



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

软件工程

## Lab 7: OO分析与设计

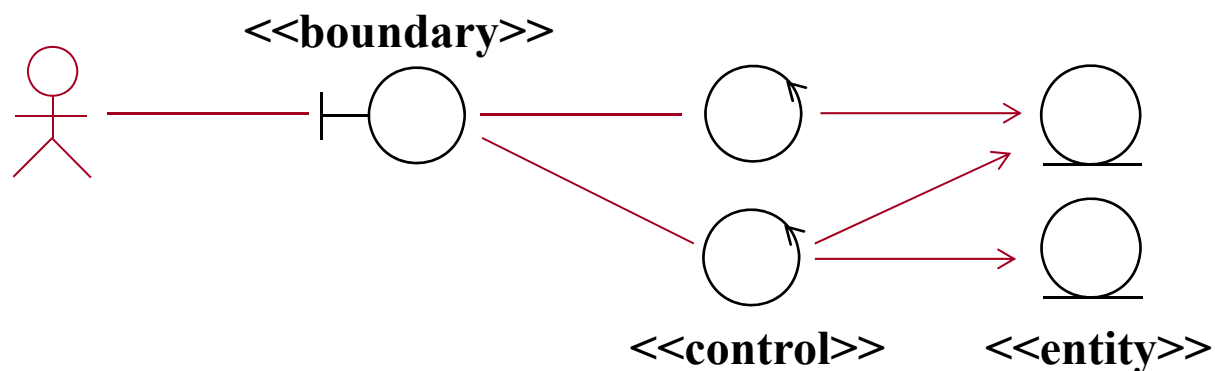


# 实验目标

- 2人一组完成本实验，与Lab1、Lab4、Lab6的分组保持一致。
- 目标：针对Lab6之后的Lab1代码，使用OO思想对其进行重构
- OO模型：
  - 边界类、控制类、实体类
  - 分析类图
  - 概念模型(实体类图)(class diagram)——此为重点
  - 时序模型(sequence diagram)——此为重点
- 使用UML建模工具绘制以上OO模型；
- 根据OO模型，对Lab1原代码进行重构。

# 实验过程

- **Step 1:** 针对Lab1, Lab4, Lab6的需求, 识别边界类、控制类、实体类, 将UI、控制逻辑、数据处理逻辑清晰的分开, 保持逻辑独立, 形成独立的类;
  - 实体类: 表示系统存储和管理的信息 (属性、操作)
  - 边界类: 表示参与者与系统之间的交互
  - 控制类: 表示系统在运行过程中的业务控制逻辑



# 实验过程

## ■ Step 2: 设计各边界类、控制类、实体类的属性。

### — 边界类:

- 需要actor输入的各数据;
- 系统反馈给actor的各数据;
- 需要临时保存的、用于在边界类和控制类之间传递的临时数据;

### — 控制类:

- 从UI接收的数据;
- 为进行事件流执行所需的临时数据;
- 需要调用的实体类;
- 经过计算之后、需要发送给UI的数据。

### — 实体类

- 基本属性
- 状态属性: 识别对象需要区别的状态, 考虑是否增加属性来区别这些状态;
- 关联属性: 确定属性表示整体与部分结构和实例连接;
- 派生属性: 通过计算其他属性的值所得到的新属性。

# 实验过程

- **Step 3: 设计各边界类、控制类、实体类的方法。**
  - 边界类的操作：
    - 提供给用户的、可在UI上进行的各类操作；
    - 对从控制类返回的数据进行各类临时处理而进行的操作；
    - 提供给其他系统的API；
  - 控制类的操作：
    - 对从边界类接收到的数据进行各类临时处理而进行的操作；
    - 向实体类所发出的调用操作；
    - 对从实体类接收到的数据进行临时处理而进行的操作；
  - 实体类的操作：
    - 对属性进行CRUD的操作；
    - 对状态进行更新的操作；
    - 辅助操作。
  - 对各方法设计其内部的业务逻辑；

