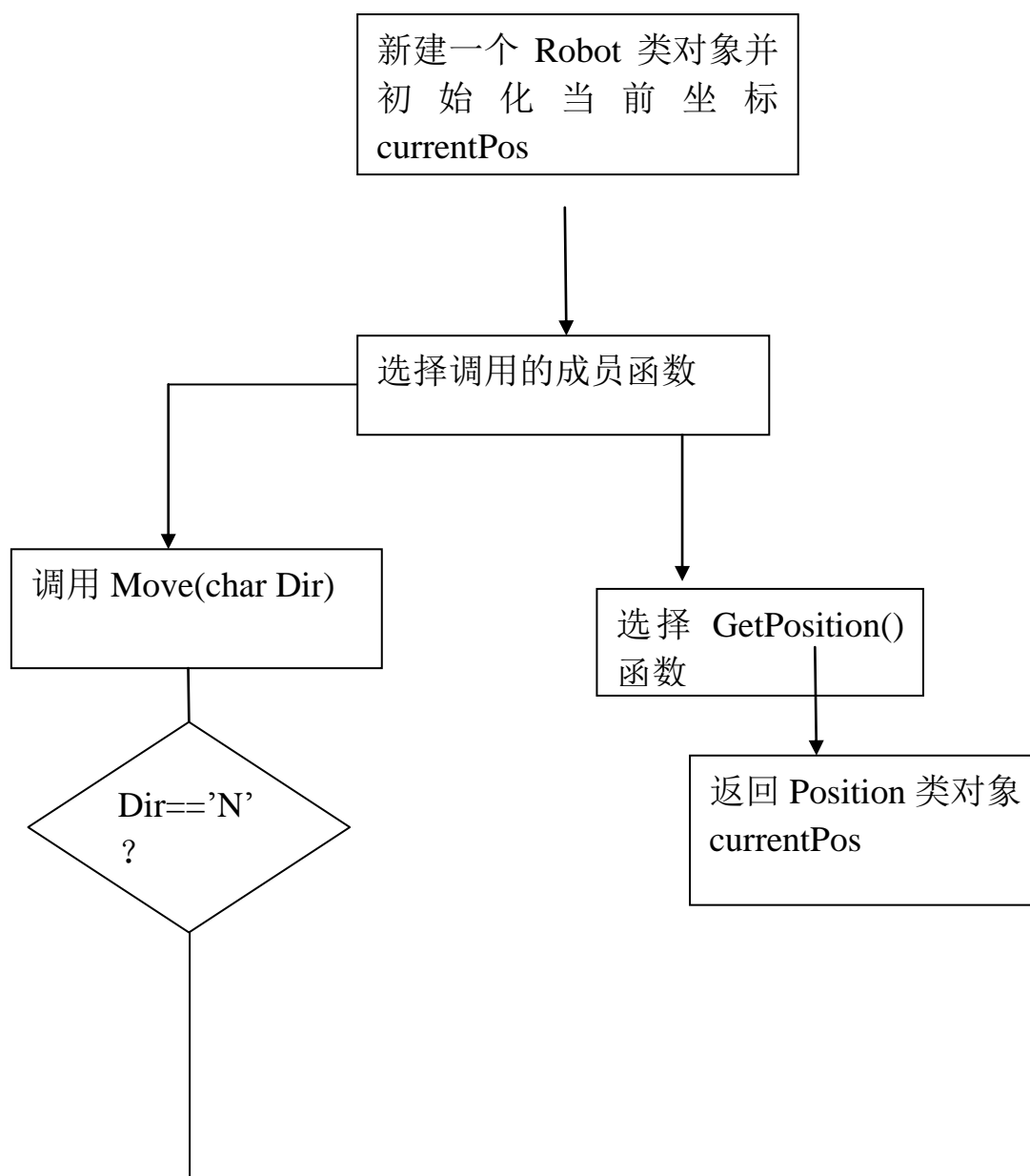


