

第3章 管理信息系统的开发与建设

管理信息系统开发涉及的知识领域广泛,涉及的单位部门众多,需要在计算机技术、管理业务、组织及行为等方面全面把握。管理信息系统建设是一项复杂的系统工程,涉及面广,因素多,涉及企业组织内部各级机构、人员及外部环境,涉及管理体制、管理方法和管理手段及人们的工作习惯。管理信息系统的目标、规模、功能和实施步骤必须与企业组织的目标、管理水平、职工素质相适应,甚至要在一定范围改革企业不适合的规章制度、工作惯例,使企业适应先进的管理信息系统要求。本章将对管理信息系统开发、建设方面的相关内容展开讨论。

3.1 管理信息系统开发概述

3.1.1 管理信息系统开发的概念和特点

1. 管理信息系统开发的概念

MIS 开发是指建立 MIS 的全过程。“全过程”指从提出建立 MIS(系统规划)、经过系统分析、系统设计、系统实施和评价直到用新系统代替原系统的全过程。然后进入系统运行和维护,这也称为 MIS 的一个生命周期(Life Cycle)。

2. 管理信息系统开发的特点

开发、建设一个大的管理信息系统是一个相当复杂的系统工程,不仅技术要求高、难度大,工程的组织管理也非常复杂,需耗费大量人力、财力和时间。同时,管理信息系统的建立、应用及其发展还要直接受到社会、组织文化等多方面的影响。MIS 的开发主要涉及信息技术、组织管理和系统工程等,如图 3-1 所示。

系统方法或系统方法论是研究管理信息系统开发方法的重要思想。系统工程有两层含义,作为科学,它是以研究大规模复杂系统为对象、以系统概念为主线,引用其他学科的一些理论、概念和思想而形成的多元目的科学;作为工程,它又是一门工程技术,具有和一般工程技术相同的特征,但又具有本身的特点。信息系统的开发并不研究特定的工程物质对象,而是研究为协调物质对象而存在的信息系统,研究如何将现有的人工信息管理模式转换成利用计算机、通信等技术的现代化管理模式。这种转换过程就是信息系统开发,从这个意义上说,可以将信息系统的开发过程称为信息系统工程,一方面从系统的概念出发,首先考虑系统的全局结构,着眼于整体最优,再进一步考虑系统的各个组成部分的主要功能以及组成部分之间的协

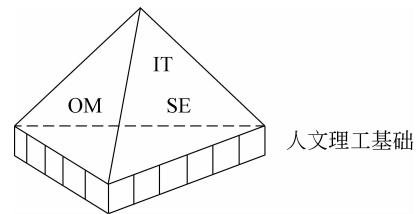


图 3-1 MIS 开发模型

调一致,进而达到系统的最终目标。在开发过程中要将管理学、人际关系学、组织行为学、计算机科学、通信技术等先进的科学技术有机地结合在一起。另一方面系统的开发体现出了些工程的特性,即所有开发人员的工作必须遵循一个正确的方法、按照一定的工作标准和程序、利用有效的工具来进行,整个开发过程要分阶段、分步骤地逐步实施。每个阶段、每一步骤都应该有一系列的文档资料作为开发工作的阶段性成果,这些成果一定要经过正确性验证。

系统方法的要点可以归结为:系统的思想、数学的方法和计算机技术。系统的思想即把研究对象作为一个系统,考虑系统的一般特性和被研究对象的个性;数学的方法是用定量技术即数学方法来研究系统,通过建立系统的数学模型和运行模型,将得到的结果进行分析、再用到原来的系统中;计算机技术是求解数学模型的工具,在计算机上用数学模型对现实系统进行模拟,以实现系统的最优化。

3.1.2 管理信息系统的开发原则

MIS 开发是一个复杂的系统工程,为使其开发成功,要考虑各方面的因素坚持以下原则。

1. 实用性原则

系统必须满足用户管理上的要求,既保证系统功能的正确性又方便实用,需要友好的用户界面、灵活的功能调度、简便的操作和完善的系统维护措施。

2. 系统性原则

在 MIS 的开发过程中,必须十分注重其功能和数据上的整体性、系统性。

3. 符合软件工程规范的原则

MIS 的开发是一项复杂的应用软件工程,应该按软件工程的理论、方法和规范去组织与实施。

4. 逐步完善,逐步发展的原则

MIS 的建立不可能一开始就十分完善和先进,而总是要经历一个逐步完善、逐步发展的过程。

3.1.3 管理信息系统开发的组织

在 MIS 项目的开发过程中,只有组织管理严密才能保证系统顺利开发并且取得成功。管理信息系统开发人员的组织和人员结构是否合理,将直接影响管理信息系统的开发进度和工作质量,根据开发工作的实际需要和工作性质以及职责要求,可以按如图 3-2 所示框架设立开发人员的组织机构。