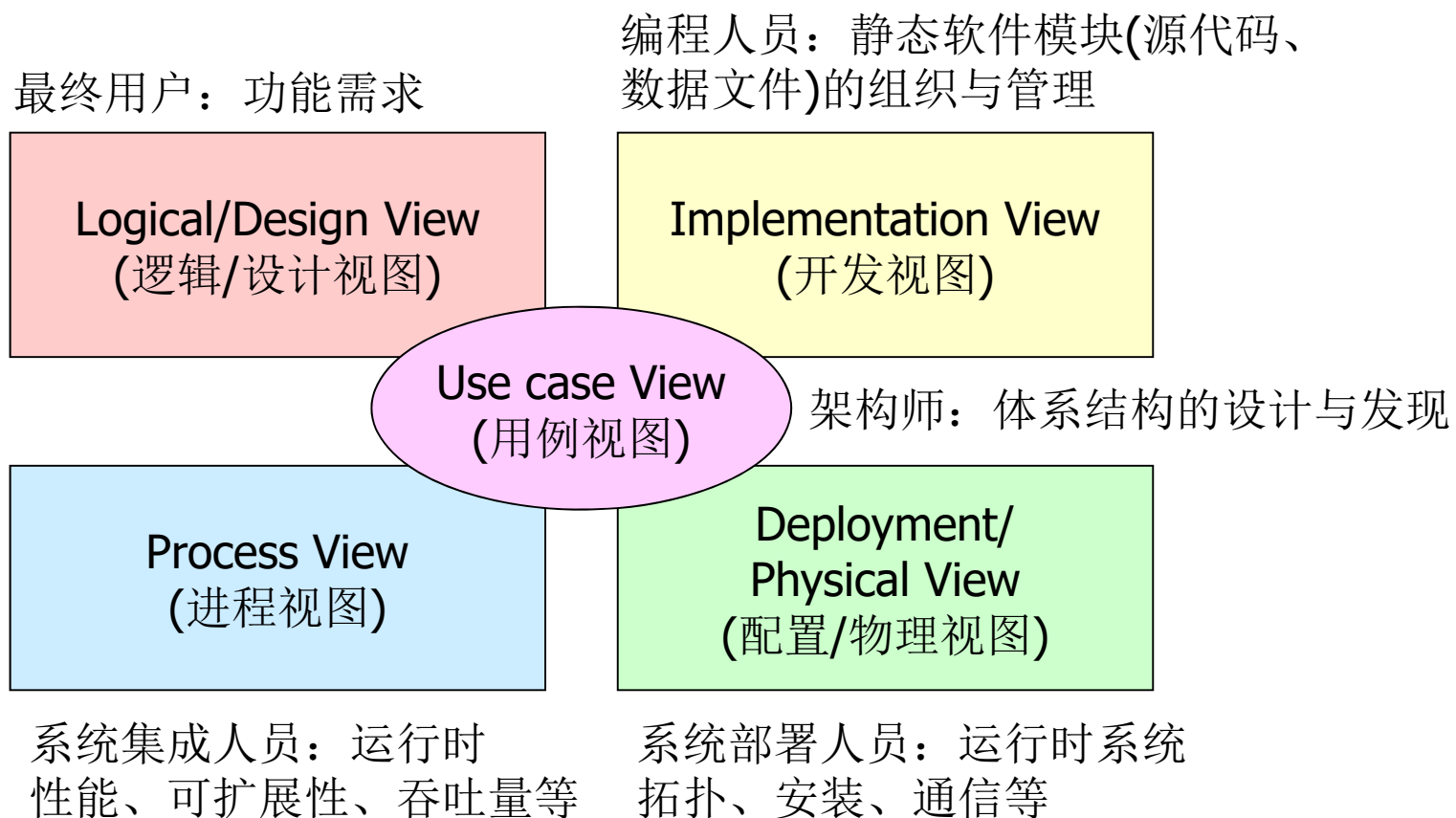
# 实验过程

- Step 4: 建立分析类图，描述三种分析类之间的逻辑关系；
- Step 5: 建立领域类图，描述实体类之间的五种关系 (继承、关联、组合、聚合、依赖)
- Step 6: 建立时序模型，描述三种分析类之间的消息流和调用关系。

# 实验过程

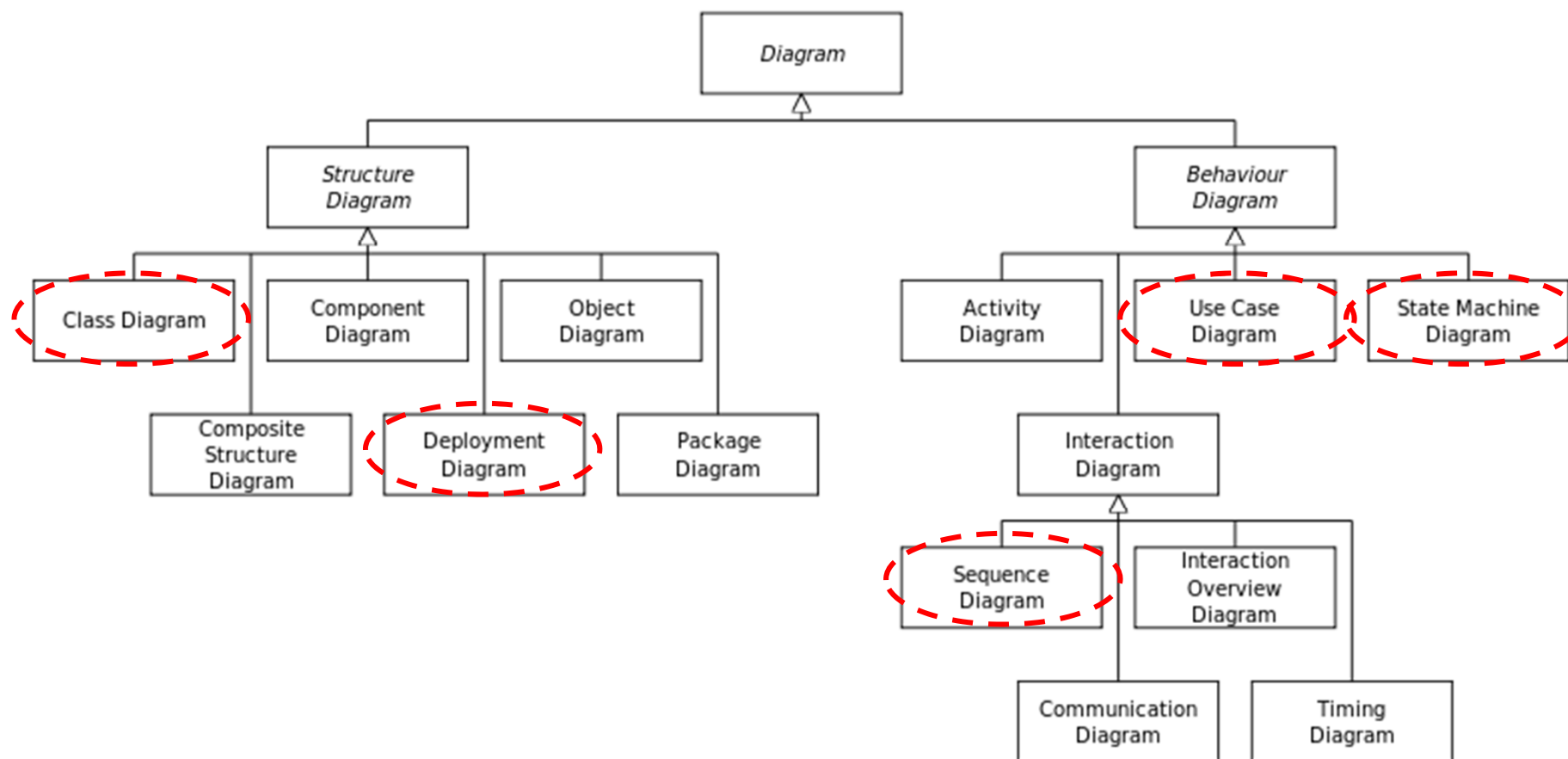
- **Step 7:** 使用StarUML (本地安装程序)或GenMyModel(Web网站)或UML Designer(Eclipse插件)建立以上模型：分析类图、领域类图、时序模型。
- **Step 8:** 重构Lab6的代码。该过程中可能要回头修改各OO模型，不断迭代；
- **Step 9:** 基于Lab6中设计和实现的各黑盒/白盒测试用例，进行回归测试，发现重构代码中存在的问题，并加以修复。
- **Step 10:** 在Git仓库内建立Lab7分支，提交重构后的代码，push到GitHub仓库。