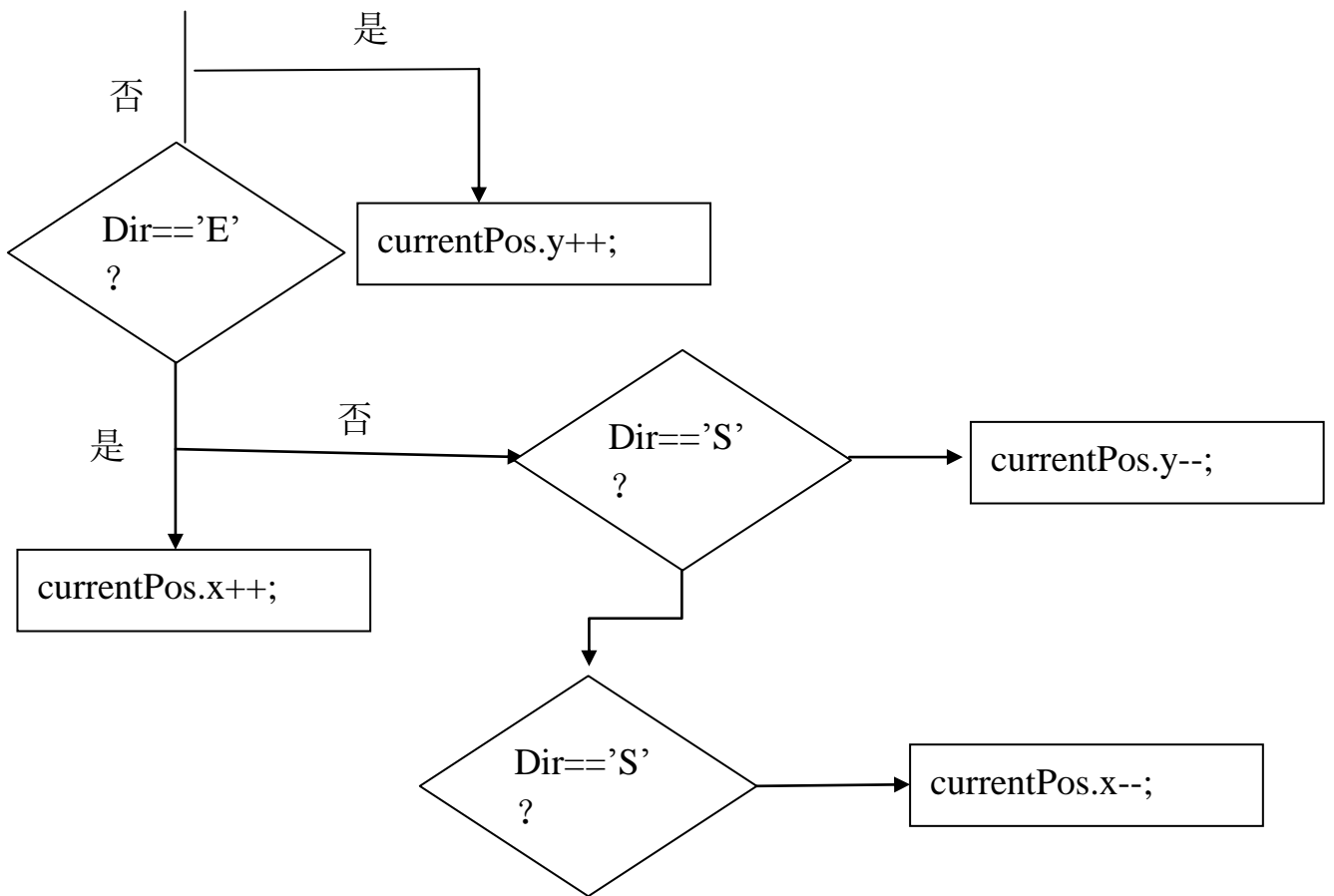


1001:

此题题干给出了类定义，有两个类，分别是 Position 类, 用来保存坐标，还有一个 Robot 类，用来维护机器人的位置，Robot 类中的成员有 Position。机器人移动的时候，按照 NESW 的指令不同分别对坐标进行相应的加减即可。

流程图如下：





两题完整可测试代码详见附件 1000.cpp 和 1001.cpp

4. 程序运行与测试

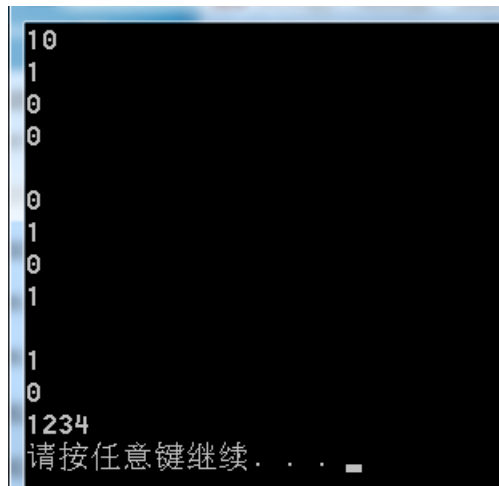
A: 按照如下主函数运行:

```

int main() {
    MyInteger myint(10);
    cout<<myint.getValue()<<endl;
    cout<<ostream std::cout<<endl;
    cout<<myint.isOdd()<<endl;
    cout<<myint.isPrime()<<endl;
    cout<<endl;
    cout<<myint.isEven(-5)<<endl;
    cout<<myint.isOdd(-5)<<endl;
    cout<<myint.isPrime(-5)<<endl;
    cout<<myint.isPrime(97)<<endl;
    cout<<endl;
    MyInteger others(13);
    cout<<myint.equals(10)<<endl;
    cout<<myint.equals(others)<<endl;
    cout<<myint.parseInt("1234")<<endl;
}

```

结果:



```

10
1
0
0
0
1
0
1
1
1
0
1234
请按任意键继续...

```

B: 按照如下主函数运行:

```

int main() {
    Position c;
    c.x = 0;
    c.y = 1;
    Robot a;
    cout << a.GetPosition().x << ' ' << a.GetPosition().y << endl;
    Robot b( c );
    cout << b.GetPosition().x << ' ' << b.GetPosition().y << endl;
    b.Move('E');
    cout << b.GetPosition().x << ' ' << b.GetPosition().y << endl;
    b.Move('N');
    cout << b.GetPosition().x << ' ' << b.GetPosition().y << endl;
    b.Move('W');
    cout << b.GetPosition().x << ' ' << b.GetPosition().y << endl;
    b.Move('S');
    cout << b.GetPosition().x << ' ' << b.GetPosition().y << endl;

    b.Move('s');
    cout << b.GetPosition().x << ' ' << b.GetPosition().y << endl;
    return 0;
}

```

结果:



```

0 0
0 1
1 1
1 2
0 2
0 1
0 1
0 1
请按任意键继续...

```

5. 实验总结与心得

通过这两道题，我对类的应用的掌握又更上了一层楼。面向对象编程将我们的思维更多的是引向大局观，这对理清代码思路有很好的帮助。另外，这次也对 static 关键字的应用有了进一步的理解与掌握，以往只知道 static 关键字在全局变量上的用法，却不知在类中的使用。这次在调试第一题的代码的时候，我知道了静态成员在类外编写时前面不用加 static 关键字；静态成员只能调用静态成员，不能再类内初始化；静态成员之间可互相访问，但静态成员函数必须通过对象名来访问非静态数据成员。还有这次知道了可以一个类的数据成员可以是另一个类，收获颇丰。

附录、提交文件清单

实验报告一份：实践报告.pdf

代码两份:1000.cpp

1001.cpp