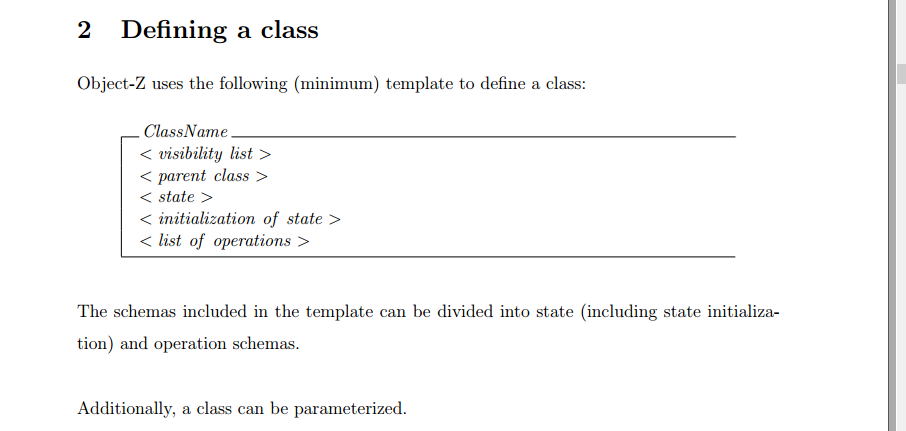
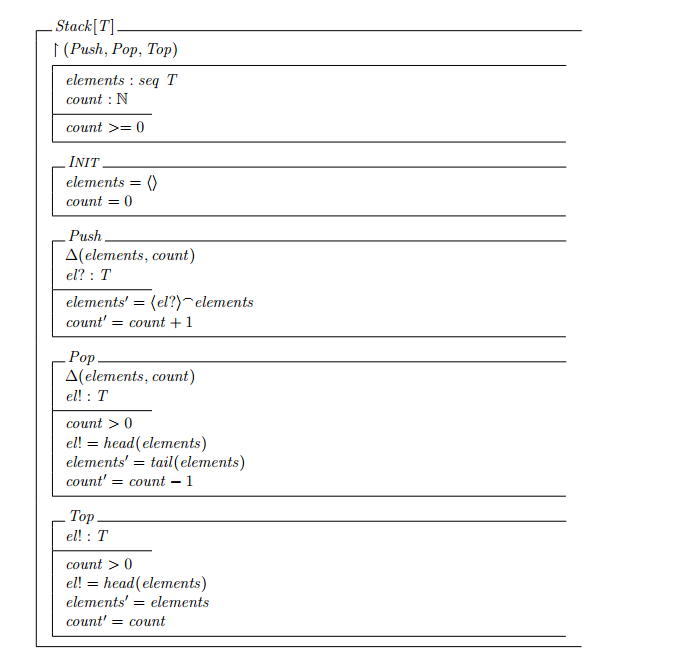
Object z是 Z的一个延伸，这样就可以支撑面向对象规则specification



