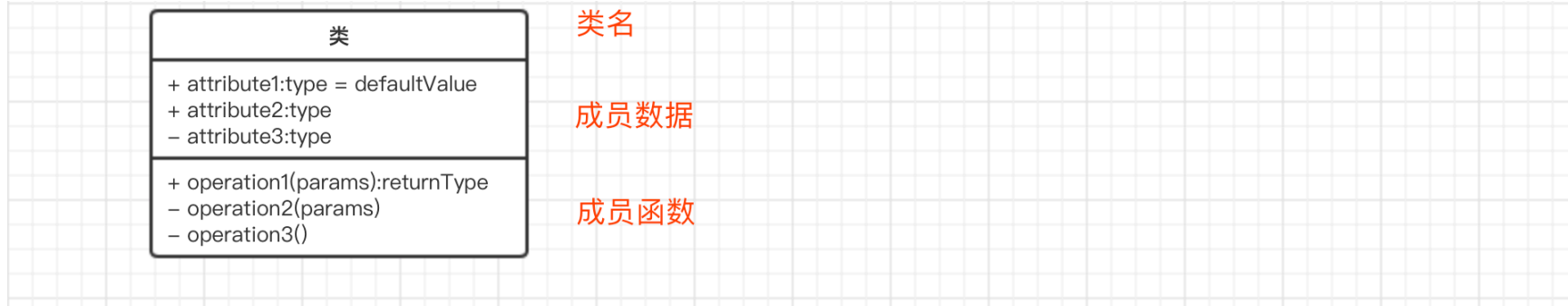


1 类图

1. 工具：

- <https://www.processon.com>

2. 类图的UML表示



3. 类图说明

- 区域说明：

- 类名：指定类名；
- 数据成员：定义成员变量；
- 函数成员：定义成员函数；

- 符号说明：

- + 表示public；
- - 表示private；
- # 表示protected；

- 成员说明：

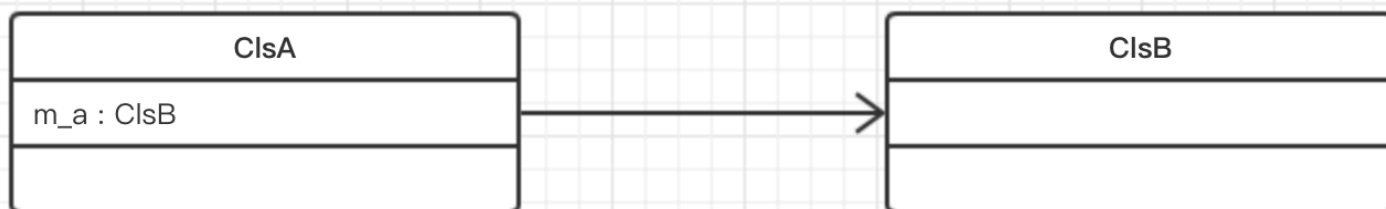
- 成员变量说明： 成员变量名： 类型 = 默认值
- 成员函数说明： 函数名(参数： 类型,): 返回类型

