读书内容：C++ prime plus 第9~10 章并总结关键内容

总结：

第九章：

1. 组织程序的一种策略：

* 头文件：包含结构声明和使用这些结构的函数的原型；
* 源代码文件：包含与结构有关的函数的代码；
* 源代码文件：包含调用与结构相关的函数的代码。

1. 头文件常包含的内容：

* 函数原型；
* 使用#define或const定义的符号常量；
* 结构声明；
* 类声明；
* 模板声明；
* 内联函数。

1. 作用域为局部的变量只在定义它的代码块中可用，作用域为全局的变量在定义位置到文件结尾之间都可用。自动变量的作用域为局部，静态变量的作用域取决于它如何被定义。
2. 动态内存的分配和释放是用new和delete进行的。
3. 静态持续变量：

* 外部链接性：在代码块的外面声明它；（外部变量或全局变量）
* 内部链接性：在代码块的外面声明它，并使用static限定词；
* 没有链接性：在代码块的内部声明它，并使用static限定词。

1. 如果要在多个文件使用外部变量，只需在一个文件中包含该变量的定义，但在使用该变量的其他所有文件中必须使用关键字extern声明它。
2. 声明区域是可以在其中进行声明的区域，潜在作用域从声明点开始，到其声明区域的结尾。
3. 名称空间可以是全局的，也可以位于另一个名称空间中，但不能位于代码块中，因此，在默认情况下，在名称空间中声明的名称的链接性为外部的，除非引用了常量。
4. 名称空间的一些指导原则：

* 使用在已命名的名称空间中声明的变量，而不是使用外部全局变量或者静态全局变量；
* 如果开发了一个函数库或类库，将其放在一个名称空间中；
* 仅将编译指令using作为一种将旧代码转换为使用名称空间的权宜之计；
* 不要在头文件中使用using编译指令；
* 导入名称时，首选使用作用域解析运算符或using声明的方法；
* 对于using声明，首选将其作用域设置为局部而不是全局。

第十章：

1. 用户与数据交互的方式有三种：初始化、更新和报告。
2. 采用OOP方法时，首先从用户的角度考虑对象——描述对象所需的数据以及描述用户与数据交互所需的操作。完成对接口的描述后，需要确定如何实现接口和数据存储。最后，使用新的设计方案创建出程序。
3. 指定基本类型决定数据对象需要的内存数量、如何解释内存中的位和可使用数据对象执行的操作或方法。
4. 类规范由两个部分组成：

* 类声明：以数据成员的方式描述数据部分，以成员函数（被称为方法）的方式描述公有接口。
* 类方法定义：描述如何实现类成员函数。

1. C++程序将接口（类定义）放在头文件中，并将实现（类方法的代码）放在源代码文件中。
2. class定义类设计。
3. 要存储的数据以类数据成员的形式出现。要执行的操作以类函数成员的形式出现，成员函数可以就地定义也可以用原型表示。
4. 关键字private和public描述了对类成员的访问控制。使用类对象的程序都可以直接访问共有部分，但只能通过公有成员函数或友元函数来访问对象的私有成员。防止程序直接访问数据称为数据隐藏。不必在类声明中使用关键字private，因为这是类对象的默认访问控制。
5. 成员函数有函数头和函数体，也可以有返回类型和参数，它们还有两个特殊的特征：

* 定义成员函数时，使用作用域解析运算符（::）来标识函数所属的类；
* 类方法可以访问类的private组件。

1. 内联：定义位于类声明中的函数都将自动成为内敛函数，使用inline限定符将在类声明之外的定义的成员函数成为内联函数。
2. 通过成员运算符（.）使用对象的成员函数。
3. 构造函数的参数不是类成员，而是赋给类成员的值。因此参数名不能和类成员相同。常用方法是在数据成员名中使用m\_前缀，或者在成员名中使用后缀\_。
4. 使用构造函数初始化对象有显式和隐式调用两种见P354。
5. 析构函数完成清理工作。没有返回值和声明类型，也没有参数。
6. const成员函数的const关键字放在函数的后面。
7. this指针指向用来调用成员函数的对象，所有的类方法都将this指针设置为调用它的对象的地址。
8. 对象数组，声明方法和声明标准类型数组相同。可以用构造函数来初始化数组元素，此时必须为每个元素调用构造函数。标准格式是用括号括起的、以逗号分割的值列表。也可以对不同的元素使用不同的构造函数。
9. 类作用域：在类中定义的名称的作用域为整个类，只在该类中是已知的，在类外是不可知的。
10. 类作用域的常量：一是在类中声明一个枚举，二是使用关键字static。
11. ADT以通用的方式描述数据类型，类很适合描述ADT。