OBJECTZ用这个格式来定义一个class

这个格式里的schema可以分为state(包括初始state)schema以及operation schema操作schema



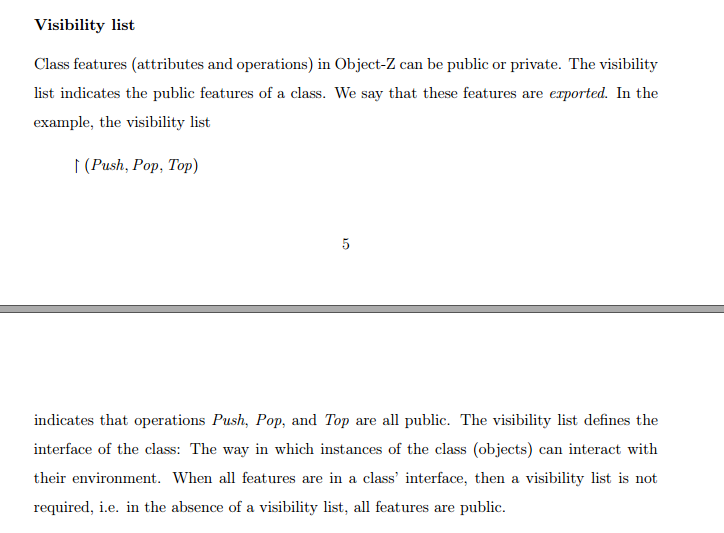
VISIBILITY LIST

CLASS 的attribute与operation可以是public或private的，

visibility list的目的就是列出所有的public操作与变量（public features），我们说这些features是exported

介绍了三种public的操作，

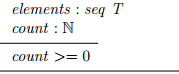
visibility list实际上描述了这个class的接口interface。当这个class所有的feature都是public，那其实没必要写visibility list，默认就是全public

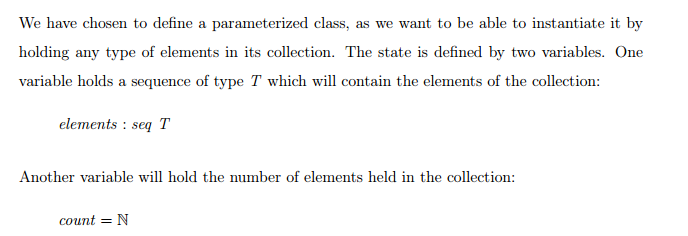


Parent Class

这个例子里面，stack并没继承任何别的class

State





State描述的是这个class需要的变量

elements :seq T 包含了这个collection中的元素

count记录了有几个元素

注意，在这一步并不赋值，再Initial的时候赋值,我们这里只是阐述并加上限制（count>=0）

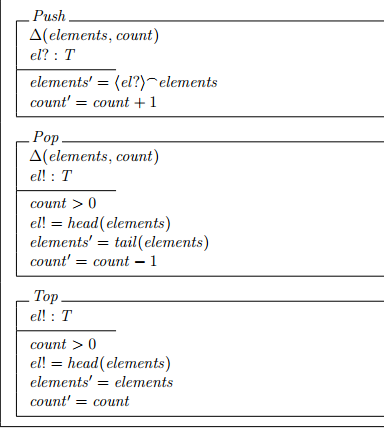
initial



list of operation

state 中所描述的变量element与count是任何Opeeration都可以接触的，他们可以使用





每个操作可能需要用到一个list，三角来表示

push没有先决条件，因为是UNbound collection

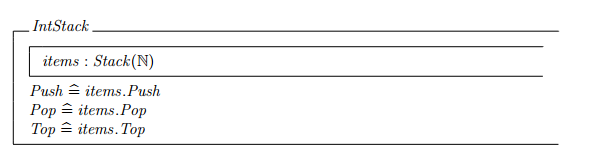
输入一个东西，用连接符加到头上，count+1

pop先决条件就是count>0,，注意pop是必须提示pop出来什么东西的，pophead，element变为原来的tail，count—

TOP，element count不变，只是告诉你top是啥

实例化 Stack,

intstack，用整数来实例化他



state:

变量，所有的item都是stack类型用整数来实例化



操作就是items.XX,

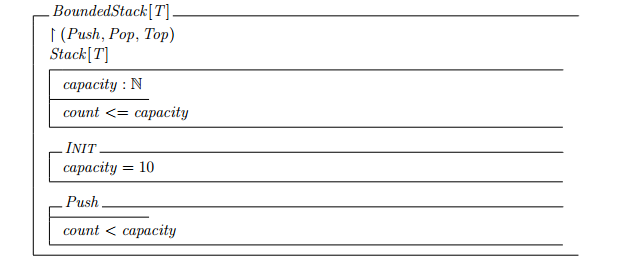
继承

class可以继承另外一个class，继承一切，除了visibility list(决定可视性)，原先public的东西可以改成private，原先private的东西可以改成Public,

state:

子类的state里定义的variable可以覆盖父类的variable，但是variable必须是兼容的，比如父类的变量number类型是int,我们可以限制他，让他直接收正整数