2 类关系

2.1 关联关系

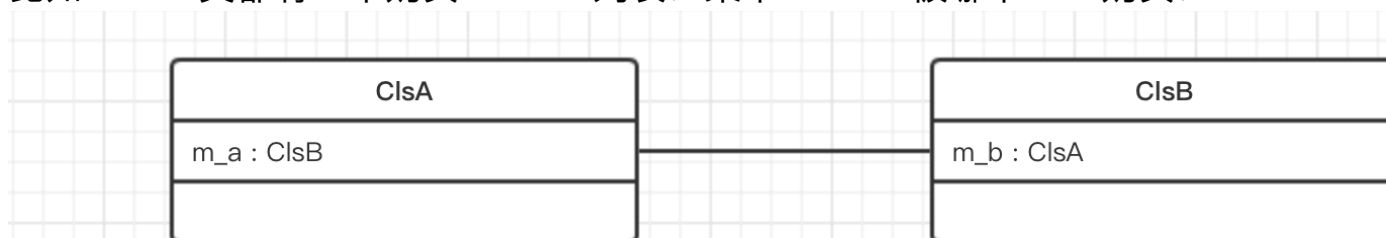
2.1.1 单向关联

- 一般用来描述实体关系
 - 实体ClsA 有 ClsB作为成员；
 - 比如：User类都有一个地址Address类型的数据；



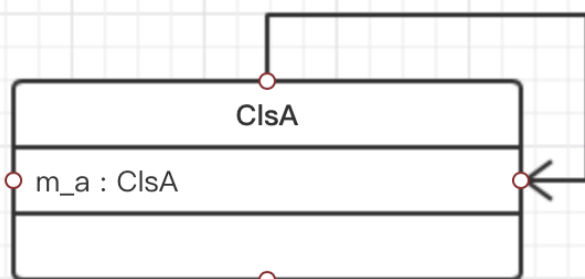
2.1.2 双向关联

- 一般用来描述实体类关系：
 - 表示双向的拥有关系；
 - 比如：User类都有一个购买Product列表；某个Product被哪个User购买；



2.1.3 自关联

- 一般用来描述实体；
 - 自己拥有自己；
 - 比如数据结构的链表节点，就是自己拥有自己；



2.2 依赖关系

- 一般用来描述调用关系
 - ClsA的call需要ClsB的called才能工作；
 - 一般是一个类的函数参数使用了另外一个类型的对象；



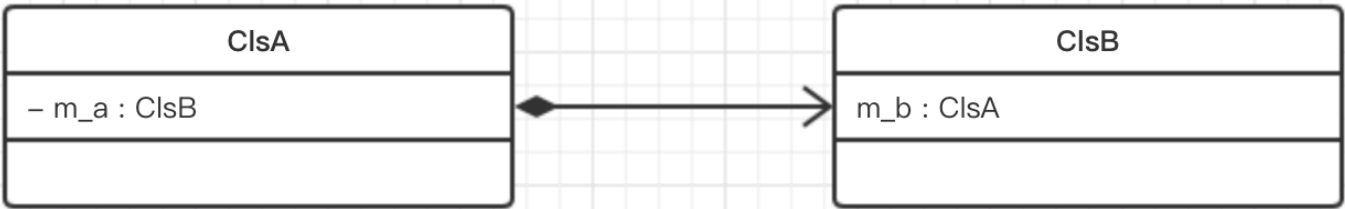
2.3 聚合关系

- 一般用来描述调用关系
 - ClsA的实现需要ClsB；
 - 并且ClsB作为ClsA的成员变量，但是ClsB对象的构建不在ClsA中构建，ClsB对象可以独立存在，不会随ClsA对象的释放而释放。



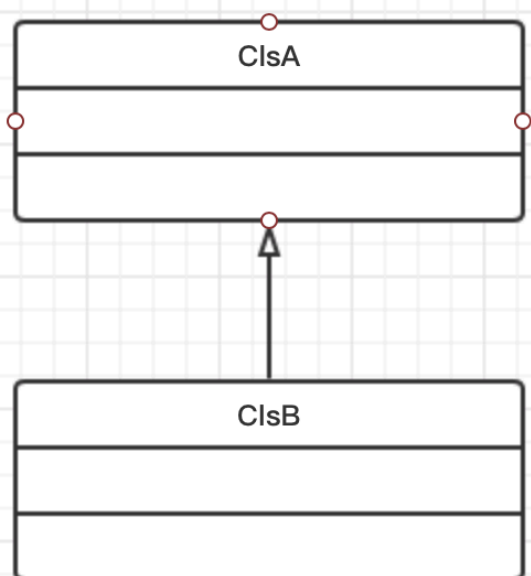
2.4 组合关系

- 一般用来描述调用关系
 - ClsA的实现需要ClsB；
 - 并且ClsB作为ClsA的成员变量，ClsB对象的构建在ClsA中构建；
 - ClsB对象不是独立存在的，ClsA对象释放时，ClsB对象也会被释放。



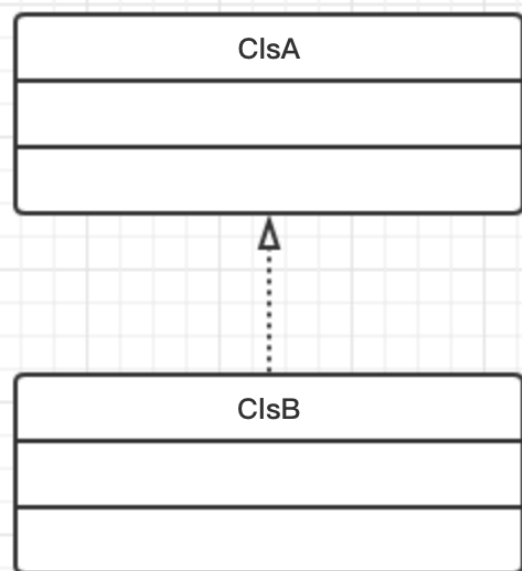
2.5 继承关系

- 继承关系是面向对象中最基本的关系了；



2.6 接口继承关系

- 这个关系在Python非常淡化，因为Python在语法上是弱类型语言（因为类型是动态判定的）；
 - 但是在Python从编程技巧上，可以实现与接口类似的程序结构。

