XX项目概要设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| **文件版本：** |  | **文件编号：** |  |
| **发布日期：** |  | **编 制：** |  |
| **审 核：** |  | **批 准：** |  |

*【模版使用说明：*

*1、模版中斜体字相当于撰写指南，最后文稿请将本模版中所有的斜体字部分全部删除。*

*2、模版中并不说明设计技术和方法，而只是说明包含哪些内容，以及如何描述、组织这些内容。*

*3、文档正文应使用常规、黑色、五号字体即系统设置的“正文”样式。】*

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **日期** | **版本说明** | **作者** | **备注** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

*【修订记录说明：*

*草稿的版本号使用V0.y,如V0.1、V0.2*······*，评审后正式发布文档使用Vx.0,如V1.0、V2.0*······，*修订版使用Vx.y，如V1.1、V1.2*······*】*

目录

[XX项目概要设计说明书 1](#_Toc435107526)

[1 引言 1](#_Toc435107527)

[1.1 目的 1](#_Toc435107528)

[1.2 范围 1](#_Toc435107529)

[1.3 术语定义 1](#_Toc435107530)

[1.4 参考文档 1](#_Toc435107531)

[2 系统目标和约束 2](#_Toc435107532)

[2.1 系统目标 2](#_Toc435107533)

[2.2 设计约束 2](#_Toc435107534)

[3 系统部署视图 2](#_Toc435107535)

[4 逻辑视图 3](#_Toc435107536)

[4.1 概述 3](#_Toc435107537)

[4.2 包/子系统N设计 3](#_Toc435107538)

[4.2.1 简要说明 3](#_Toc435107539)

[4.2.2 组成 4](#_Toc435107540)

[4.2.3 设计重用 4](#_Toc435107541)

[4.2.4 接口 4](#_Toc435107542)

[4.2.5 时序图 4](#_Toc435107543)

[4.3 包/子系统N设计 5](#_Toc435107544)

[4.3.1 简要说明 5](#_Toc435107545)

[4.3.2 组成 5](#_Toc435107546)

[4.3.3 设计重用 5](#_Toc435107547)

[4.3.4 接口 6](#_Toc435107548)

[4.3.5 时序图 6](#_Toc435107549)

[5 外部接口 6](#_Toc435107550)

[6 数据视图 7](#_Toc435107551)

[7 运行视图 7](#_Toc435107552)

[8 附录 7](#_Toc435107553)

# 引言

*【本文档用于描述软件产品或系统的概要设计。*

*对于规模较大或风险较大的项目，可在概要设计前单独进行架构设计活动，设计的内容写入“架构设计报告”；*

*如果项目已单独建立“架构设计报告”，可裁剪本文档中的“架构设计”部分;*

*文档中的提示性语言，在文档编写完成后删掉。】*

## 目的

*【本文确定概要设计文档在整个项目文档中的作用或目的，应确定此此文档的特定读者，并指出他们应该如何使用次文档】*

## 范围

*【简要说明概要文档使用的范围和影响的范围】*

## 术语定义

*【列出本文当中所使用的专业术语和缩写词的原文。】*

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **解释** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 参考文档