# (Kruchten) 4+1视图模型结构



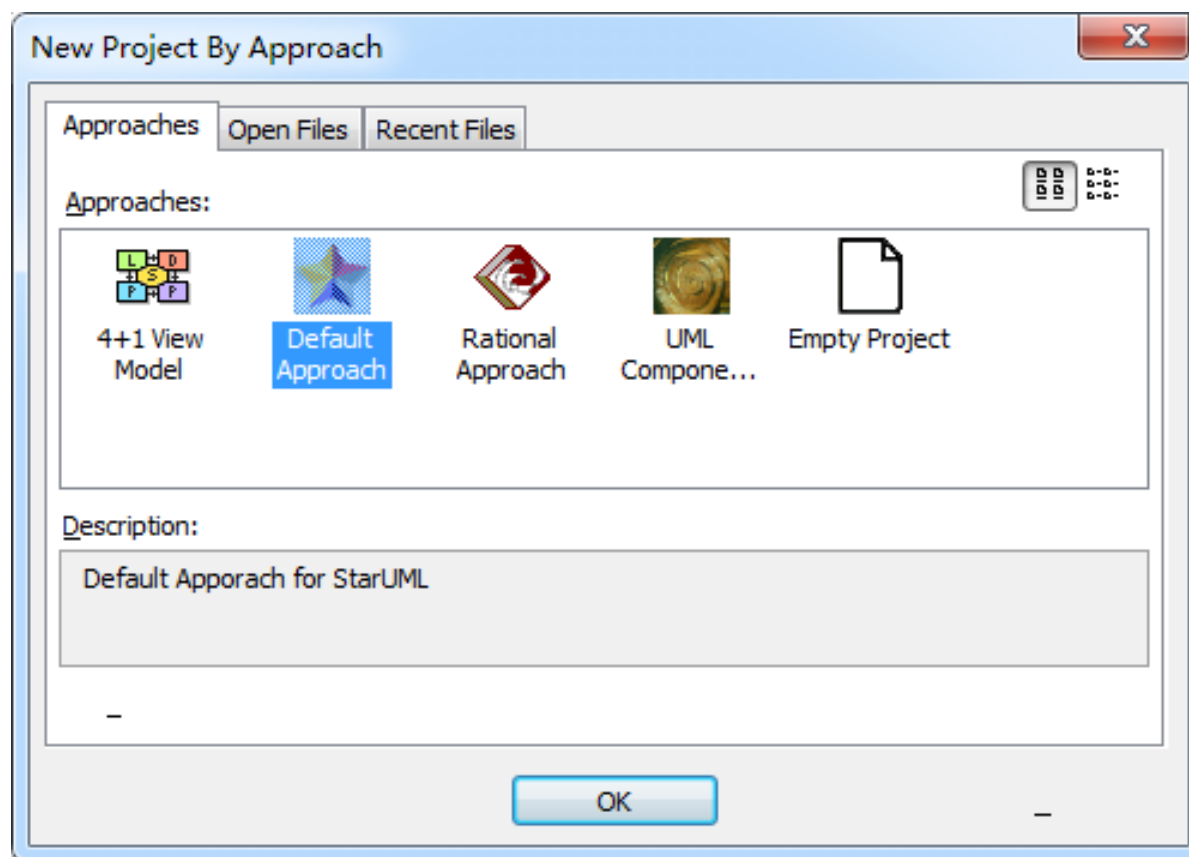


# UML模型



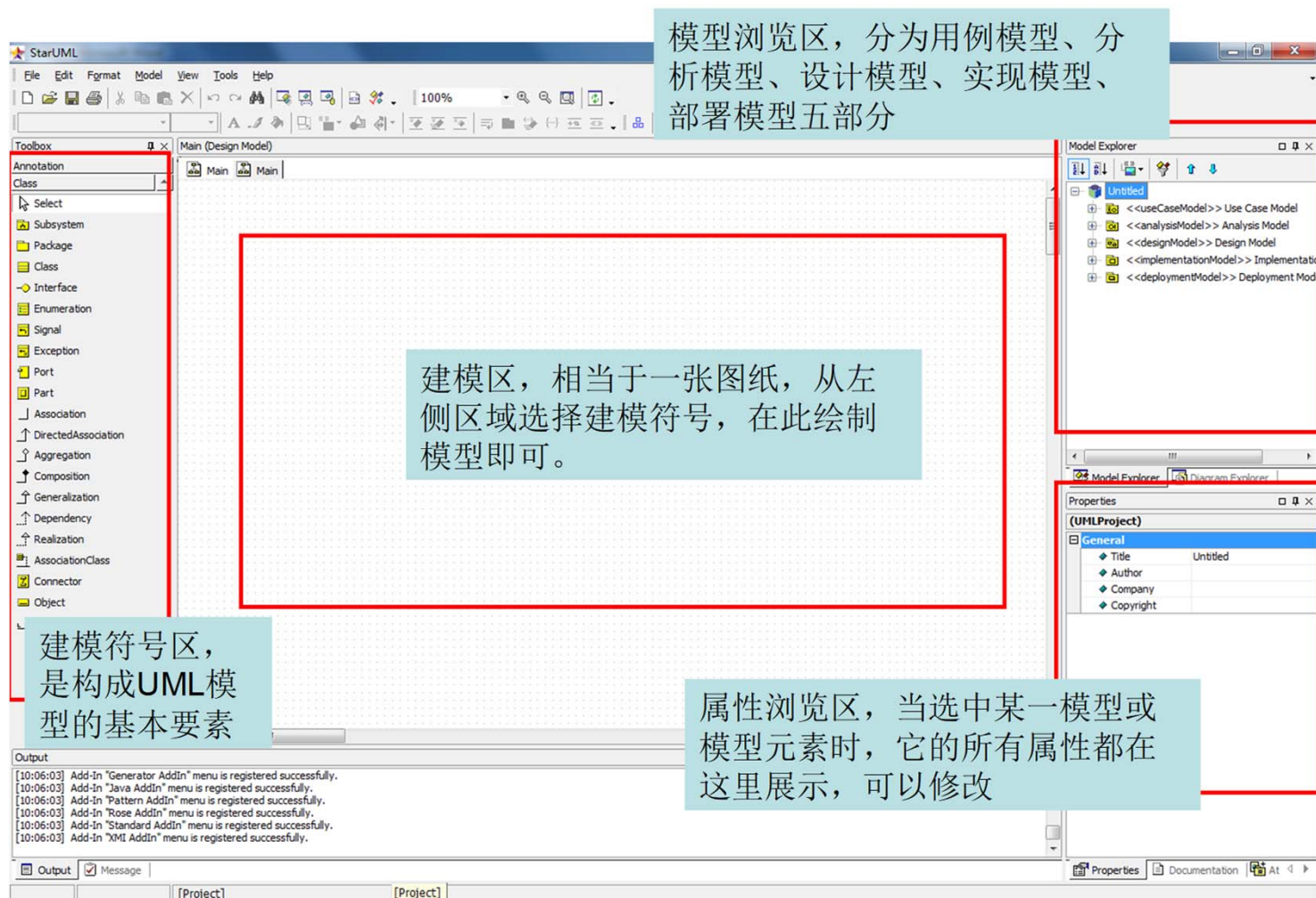
# StarUML建模指南

- 安装StarUML;
- 启动, 建立project, 选择default



