



软件工程导论实验三



概要设计说明书主要包含如下内容：

- 系统的**运行环境**。
- 系统的**开发环境**。
- 系统的**总体功能结构图**。
- **功能需求描述**（可用文字详细描述）。
- 各功能模块的**接口设计**（主要指各功能模块的输入、输出）。
- **界面设计**。



系统的**运行环境**：

运行环境包括**软件**和**硬件**两方面

1. 软件：主要指操作系统，比如windows, 或者Linux等等，也包括一些其他软件，比如office，还有第三方运行库，比如DotNet, DirectX等。
2. 硬件，主要是电脑的配置，比如CPU，内存，显卡，硬盘等等。



系统的常用**运行环境**：

- WIN7、WIN10。
- Android、IOS。
- 微信小程序。
- Web应用。



系统的**开发环境**：

- 常用的开发语言：Java、Python、C++、.net、C#等。



系统的**常用开发工具**：

程序员常用的IDE有哪些？

- **集成开发环境**（Integrated Development Environment，IDE）是用于程序开发的应用程序，**集成了代码编写功能、分析功能、编译功能、调试功能**等功能。简而言之，程序员平时就在IDE上写代码，也在IDE上进行调试运行，等程序稳定之后才会上线运行。



主流的有以下几种IDE。



- **Microsoft Visual Studio**（简称VS）是微软公司提供的IDE，可以在VS上编写C、C++、C#等多种语言的项目，所写的代码适用于微软支持的所有平台，包括Microsoft Windows、Windows Phone、Windows CE、.NET Framework、.NET Compact Framework、Microsoft Silverlight等。



- **Android Studio** Android Studio 是谷歌推出的一个Android集成开发工具，基于IntelliJ IDEA. 类似 Eclipse ADT, Android Studio 提供了集成的 Android 开发工具用于开发和调试。



- **Xcode**是苹果公司向开发人员提供的IDE，用于开发macOS、iOS、WatchOS和tvOS的应用程序，Xcode只能运行在OS X上，开发者可以免费使用。



- **IntelliJ IDEA**由JetBrains公司开发，用于开发Java，可以运行在Windows、OS X和Linux平台上，有社区版本和商业版本，社区版本具备大多数常用功能，可以免费下载使用。Google旗下的Android Studio，就是基于IntelliJ IDEA的社区版本发展而成。



- **PyCharm**由JetBrains公司开发，主要用于Python语言开发，可以运行在Windows、OS X和Linux平台上，有社区版本和商业版本，社区版本具备大多数常用功能，可以免费下载使用。



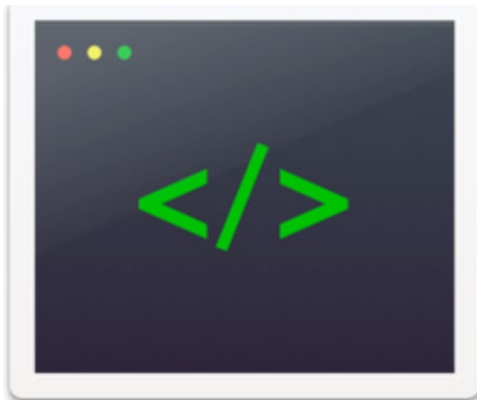
- **Eclipse**最初主要用来开发Java语言，但是众多插件的支持使得Eclipse同样可以用于C++、Python、PHP等其他语言的开发。



- **Adobe Dreamweaver**，简称“DW”，是Adobe旗下的集网页制作和管理网站于一身的所见即所得网页代码编辑器，支持 HTML、CSS、JavaScript 语言的开发，设计师和程序员可以快速建设网站。



- **Unity** 是一款由 Unity Technologies 研发的跨平台2D / 3D 游戏引擎，可用于开发 Windows、MacOS 及 Linux 平台的单机游戏，PlayStation、XBox、Wii、3DS 和 任天堂Switch 等游戏主机平台的视频游戏，或是 iOS、Android 等移动设备的游戏。



- **微信开发者工具**，帮助开发者简单高效地开发和调试微信小程序，集成了公众号网页调试和小程序调试两种开发模式。使用公众号网页调试，开发者可以调试微信网页授权和微信JS-SDK 详情；使用小程序调试，开发者可以完成小程序的 API 和页面的开发调试、代码查看和编辑、小程序预览和发布等功能。

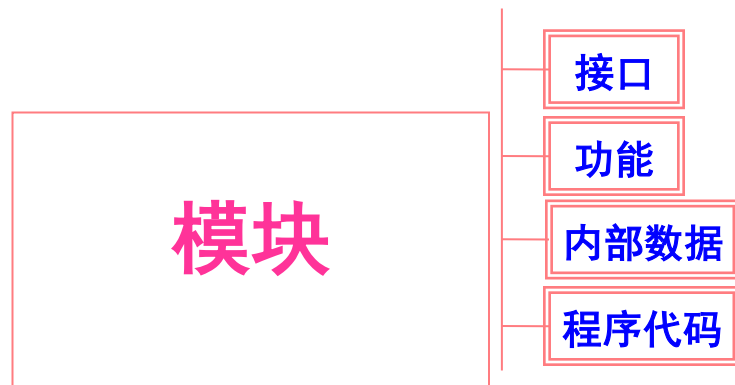


系统的**开发环境**（常用的后台DBMS）：

- 数据库管理系统软件： SYBASE、DB2、ORACLE、MySQL、ACCESS、Visual Foxpro、MS SQL Server 、Informix、 PostgreSQL。



