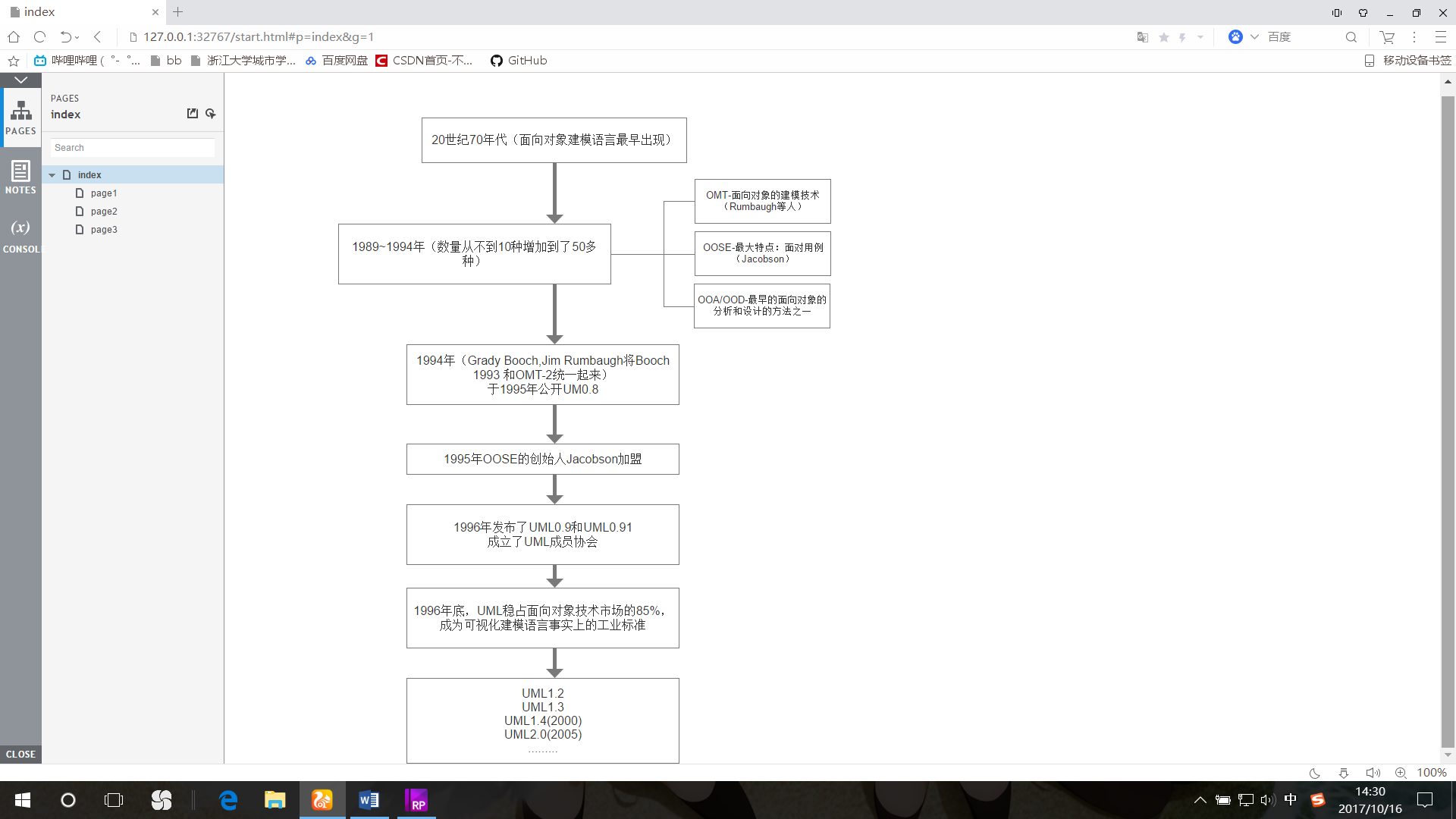
# UML语言

## UML的发展历程

UML发展历程简略图



## UML的组成部分

### 事物：是UML中的重要组成部分

构件事物：是UML模型图的静态部分，描述概念或物理元素

类：对一组具有相同属性、操作、关系和语义的对象的抽象

接口：类或组件提供特定服务的一组操作集合

协作：事物间的相互作用的集合

用例：系统对一个特定的角色执行的一系列动作

构件：也称“组件”，物理上火可替换的系统部分

节点：描述基础硬件，运行时的物理元素

行为事物：是UML模型图的动态部分，描述跨越空间和时间的行为

分组事物：是UML模型图的组织部分，描述事物的组织结构，主要由包来实现

注释事物：是UML模型的解释部分，用来对模型中的元素进行说明，解释

### 图：是很多有相互关系的元素的组

用例视图（外部视图、用户视图、功能视图）：主要描述一个系统应该具备的功能

逻辑视图（静态视图、结构模型图）：主要用于描述在用例图中提出的系统功能的实现

并发视图（动态视图、进程视图）：从资源的有效利用、代码的并行执行以及系统环境中异步事件的处理等方面来考虑

组件视图（实现视图、物理视图）：描述系统的实现模块及它们之间的依赖关系

配置视图（部署视图）：显示系统的物理部署，位于节点上的运行实例的部署情况

### 关系：把元素紧密联系在一起

依赖：是两个模型元素间的语义关系，其中一个元素发生变化会影响另一个元素的语义。

关联：指明了一个对象与另一个对象间的关系。

泛化：一般化-特殊化的关系。

实现：一个类指定了又另一个类必须执行的约定

## UML的13种图：

用例图：从用户角度描述系统功能，并指出各功能的操作者

类图：更加直观地了解一个系统的体系结构

对象图：对象图是类图的实例，一个对象图是类图的一个实例

状态机图：描述一个实体基于事件反映的动态行为，显示了改实体是如何根据当前所处的状态对不同的事件做出反映

活动图：记录了单个操作或方法的逻辑，或者单个业务流程的逻辑

顺序图：描述了对象之间动态的交互关系，主要体现对象之间进行消息传递的时间顺序

通信图：描绘对象之间消息的移动情况来反映具体的方案

构件图（组件图）：描述代码部件的物理结构及各部件之间的依赖关系，有助于分析和理解部件之间的相互影响程度

部署图（配置图）：描述系统中的硬件和软件的物理配置情况的系统体系结构

包图：展现模型要素的基本组织单元，以及这些组织单元之间的依赖关系。