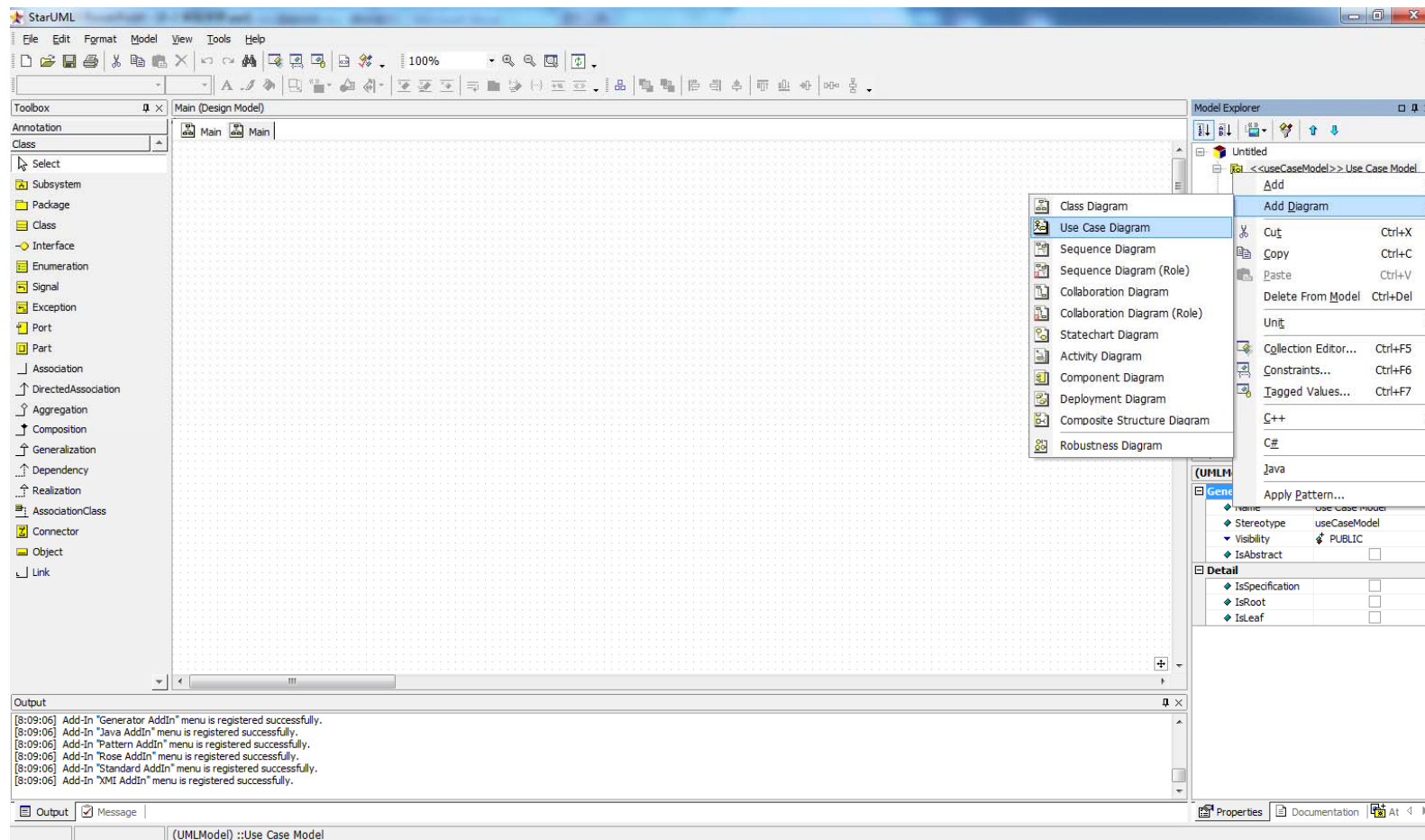
# StarUML建模指南

## ■ 进入主界面



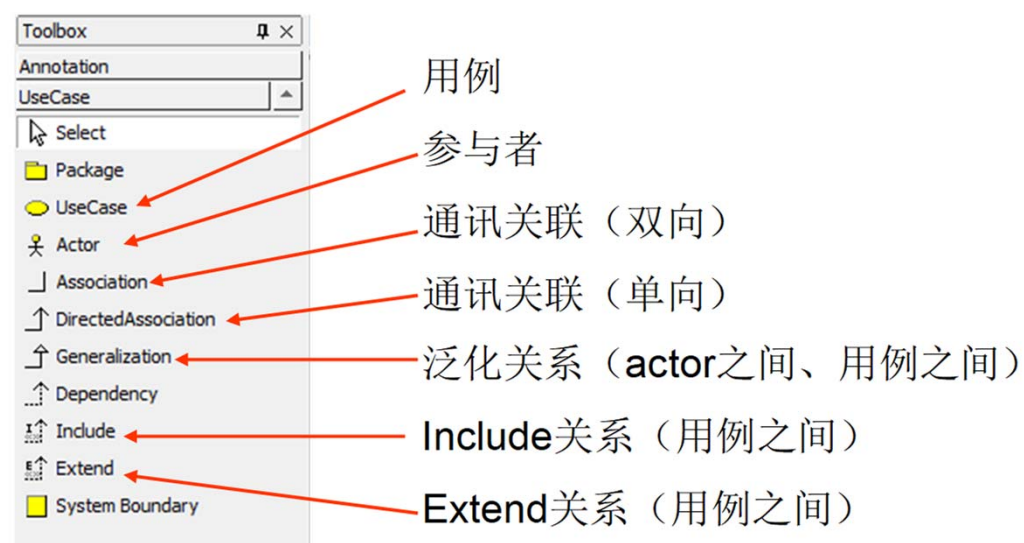
# StarUML建模指南

- 用例模型。在模型浏览区的<<useCaseModel>>树节点上点击右键，选择Add Diagram、Use Case Diagram，并为新建立的图命名。