模块的组成

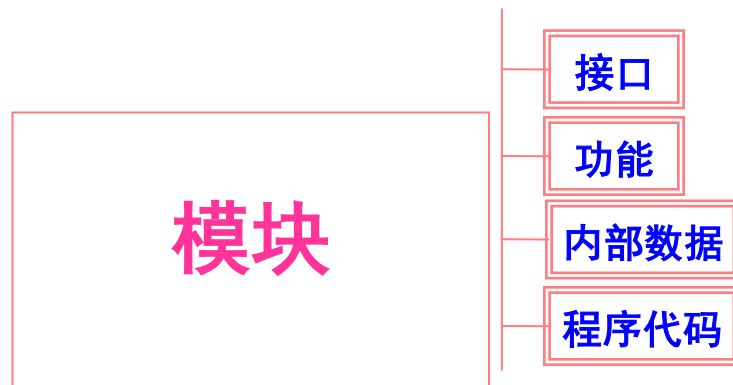


➤ 模块的组成：

- ◆ **接口**，模块的**输入输出**；用于实现模块与其他模块间的**数据传送**，即**向模块传入**所需的**原始数据**及**从模块传出**得到的**结果数据**。



模块的组成



➤ 模块的组成：

- ◆ **功能**，指模块实现什么功能，有什么**作用**；
- ◆ **内部数据**，描述**模块内部实现功能需求所需数据**；
- ◆ **程序代码**，用于描述**实现模块功能的具体方法和步骤**；



模块的特征

- 模块的**输入/输出**和**功能**构成了模块的**外部特征**。
- 模块的**内部数据**和**程序代码**反映的是模块的**内部特征**。



结构化方法

结构化分析(SA)

- 典型的结构化分析方法可以描述为：功能分解=功能+子功能+功能接口。
- 问题域映射为功能和子功能，规格说明间接反映问题域。
- 分析的结果是系统、子系统、功能、子功能层次结构的建立。