组合结构图：描述系统中的某一部分的内部内容。

交互概览图：直观表达一组相关顺序图之间的转向逻辑

时间图：可选的交互图，展示交互过程中的真实时间信息，具体描述对象状态变化的时间点以及维持特定状态的时间段。

## 图的分类：

|  |  |
| --- | --- |
| 类型 | 包含 |
| 静态图 | 类图、对象图、包图、组合结构图 |
| 行为图 | 状态机图、活动图 |
| 用例图 | 用例图 |
| 交互图 | 顺序图、通信图、时间图、交互概念图 |
| 实现图 | 构件图、部署图 |

问题：（开放题）看了这本书的前面部分内容后，谈谈你对UML的认识与体会

# 面向对象方法（Object Oriented）

为了提高软件系统的稳定性、可修改性和可重用性，人们在实践中逐渐创造出软件工程的一种新途径——面向对象方法。

要点：

1. 客观世界有各种对象组成，任何事物都是对象，复杂的对象可以有简单的组合而成
2. 把所有对象都划分成各种对象类，每个对象类都定义了一组数据和一组方法，数据用于表示对象的静态属性，是对象的状态信息
3. 按照子类和父类的关系，吧若干对象类组成一个层次系统
4. 对象之间彼此通过消息进行联系

## 对象（Object）

是面向对象的基本构造单元，是系统中用来描述客观事物的一个实体。

对象=属性+对属性的操作

对对象的操作通常叫做方法

对象的特征

1. 模块性：独立存在的实体
2. 继承
3. 动态连接性：对象之间通过消息来传递

## 面向对象开发分成以下4个阶段

1. OOA（Object-Oriented Analysis面向对象分析）系统调查和需求分析，分析问题并求解，通过识别并筛选对象，标识对象的属性，识别对象的行为来进行分析。
2. OOD（Object-Oriented Design 面向对象设计）整理问题，对第一阶段结果进一步抽象、归类整理。提出程序设计的思路和方法，对算法进行设计
3. OOP（Object-Oriented Program 面向对象编程）
4. OOT（Object-Oriented Test 面向对象测试）发现程序中的错误，进行改正，使得系统更健壮

# 软件建模概述

建模是为了更好的理解正在开发的系统。

软件建模是开发优秀软件的一个核心工作，是要把设计的结构和系统的行为联系起来，并对系统的体系结构进行可视化和控制。

要达到的目的

1. 有助于按照实际情况或安装所需要的样式对系统进行可视化
2. 规约系统的结构或行为
3. 给出了知道构造系统的模板
4. 对做出的决策进行文档化