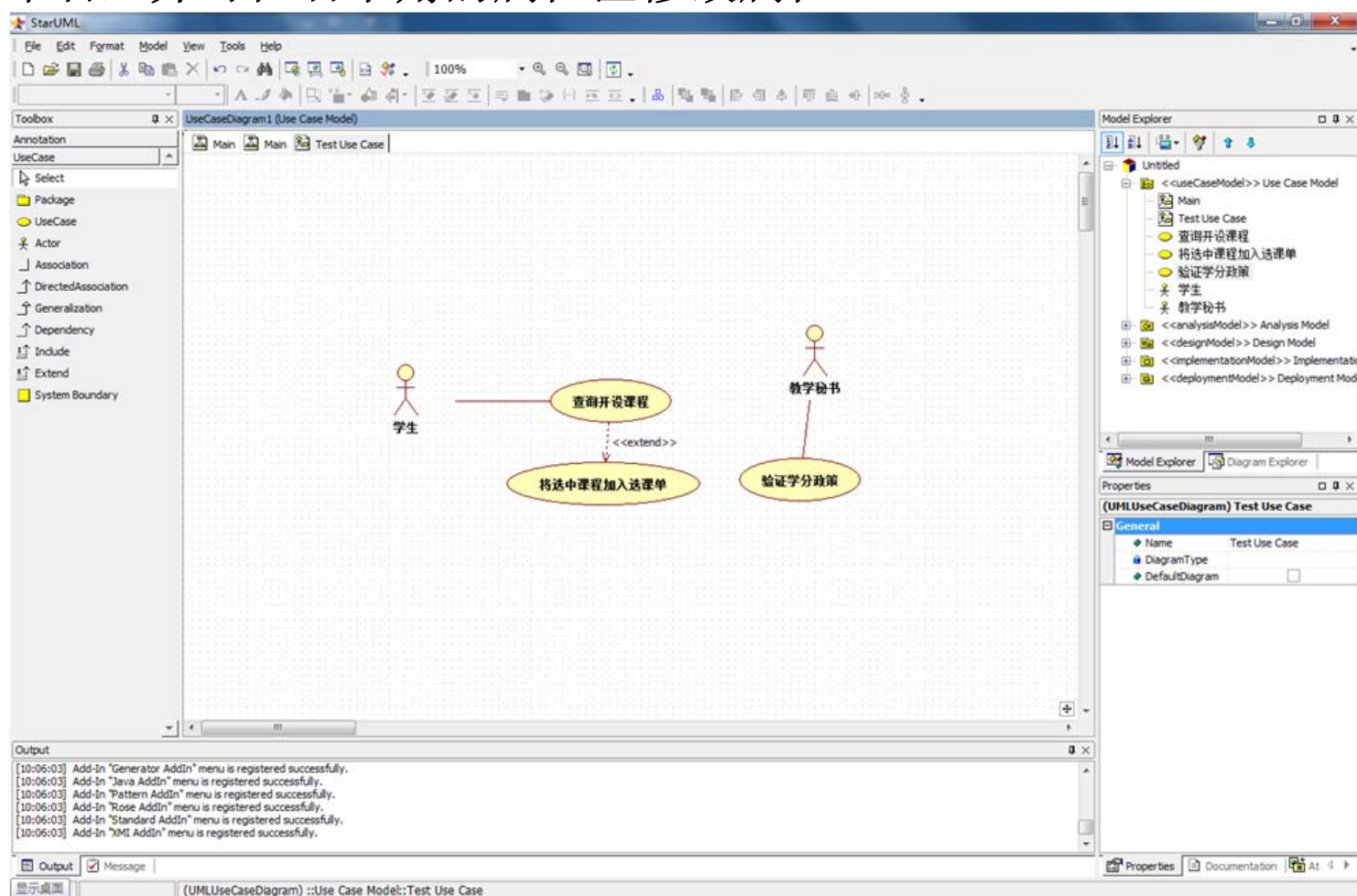
# StarUML建模指南

- 左侧建模符号区展现了用例模型的基本要素



# StarUML建模指南

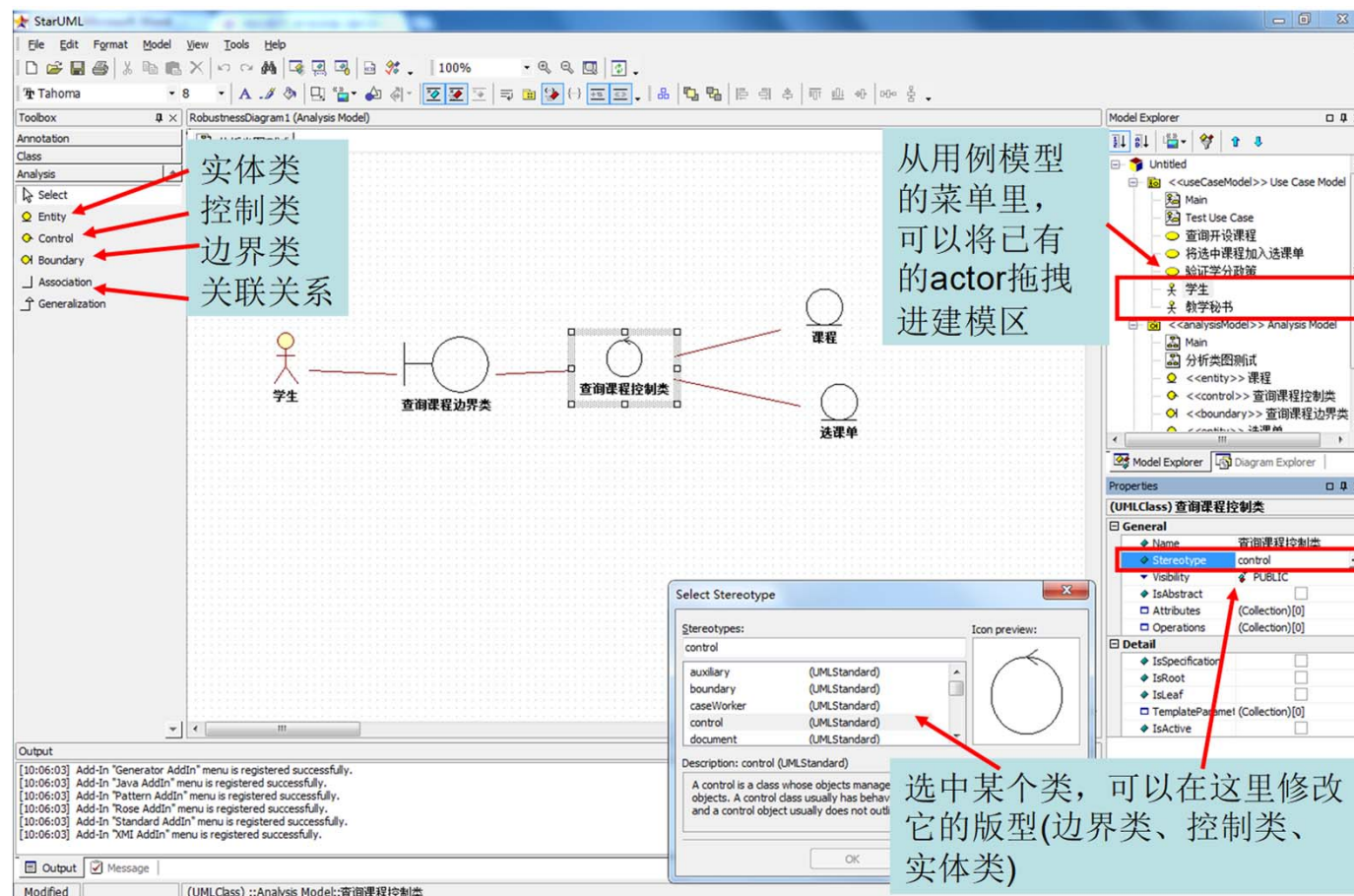
- 选中某种建模符号，在绘图区单击，即可建立相应的模型要素。对其进行命名，并可在右下角的属性区修改属性。