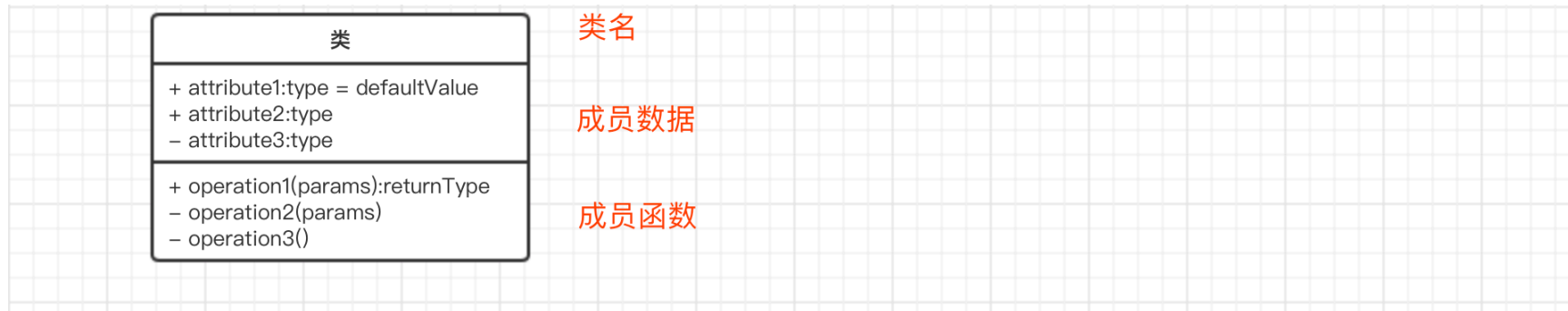


3 类图

1. 工具：

- <https://www.processon.com>

2. 类图的UML表示



3. 类图说明

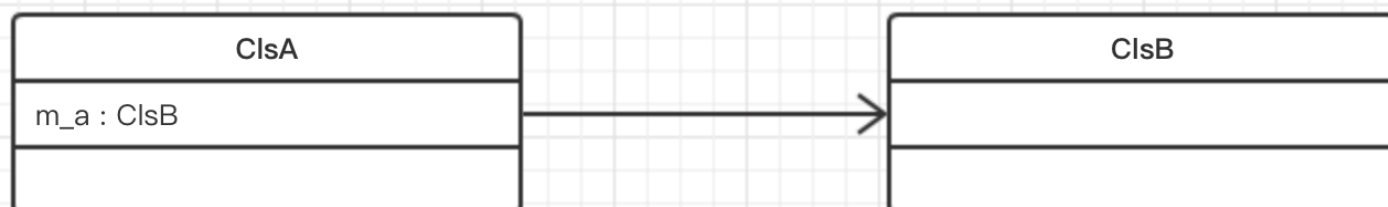
- 区域说明：
 - 类名：指定类名；
 - 数据成员：定义成员变量；
 - 函数成员：定义成员函数；
- 符号说明：
 - + 表示public；
 - 表示private；
 - # 表示protected；
- 成员说明：
 - 成员变量说明： 成员变量名： 类型 = 默认值
 - 成员函数说明： 函数名(参数： 类型，): 返回类型