*【列出有关的参考文件，如：*

*本项目的经核准的计划任务书或合同，上级的批文；*

*属于本项目的其他已发表文件；*

*本文件中各处引用的文件、资料，包括所要用到的软件开发标准。列出这些文件的标题、文件编号、发表日期和出版单位，说明能够得到这些文件资料的来源。】*

# 系统目标和约束

## 系统目标

*【本节说明架构设计所要达成的系统目标，定义系统的总体功能，并指出系统总体的设计原则、性能要求】*

## 设计约束

*【记录可能适用的特殊约束：设计与实施策略、开发工具、团队结构、时间表、旧代码等。】*

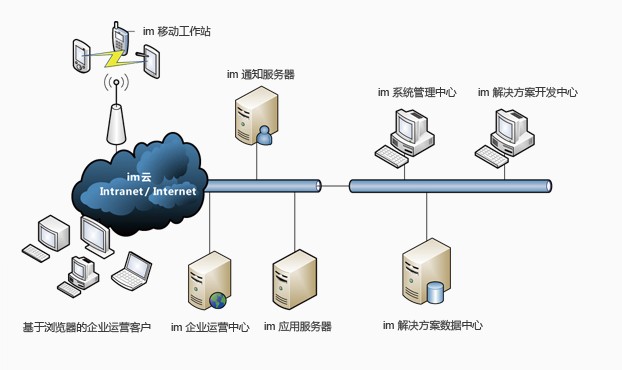
# 系统部署视图

*【描述使用本软件系统典型用户的部署设备分布图及设备上相应基本软件配置。软件环境包括：操作系统、数据库、以及其它支撑软件；硬件环境包括：主机类型、网络类型、存储器容量、其它特殊设备。*

*结合运行体系图中提到的“程序”，将相关的“运行程序”分布到各个部署节点中；*

*可采用网络拓扑图表示。】*

*如：*



# 逻辑视图

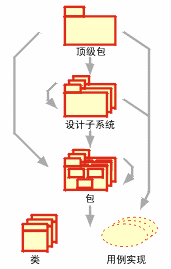
*【说明系统的层次、子系统/模块的划分，描述之间的关系，以系统体系结构图及软件结构图的方式表示系统的层次、子系统/模块划分并加以说明。*

*按照复用的原则进行分层，即上层的设计元素依赖下层的设计元素，而不是相反。分层是最高的组织，子系统、包或模块应位于特定的层内。例如：客户层、请求控制层、业务逻辑层、数据访问层、基础架构层。】*

## 概述

*【本节按照设计模型中包的层次结构来说明设计模型的整体分解情况。该视图以图形方式说明关键的用例实现、子系统、包和类，它们包含了在构架方面具有重要意义的行为。逻辑视图在每次迭代过程中都会加以改进。】*