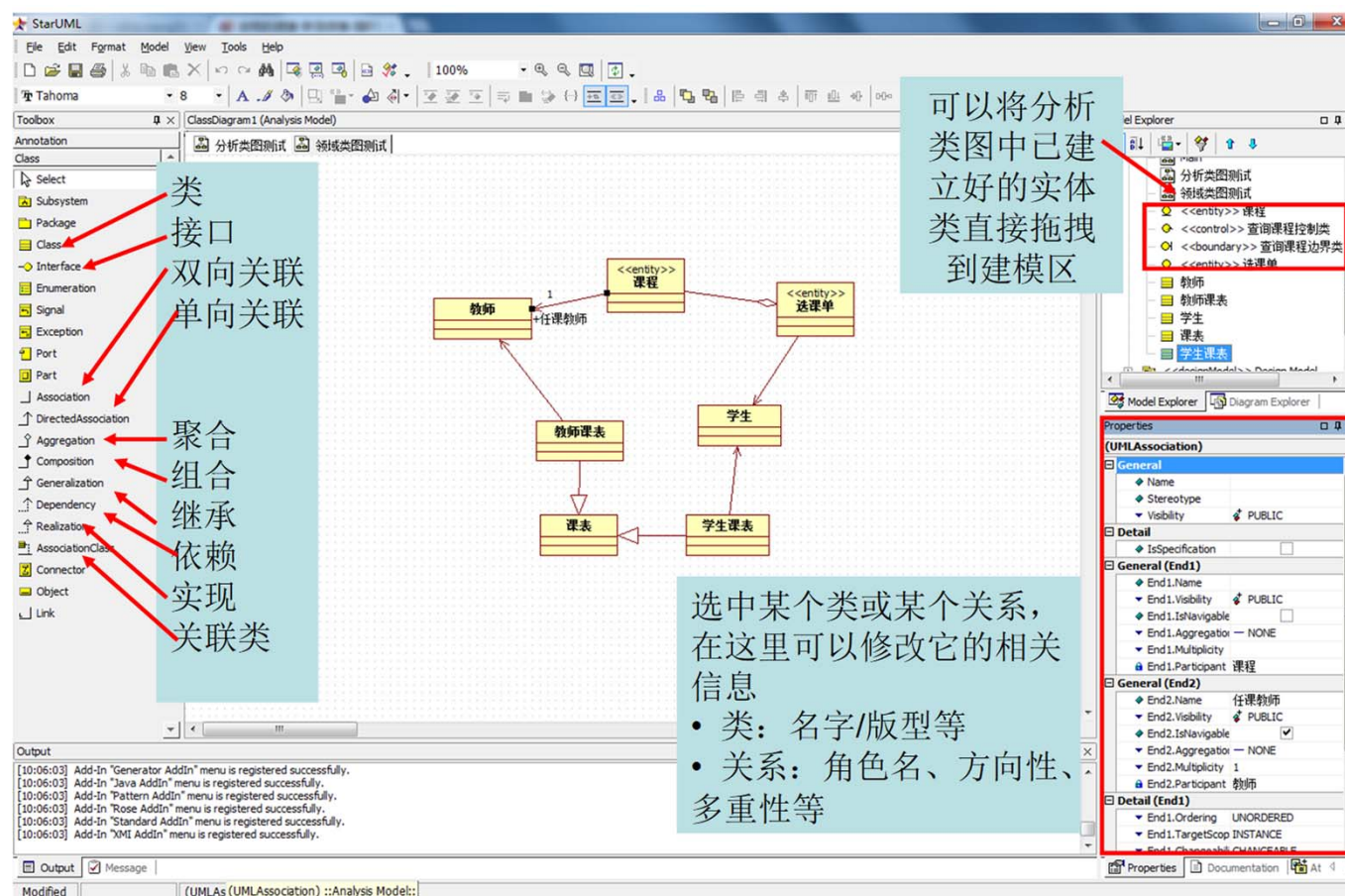
# StarUML建模指南

- 建立分析类图。在模型浏览区的<<analysisModel>>节点上点击右键，选择Add Diagram、Robustness Diagram，并为新建立的图形命名。此时左侧符号区展示了分析类图的要素。



# StarUML建模指南

- 建立领域类图。在模型浏览区的<<analysisModel>>节点上点击右键，选择Add Diagram、Class Diagram，并为新建立的图形命名。此时左侧符号区展示了领域类图的要素。





# StarUML建模指南

## ■ 建立类的属性和操作

The image shows the StarUML software interface with a class diagram and an open 'Collection Editor' window. The class diagram includes classes like '教师' (Teacher), '课程' (Course), '选课单' (Course Selection Form), '教师课表' (Teacher Schedule), and '学生' (Student). The 'Collection Editor' window is open for the '课程' class, showing attributes '课号' (Course ID) and '课程名' (Course Name). The 'General' tab in the Properties window on the right shows the 'Attributes' section with a red box highlighting the '...' button next to the attribute list.

