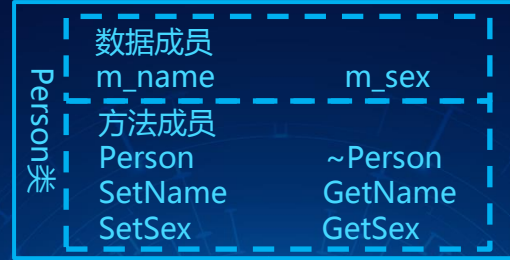


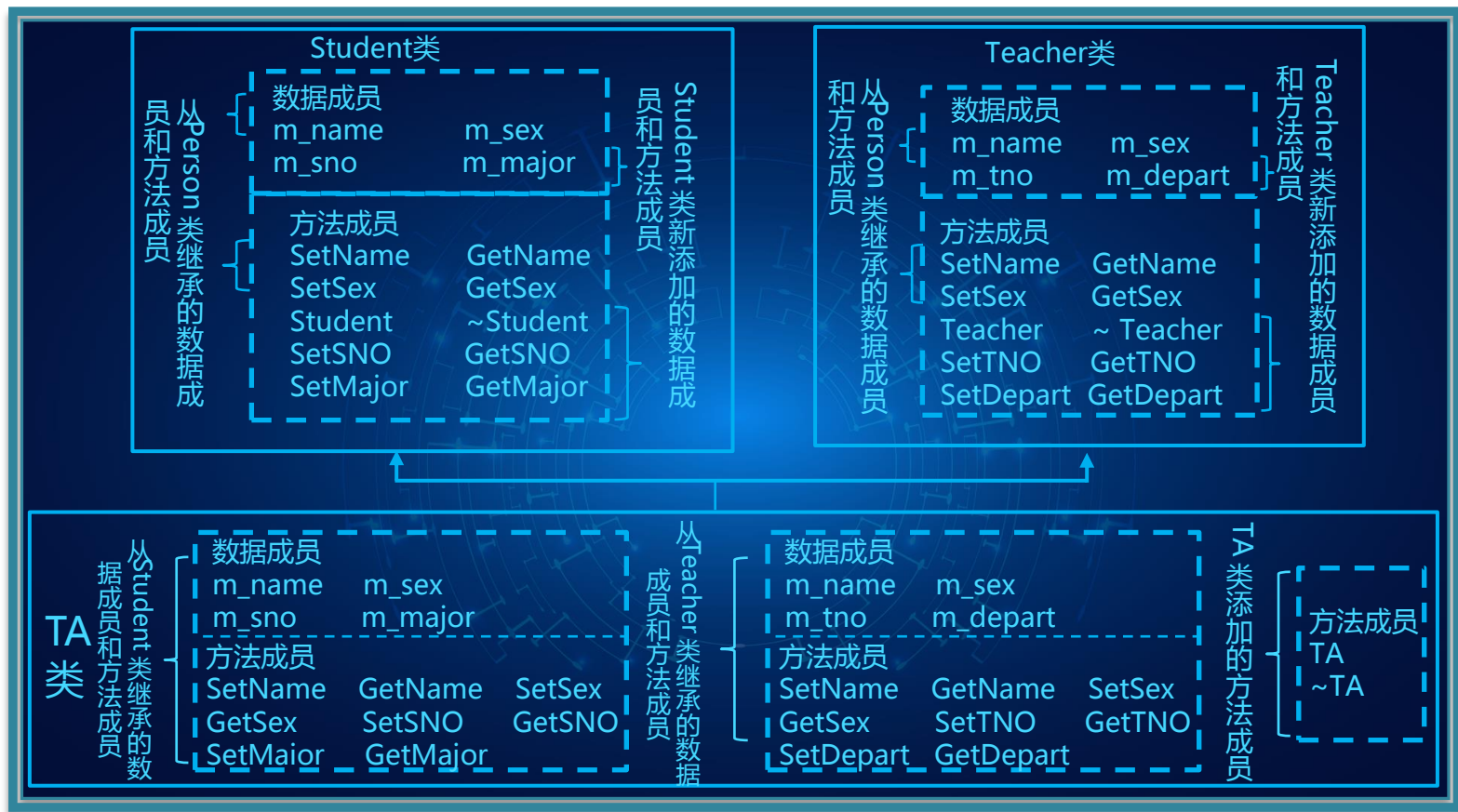
## 多重继承中的二义性问题

- 在例3-3中，Student类和Teacher类都继承了Person类的成员，TA类继承了Student类和Teacher类的成员，因此，TA中包含了两份Person类的成员（分别从Student类和Teacher类继承）。如果我们通过TA类的对象调用Person类的成员，则在编译程序时会报错。如：

```
cout<<ta.GetName()<<endl;    // 输出姓名
```

这是由于TA类中有两个分别从Student类和Teacher类继承过来的GetName()函数，直接使用ta对象调用GetName()函数时会有二义性问题，即编译程序不知道应该调用哪个GetName()函数。





为了解决这个问题，我们可以在调用GetName()函数时通过作用域运算符指定要调用从哪个类继承过来的函数。例如：

```
cout<<ta.Student::GetName()<<endl;
```

或

```
cout<<ta.Teacher::GetName()<<endl;
```

但这样调用函数需要知道类的继承关系，不方便类的使用。下一讲介绍另一种使用虚拟继承解决多重继承中二义性问题的方法。