*如：*

**

*【逻辑视图表示了类、子系统、包和用例实现的子集。】*

## 包/子系统N设计

*【对于每个重要的包/子系统，都用一个小节来加以说明，其中应包括该包/子系统的名称、简要说明以及显示该包/子系统中所有重要的类和包的图。*

*对于该包/子系统中的每个重要类，应包括其名称、简要说明和（可选）对其主要职责、操作和属性等的说明。】*

### 简要说明

*【对该包/子系统的简要说明】*

### 组成

*【本节枚举了该包/子系统中包含的所有元素。首先列出的是所包含的包/子系统，然后再列出所包含的所有类。给出了各元素的名称和简要说明。】*

### 设计重用（可选）

#### 以往设计的重用

*【说明将过去设计的信息、知识或者经验如何重新应用到本次软件设计及后续设计活动，包括：库函数，模板，面向对象、设计模式等】*

#### 可重用性设计

*【说明本次软件设计及后续设计活动的信息、知识或者经验可重新应用到其他设计的考虑，包括：库函数，模板，面向对象、设计模式等】*

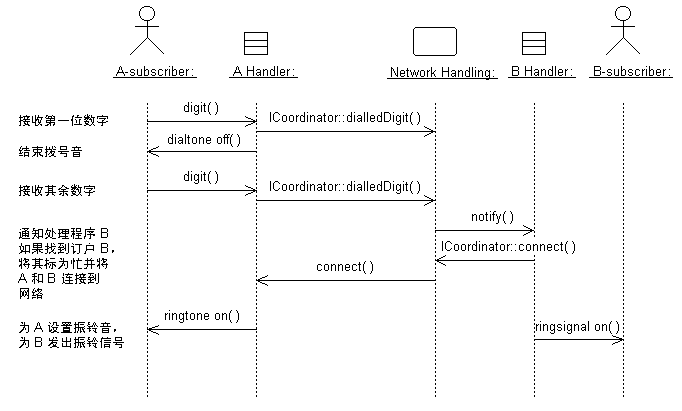
### 接口

*【定义和描述本包/子系统与其它包／子系统存在的数据交互的接口，其中应包括对接口的数据名称、数据类型和数据说明等的描述。】*

### 时序图

*【由于脚本和消息（临时消息规约，尚未指定给操作）说明了类要执行的既定行为，因此操作应该支持序列图中的消息内容。时序图的示例如下：】*

*如：*



## 包/子系统N设计

*【对于每个重要的包/子系统，都用一个小节来加以说明，其中应包括该包/子系统的名称、简要说明以及显示该包/子系统中所有重要的类和包的图。*

*对于该包/子系统中的每个重要类，应包括其名称、简要说明和（可选）对其主要职责、操作和属性等的说明。】*

### 简要说明

*【对该包/子系统的简要说明】*

### 组成

*【本节枚举了该包/子系统中包含的所有元素。首先列出的是所包含的包/子系统，然后再列出所包含的所有类。给出了各元素的名称和简要说明。】*

### 设计重用（可选）

#### 以往设计的重用

*【说明将过去设计的信息、知识或者经验如何重新应用到本次软件设计及后续设计活动，包括：库函数，模板，面向对象、设计模式等】*

#### 可重用性设计

*【说明本次软件设计及后续设计活动的信息、知识或者经验可重新应用到其他设计的考虑，包括：库函数，模板，面向对象、设计模式等】*

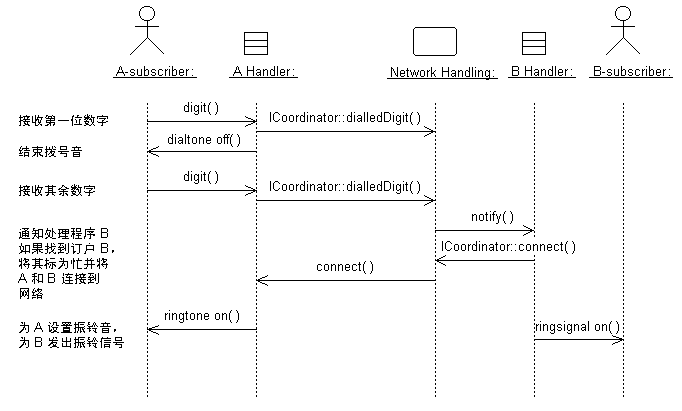
### 接口

*【定义和描述本包/子系统与其它包／子系统存在的数据交互的接口，其中应包括对接口的数据名称、数据类型和数据说明等的描述。】*

### 时序图

*【由于脚本和消息（临时消息规约，尚未指定给操作）说明了类要执行的既定行为，因此操作应该支持序列图中的消息内容。时序图的示例如下：】*

*如：*



# 外部接口

*【描述本系统所使用的外部系统提供的编程接口、本系统向其它系统提供的编程接口以及与外部系统相关的硬件接口。需要详细说明接口类型、通信协议、数据交换格式等接口实现内容，同时需要概要描述与本系统相关的外部系统的功能。*

*针对复杂的接口，可以采用单独的接口设计文档进行描述。】*

# 数据视图

*【说明系统的主要数据流向，持久化数据的存储和数据分布方案，根据数据分布方案确定数据同步、复制、传递的策略。】*

# 运行视图

*【以组件图的形式说明系统在运行时的体系结构，主要考虑以下几个问题：*

*系统有多少个程序要运行，系统有哪些地方需要并发，采用什么方式？是多线程/进程的方式吗？*

*每个程序或线程的生命周期（什么时候开始，什么时候结束）；生命周期问题的解决方案；*

*进程/线程间如何通讯? 如：如内存共享、信号、消息传递、PRC、事件广播等;*

*进程或线程间共享资源如何进行协调和分配？*

*确定在进程或线程内执行的设计元素，如：子系统、类和模块，必须至少在一个进程内执行指定子系统、类和模块的实例.*

*上述问题尽量采用图表方式描述，有必要的话采用多个图表方式，并加以文字说明。】*

# 附录

*【与本设计文档相关，但不方便在以上章节中包含的内容，或者由于在多个地方需要使用而篇幅较长的信息等，例如，相关的数据辞典，数据结构，方案原型（对比）测试结果等】*