Main里面放了计时器，模型里的1day是我们的一周。所以每经过1day，Main里面的day变量会+7。其它统计时间的会跟着变。学期变了会给很多智能体发upgrade消息。

Library里有所有书Book，Book有属性存量，每本书有一个BookInstance实体。Book一对多BookInstance。

Customers从数据库加载的，年级、专业是等分。Customer的“相关度rel”和他们的借阅频率、时长的期望值有关系。

Customer有很多Thought，一门课诞生一个Thought，Thought里存着这门课的BookWithWeights(教材池)。

应该有个包含所有书，和课程无关的Thought，但是太大了放进来模型会卡死，我偷偷删了。

Thought会根据学分占比，把之前“相关度rel”的期望频率，分出属于自己的频率。当作Thought自己的期望频率。然后Thought上半部分一大堆连线的系统动力学变量是拿来跨学期变异这个期望频率的，得到每个学期的期望频率。状态图是拿来变异学期内的期望频率，得到每周的期望频率(引用了父，customer里面的两个tablefunction，为什么不每个thought放一个？因为我怕太卡了，而且大家都是一样的)

Thought会把频率当概率，尝试发送一个借书请求，给customer。

Customer汇集了所有thought的借书请求，自己根据bookKeeping(当前持有的教材)，生成今天去图书馆会还的书、会办理续借的书，然后给图书馆发请求。

图书馆收到请求之后会交给teller处理，入口在service组件的action里面，写了一行代码。

会把所有请求(一个列表)丢到teller的待处理请求里。后面就是teller一个个拿出来把请求解析了生成响应Respond(类名写成动词了)，反正它是响应体的意思，会发还给customer。

具体怎么解析请求的，就是teller那个很大的状态图，包含了fdu借阅规则，借+预约+还+续借，包括查库存，算替代(所以想借书是一整个教材池子丢过去，在里面选)，计算过期时间，一站式服务。

怎么处理响应，借阅成功之类的之后会发生什么？

一部分，影响bookKeeping、Library、和customer有关的port的连接/断开，这些在Book智能体的方法里面处理。

其他的，影响customer的子智能体，就是Thought的，在customer的函数里处理，其中预约的回调是空的，因为我想不到写啥。

借阅成功的回调是暂时减少教材在所有教材池子的权重(把流出的速度的系数keeping从0变成1，开始流式权重 )(顺带一提教材的weight是用系统动力学的stock实现的，减少就是让它快速流走，过一段时间慢慢流回来，也有一部分永久消失了)；还有就是把相关的课程(整个Thought)的期望频率下调了，这个是永久的，每借一次期望前面的系数乘0.5，可能有点太狠了；另外的，如果是选课学期前减少的，进入选课学期系数会马上恢复成1哦。

还书成功的回调是把刚刚那个keeping变回1，此后会慢慢流回来但不是全部，具体看bookwithweight的系统动力学。

还有一个没提，customer具体如何生成续借/还书请求的？在bookkeeping类里，还书会参考期望时长和已持有时长，用正态分布(已持有>期望附近的随机数)看今天要不要还这本书，具体看代码；

续借会参考到期剩余时长，越临近的越容易拿去续借，也是在代码里有写。

如果一本书同时被 续借、还书 抽到了，那么优先还书(有一个优先但我记不清了，代码里removeAll括号里的优先应该是)。

最后一点，学生在假期离校的时候会把所有书还掉。