第7章 练习

7.1节

7.1

略

7.2 - 7.3

略(第一章的类头文件已经实现这些要求了)

7.4 - 7.5

```
class Person {
public:
    const string &getName() const {
        return name_;
    }

    const string &getAddress() const {
        return address_;
    }

private:
    string name_;
    string address_;
};
```

- 设计成引用，效率更高；

  设计成const，保证不会改变

7.6 - 7.7

略（第一章的代码文件已经实现）

7.8

```
在read函数中，会改变Sale_data中的数据成员；
在print函数中，只需输出即可，使用const可以确保安全
```

7.9

```
istream & operator>>(istream &is, Person &p){
    is>>p.name_>>p.address_;
    return is;
}

ostream &operator<<(ostream &out, Person &p) {
    out<<p.name_<<" "<<p.address_;
    return out;
}
// 重载的运算符函数都是Person的友元
```

7.10

```
读入data1后，再读入data2，判断读入data2是否成功
```

7.11 - 7.15

略（见代码）

7.2节

7.16

```
没有限制；
一般来说，具体实现细节在private后；
而可调用的成员函数在public中
```

7.17

```
有区别；默认的访问控制级别不同
```

7.18

略

7.19

略

7.20

```
使得函数或类可以访问private、protected限制下的成员，但是它破坏了类的封装性
```

7.21 - 7.22

略

7.3节

7.23

略

7.24

略

7.25

略

7.26

```
将函数实现写在类里面
```

7.27

略

7.28 - 7.29

```
第二个语句编译错误
```

7.30

略

7.31

```
class X;
class Y{
        X *p_X;
}
class X{
    Y *p
}
```

7.32

```
class Window_mgr{
    public:
        void clear();
}
class Screen{
    frined Window_mgr::clear();
}
```

7.4节

7.33

```
错误，pos类型的作用域在类中
Screen::pos Screem::size() const
```

7.34

```
函数声明中pos height和报错
```

7.35

```
// 存在错误
Exercise::Type Exercise::setVal(Type parm) {
    val += parm + initVal();
    return val;
}
```

7.5节

7.36

```
初始化列表的顺序和数据定义的顺序有关；
所以rem(base%j)会先执行，但此时base是未定义的
```

7.37

```
版本3；
版本1；
版本1；
```

7.38

```
Sales_data(istrream &is = std::cin);
```

7.39

```
不合法;
Sales_data nex; 编译器无法区分调用哪个构造函数
```

7.40

略

7.41 - 7.42

略

7.43

```
class C {
public:
   C():t(0){}       // 必须自定义默认构造函数
private:
    T t;
};
```

7.44

```
不合法；NoDfault不存在默认构造函数
```

7.45

```
合法
```

7.46

```
1. 对；程序员不提供构造函数时，编译器也会提供一个默认构造函数
2. 不一定；所有参数都有默认值的构造函数也可成为默认构造函数
3. 错
4. 错；对于类类
```

7.47

```
建议设置成explicit，避免在程序中出现未知的隐式类转换
```

7.48

```
item1 调用 Sales_data(const std::string)构造函数;
item2 首先发生隐式类型转换，将字符数组转换为string后，调用 Sales_data(const std::string)构造函数;
-------------------------------------
如果是explict的，item1没问题，item2编译报错
```

7.49

```
1. 合法；发生隐式类型转换
2. 不合法；引用绑定在一个临时变量是不被允许的
3. 不合法；combine函数需要改变*this,不应该设置为常量函数
```

7.50

```
将单参数的构造函数设置为explicit
```

7.51

```
int getSize(const std::vector<int>&);
//这样的使用是否显得比较迷惑
getSize(34);
```

7.52

略

7.53

略（书上有）

7.54

```
constexpr函数只能有return语句
```

7.55

```
不是；s的类型是string，不是字面值类型
```

7.6节

7.56

略

7.57

略

7.58

```
错
对
错
不能在类内定义静态数据成员
```