如果在子类中重新定义了一个operation，那么父类中operation的声明将与子类中相同操作的声明合并（横线上面，既只使用主类，不能改变·）。操作的predicate部分与主类共用（在主类的基础上写）

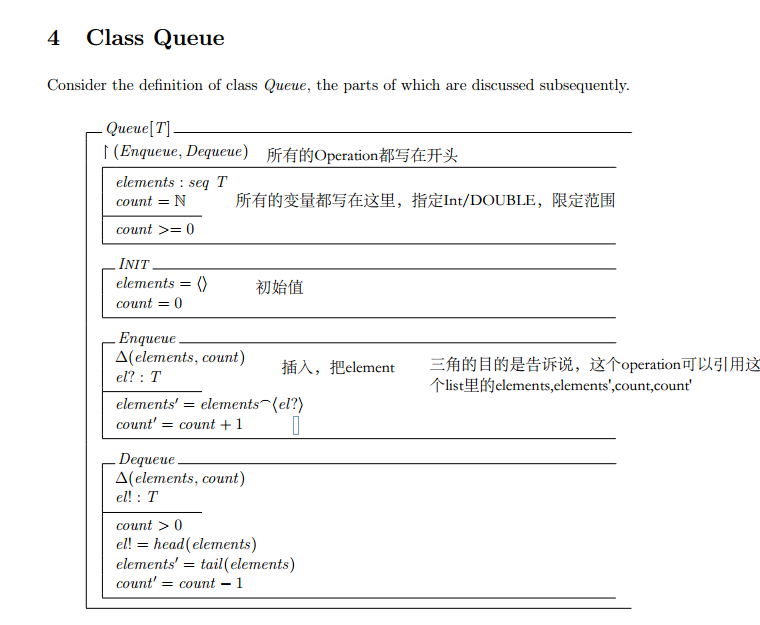


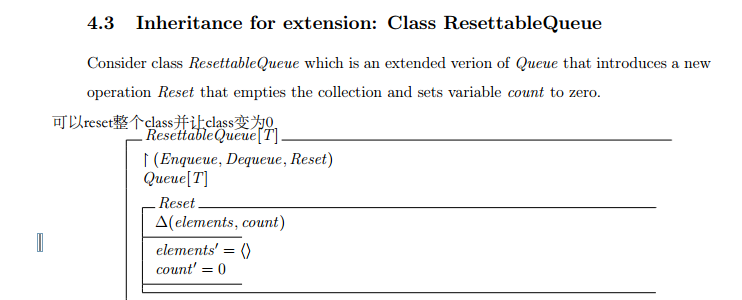
bound stack，注意，要重新写capacity list

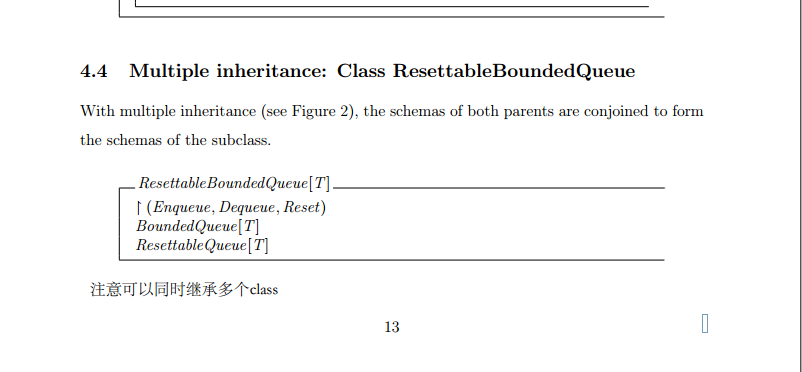
继承类Stack[T]