4 类关系

4.1 关联关系

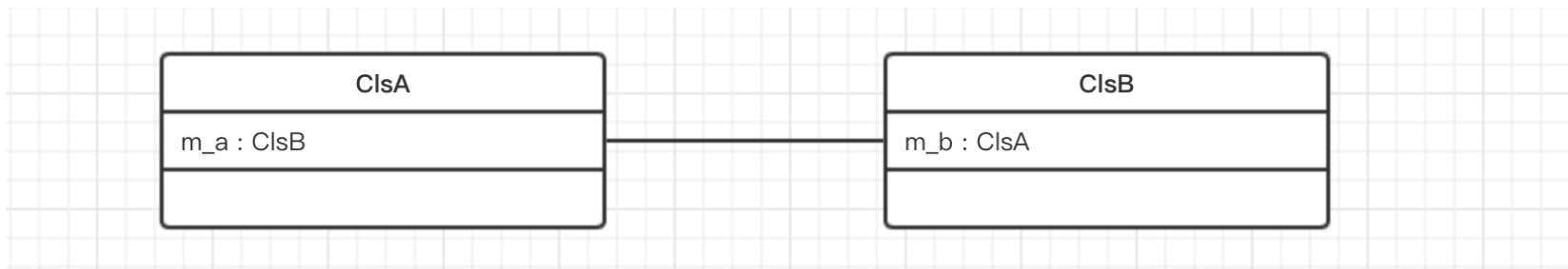
4.1.1 单向关联

- 一般用来描述实体关系
 - 实体ClsA 有 ClsB作为成员；
 - 比如： User类都有一个地址Address类型的数据；



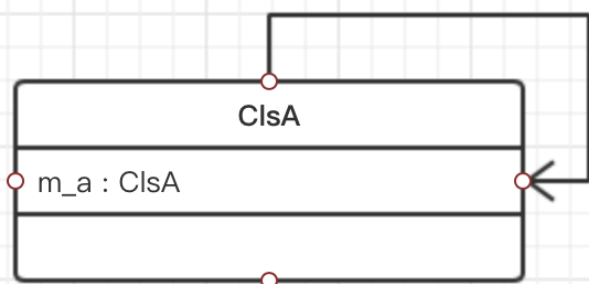
4.1.2 双向关联

- 一般用来描述实体类关系：
 - 表示双向的拥有关系；
 - 比如：User类都有一个购买Product列表；某个Product被哪个User购买；



4.1.3 自关联

- 一般用来描述实体；
 - 自己拥有自己；
 - 比如数据结构的链表节点，就是自己拥有自己；



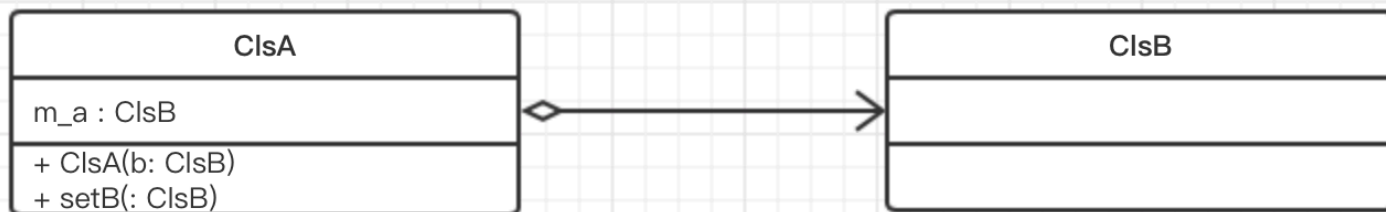
4.2 依赖关系

- 一般用来描述调用关系
 - ClsA的call需要ClsB的called才能工作；
 - 一般是一个类的函数参数使用了另外一个类型的对象；



4.3 聚合关系

- 一般用来描述调用关系
 - ClsA的实现需要ClsB；
 - 并且ClsB作为ClsA的成员变量，但是ClsB对象的构建不在ClsA中构建，ClsB对象可以独立存在，不会随ClsA对象的释放而释放。



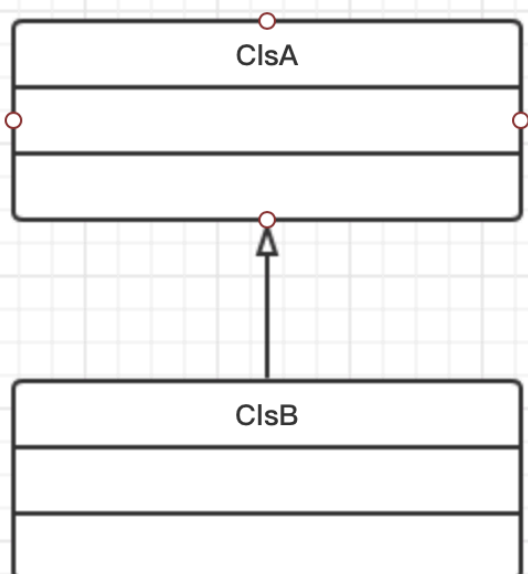
4.4 组合关系

- 一般用来描述调用关系
 - ClsA的实现需要ClsB；
 - 并且ClsB作为ClsA的成员变量，ClsB对象的构建在ClsA中构建；
 - ClsB对象不是独立存在的，ClsA对象释放时，ClsB对象也会被释放。



4.5 继承关系

- 继承关系是面向对象中最基本的关系了；



4.6 接口继承关系

- 这个关系在Python非常淡化，因为Python在语法上是弱类型语言（因为类型是动态判定的）；
 - 但是在Python从编程技巧上，可以实现与接口类似的程序结构。