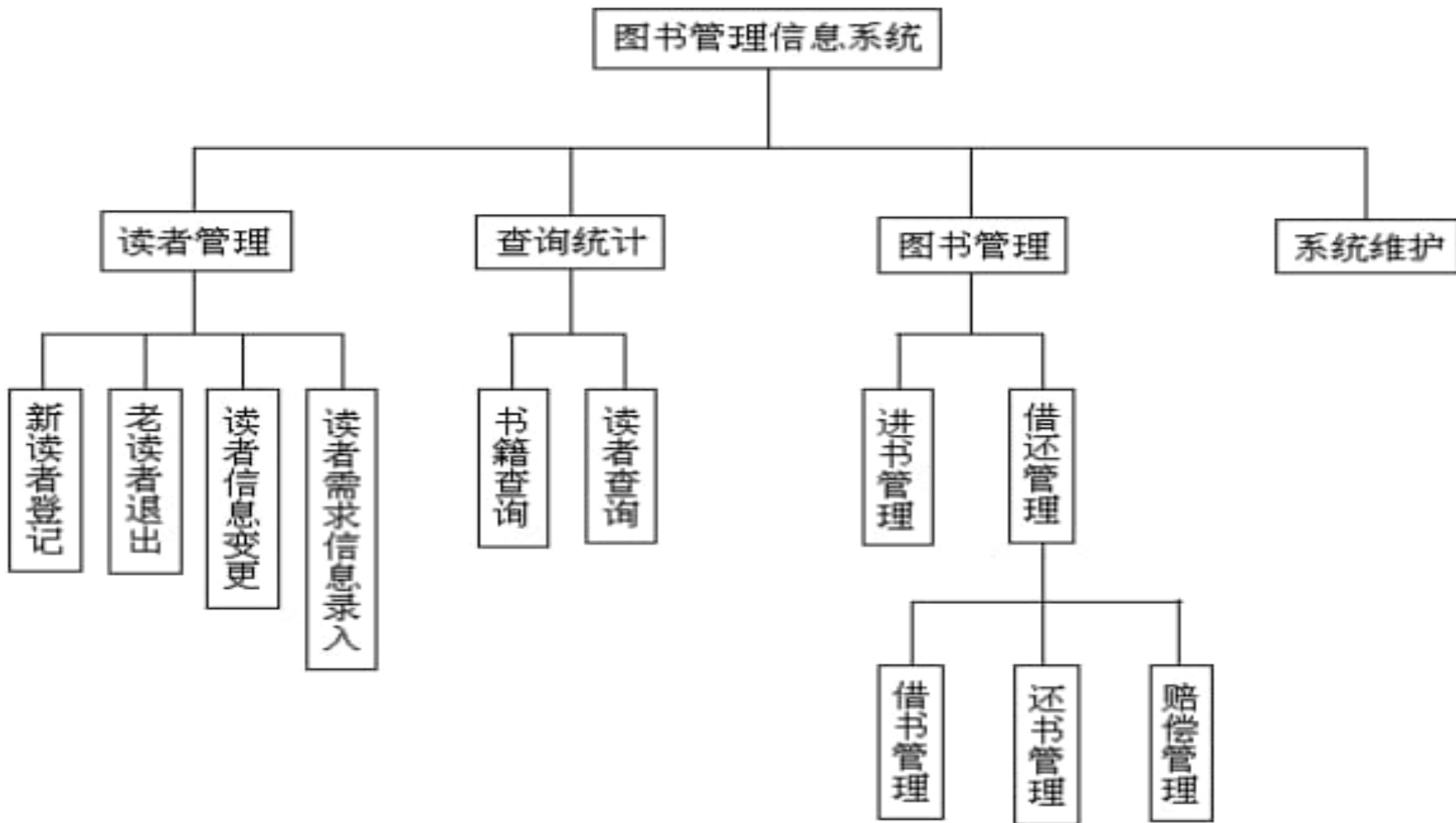
结构化方法

结构化分析(SA)

例如，假设系统很复杂，为了理解它，将它分成了4个子系统，如果子系统仍然比较复杂还可以再继续分解它，如此下去，直到每个子系统足够简单，能清楚地被理解和表达为止。



系统的总体（层次）功能结构图

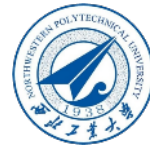




结构化方法

结构化分析(SA)

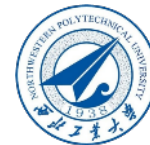
XXX信息管理系统划分功能模块时**通常划分为**系统
用户权限管理、用户注册登录管理、系统基础信息管理
(基础表的增、删、改、查)、系统的若干业务功能模
块等。



结构化方法

结构化分析(SA)

对于**系统用户权限管理**功能模块，主要包括系统用户及其所使用的的权限的添加、删除、修改和查询，本次实验中各组在分析时可不用考虑该模块。



结构化方法

结构化分析(SA)

对于**用户注册登录管理**功能模块，主要包括用户所需注册的各项信息（需明确用户所需注册的各项详细信息、数据类型及其有效性）、用户登录、密码修改、找回密码等功能，本次实验中各组在分析时可不用考虑该模块。



结构化方法

结构化分析(SA)

系统**基础信息管理**模块包括系统所需要用到的基础信息表的添加、删除、修改和查询等功能。

- “添加数据”时应明确需要添加的数据的具体信息（可用数据字典表示）、数据类型及其有效性、默认值等。
- “删除数据”时应先明确如何定位到所需删除的数据，之后在用户点击“删除”按钮时，系统应提示用户再次确认后再进行数据的删除。



结构化方法

结构化分析(SA)

- “修改数据”时应先明确如何定位到所需修改的记录，然后说明该记录的哪些信息可以修改，哪些信息不能修改，修改成功后系统应提示用户修改成功。
- “查询数据”应明确具体的查询条件及查询出来的数据显示格式。

注：本次实验中各组在分析时这样的模块（多于二个时）最多由二位同学负责。



结构化方法

结构化分析(SA)

系统的**业务功能**模块应明确三个W，一个H，三个W分别是When，Where，Who，即明确业务的触发条件，在哪儿（部门）触发，由谁（用户）触发，一个How即业务的具体执行过程（抽象成业务的逻辑模型），一般用数据流图+数据字典或**业务流程图**表示，也可用**文字进行详细的描述**，考虑到数据流图+数据字典已在实验二中完成，此次实验的业务流程可用**文字进行详细的描述**或**业务流程图**来表示。



结构化方法

结构化分析(SA)举例：

功能需求描述（用文字详细描述）见ftp。



界面设计目的

界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象，设计良好的界面能够引导用户自己完成相应操作，起到向导作用。界面设计主要是为了达到以下目的：

- 1) 以用户为中心。设计由用户控制一致的界面，而不是界面控制用户。
- 2) 清楚一致的设计。所有**界面的风格保持一致**，所有具有相同含义的术语保持一致，且易于理解和使用。



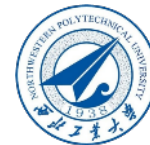
界面设计目的

- 3) 拥有良好的直觉特征。以用户所熟悉的现实世界事务的抽象来给用户暗示和隐喻，来帮助用户能迅速学会软件的使用。
- 4) 较快的响应时间。
- 5) 简洁、美观。



用户界面设计原则（启发式评估的十条准则）

- ✓ 系统状态可见
- ✓ 系统与现实匹配
- ✓ 用户自由掌控
- ✓ 保持一致且遵循标准
- ✓ 防止错误发生
- ✓ 信息重现而不是依赖记忆
- ✓ 适应用户并关注效率
- ✓ 美化并简化设计
- ✓ 帮助用户远离错误
- ✓ 帮助信息文档化



用户界面设计辅助工具

Axure, Mockups, 墨刀（适合轻量级移动应用）等。