state里面加入新变量capacity，并且给原来的count加上限制

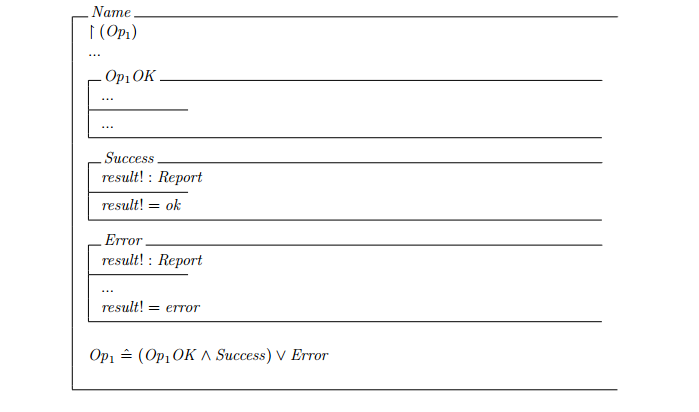
push加上新限制



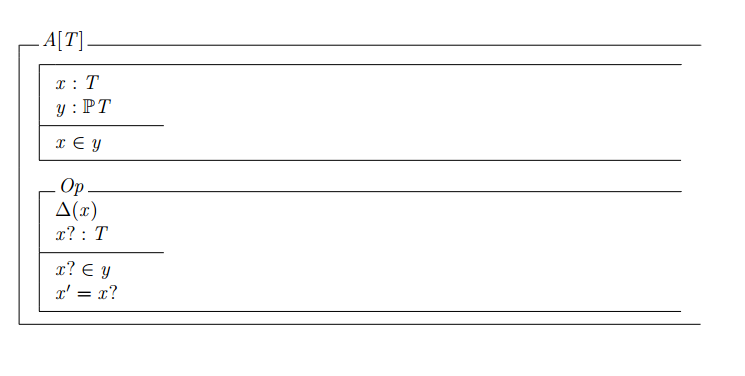




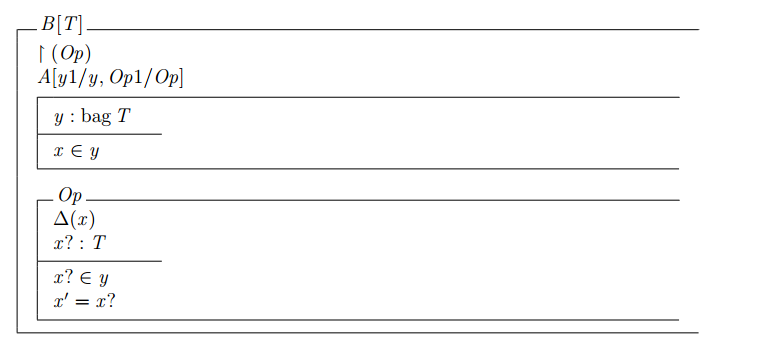
怎么控制error



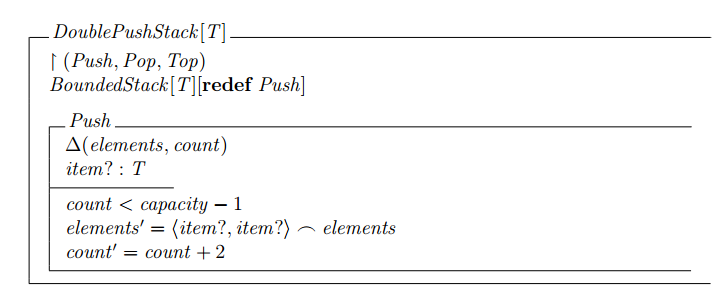
成功交success，失败并error，



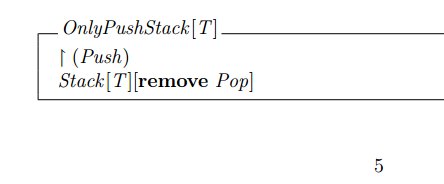
原来A[T]



继承B【T】，在继承这一步，我们可以把y改名成y1,Op改编成OP1，因为OP1与Y1都没被囊括在VISIBILITY LIST 里，就相当于删除了原来的y与OP，然后我们就可以完全重新定义Y与OP



我们可以指定要重新描述哪个operation



或者remove