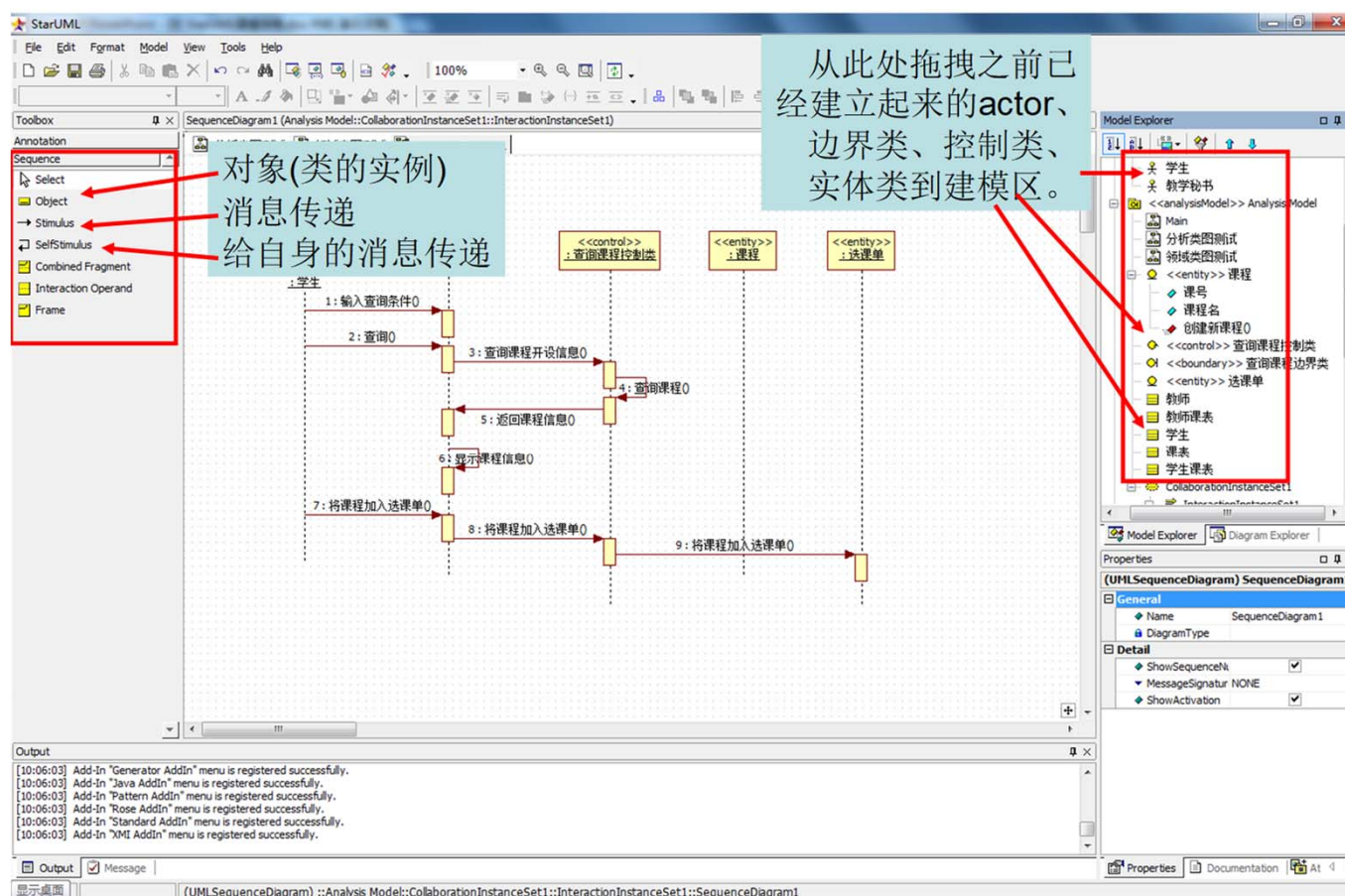
Step1: 选中某个类

Step2: 点击Attribute右侧的...按钮

Step3: 在弹出窗口增加/删除/修改属性

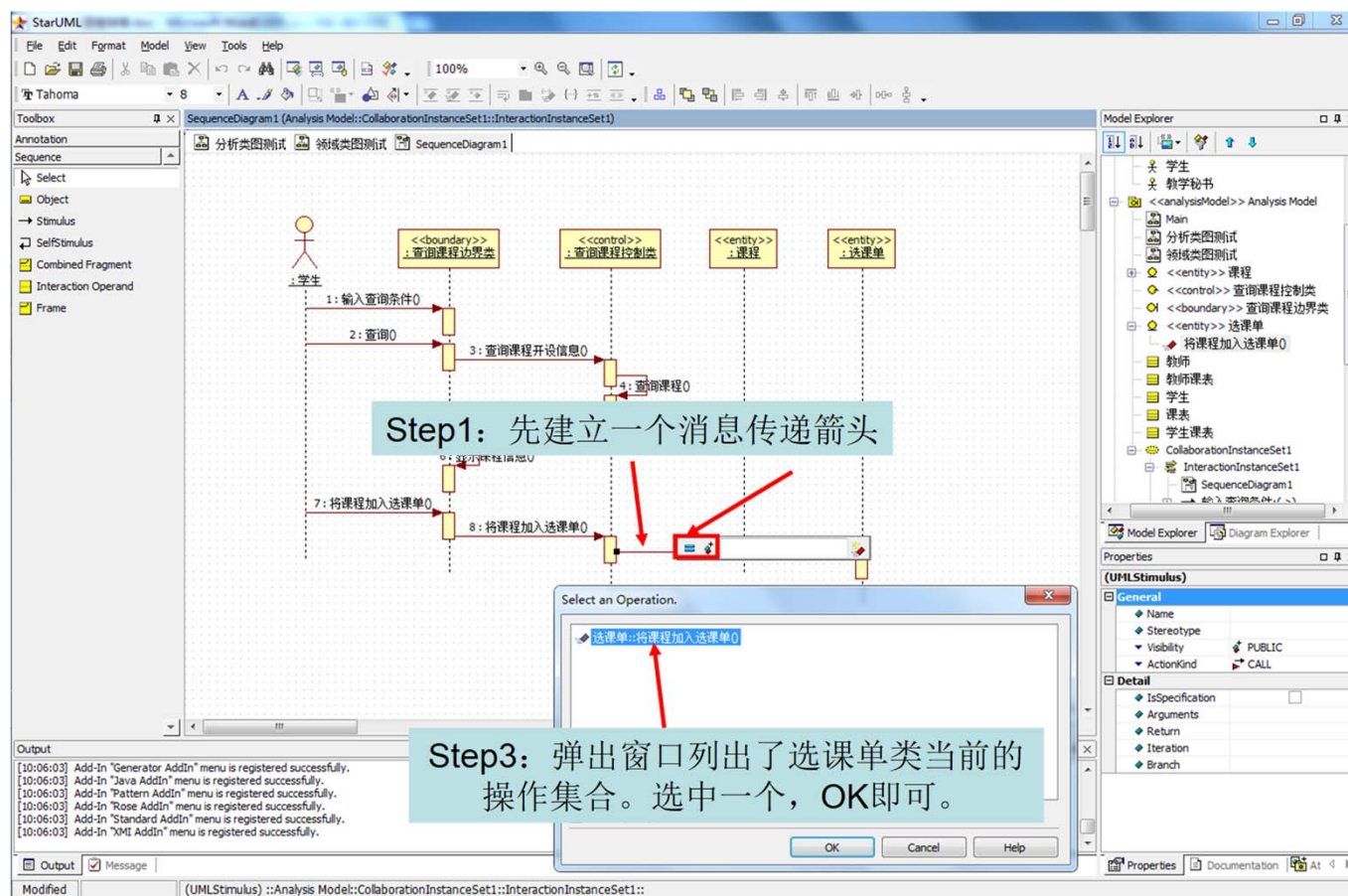
# StarUML建模指南

- 建立时序图。在模型浏览区的<<analysisModel>>节点上点击右键，选择Add Diagram、Sequence Diagram，并为新建立的图形命名。此时左侧符号区展示了时序图的要素。



# StarUML建模指南

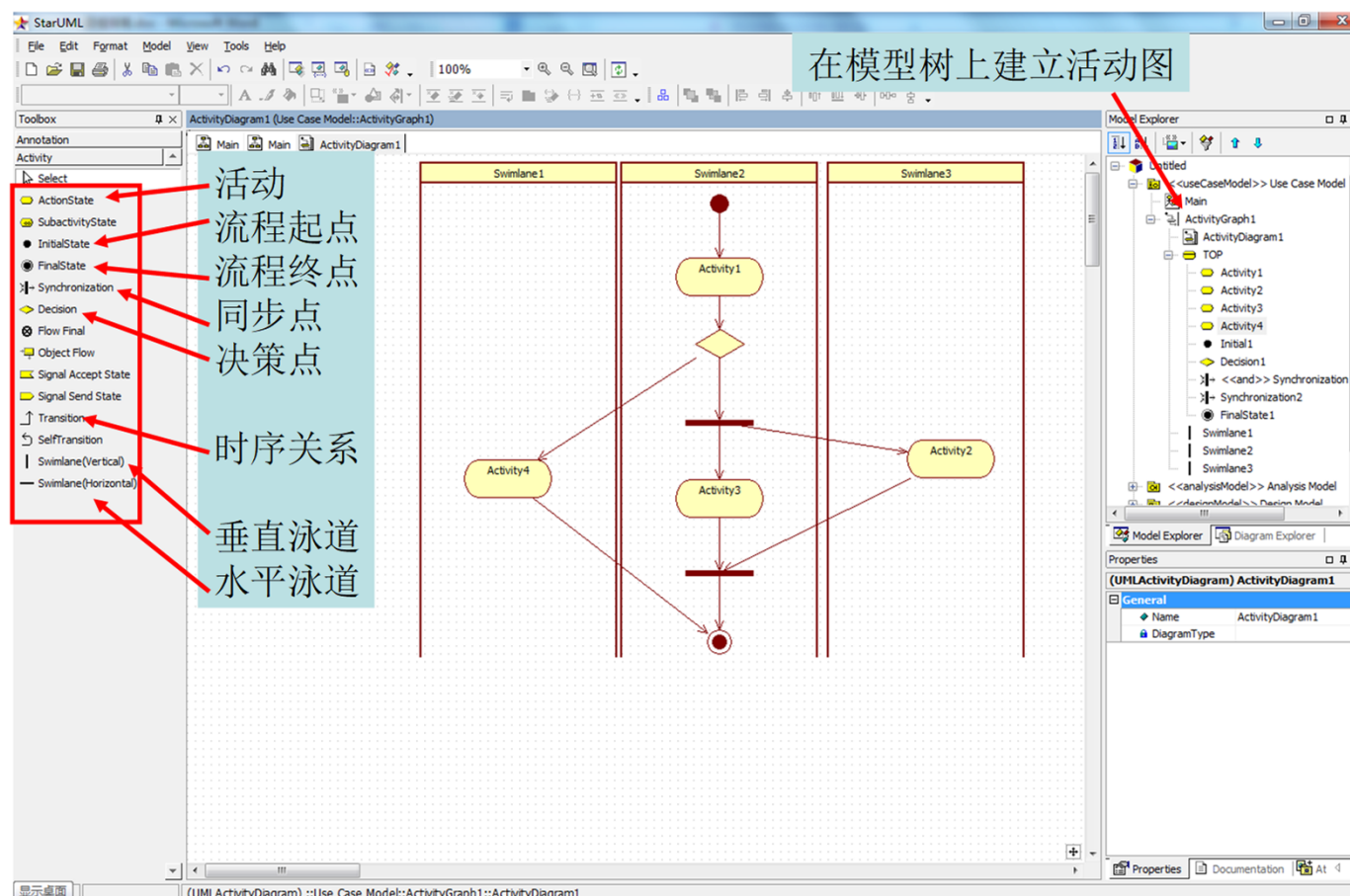
- 正常情况下，建立时序图时无需建立任何新模型要素，完全可以利用之前的用例模型、分析类图、领域类图建立起来。例如，每个消息上对应的操作，可以直接从箭头所指向的类的操作集中选取。



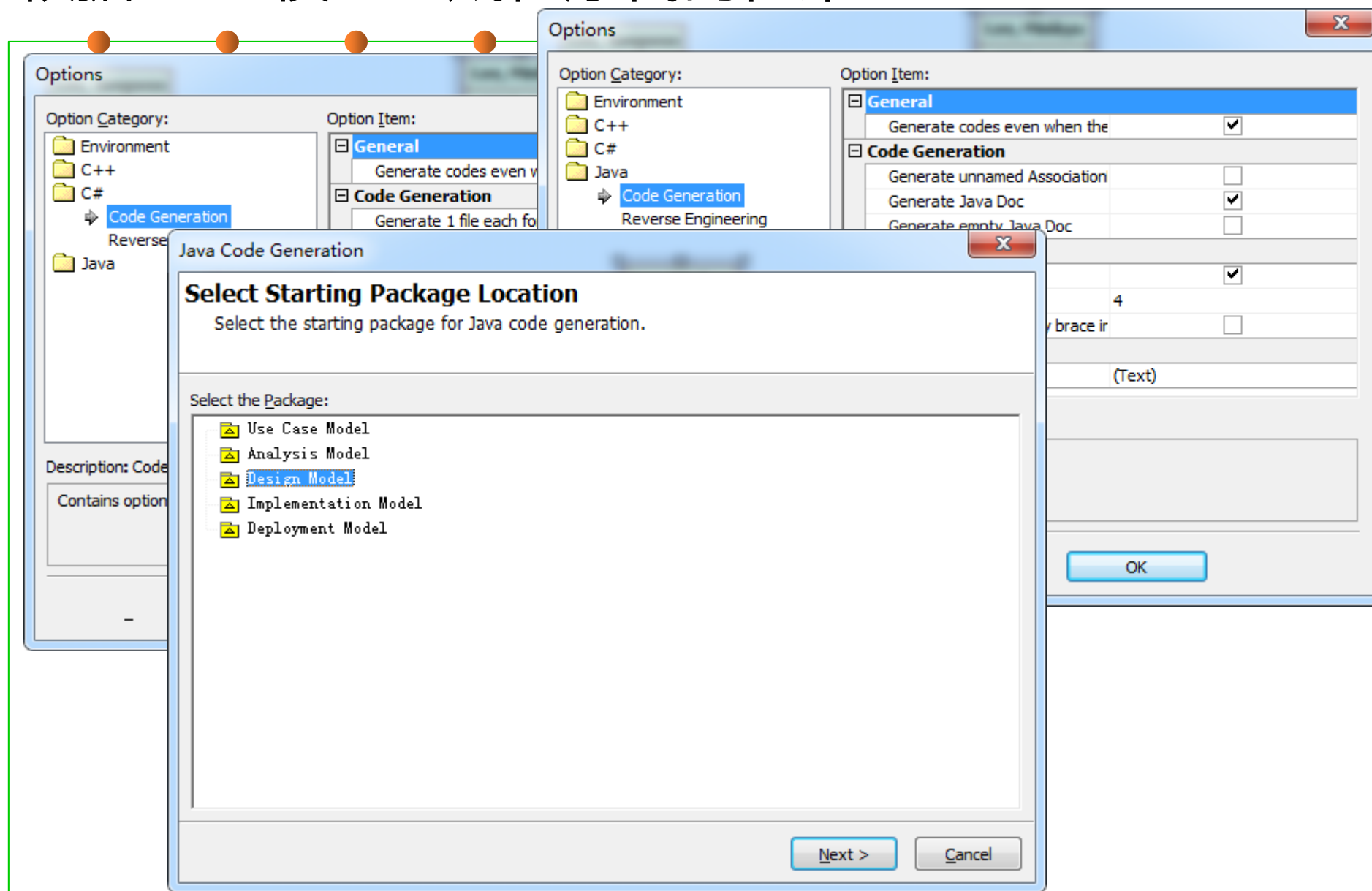


# StarUML建模指南

- 建立流程图(泳道图)。在模型浏览区的<<useCaseModel>>节点上点击右键，选择Add Diagram、Activity Diagram，并为新建立的图形命名。此时左侧符号区展示了活动图的要素。



# 根据UML模型生成程序代码框架



# StarUML建模指南

- 其他UML视图的建立方法，可以按类似的模式自行探索解决。

# GenMyModel

- “云”建模工具GenMyModel: <https://www.genmymodel.com>



# 评判标准

- 所覆盖功能的完整性；
- 所建立模型的合理性；
- 所建立模型的细节丰富程度与准确性。



# 提交方式

- 请遵循实验报告模板撰写。
- 提交日期：第12周周一晚(11月20日 23:55)
- 提交两个文件到CMS：
  - 实验报告：命名规则”学号-Lab7-report.doc”
  - StarUML或GenMyModel的导出文件：命名规则 “学号-Lab7-model.uml”  
(扩展名取决于你所用的建模工具)
  - 同组的两人均要提交。
- 确保GitHub上包含了本次实验中重构的代码。



哈爾濱工業大學  
HARBIN INSTITUTE OF TECHNOLOGY

软件工程

结束