1. 系统开发的组织机构与分工

(1) 系统开发领导小组

通常由企、事业单位管理业务的骨干人员、计算机或信息管理的主管人员、系统开发

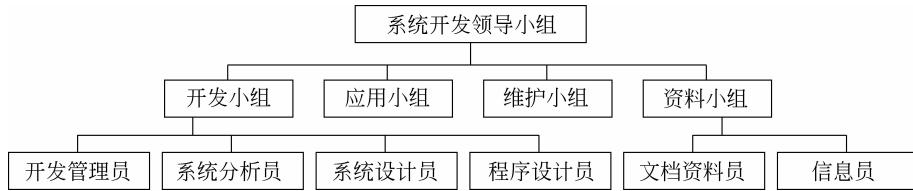


图 3-2 开发人员组织机构示意图

的技术负责人组成。主要负责新系统开发的行政组织和领导工作,具有权威的作用;具体负责机构调整,人员、设备、资金的调配,制定规章制度,项目管理及对系统开发做出重要决策。

(2) 系统开发工作小组

由参加系统开发的所有人员组成,具体负责系统开发工作的组织与实施,在系统开发领导小组的领导下,具体执行系统开发的过程。其中,技术负责人起主导作用。

2. 系统开发的人员组成与职责

(1) 系统分析员

主要负责系统的调查与分析工作,该类人员通常经过专门的培训,对计算机、MIS、现代管理理论和实践都有较丰富的知识。要求该类人员知识面广,善于学习不同行业的业务知识,有很强的负责精神,善于与不同背景的人员进行讨论交流,有较强的组织工作的能力。

(2) 系统设计员

负责系统的设计工作,主要进行系统的总体设计、物理设计。要求该类人员具有熟练的计算机专业知识,掌握建立 MIS 的技术基础,责任心强,熟悉系统实施与转换的一般技术方法。

(3) 程序设计员

负责系统的程序设计、调试和转换工作。要求该类人员精通程序设计语言与编程技巧,掌握系统测试的原理和方法,具有准确理解和贯彻系统分析与系统设计思想的素质和能力,善于学习和运用程序设计的新方法新技术,有一定的美学修养。

(4) 软件工程师

一般指从事软件开发职业的人。10 余年来软件工程师一直占据高薪职业排行榜的前列,作为高科技行业的代表,技术含量很高,职位的争夺也异常激烈。软件开发是一个系统的过程,需要经过市场需求分析、软件代码编写、软件测试、软件维护等程序。软件开发工程师在整个过程中扮演着非常重要的角色,主要从事根据需求开发项目软件工作。

(5) 操作员

参与系统调试和转换工作,负责系统正常运行期间对系统功能的执行(数据录入、查删改、统计、打印输出、数据备份与恢复等)。要求该类人员除有熟练的键盘操作技能,准确的汉字录入能力外,还要掌握基本的硬件操作知识与 OS 命令,善于学习和掌握应用系统的功能结构和性能特点,遵守操作规程,有责任心。

(6) 其他

在系统开发及正常运行后的管理与维护中,可根据需要配备相应的人员。例如,设备维护、文档管理、网络系统管理等专门或兼职人员。

3. 系统开发中的文档管理

(1) 文档的地位和作用

① 文档的含义。文档是记录人们思维活动及其结果的书面形式的文字资料。信息系统的文档是描述系统从无到有整个发展与演变过程及各个状态的文字资料。

② 文档的地位。管理信息系统文档作为系统建设中的重要技术文件,记录了信息系统项目建设过程中各类人员的思维活动,以文字资料的形式描述出了系统开发过程的轨迹。在软件工程学科领域,把程序和文档合并在一起称为软件。程序的必要解释、说明称为文档。程序与文档的作用不同,程序是交给计算机来执行的,文档是交给人们阅读的。由于人脑的思维活动具有不可见、不稳定等特点,用文档来描述人脑的思维活动,使之成为管理信息系统建设过程的可见物,以此来统一人们的思想、认识,同时也可方便系统开发中各类人员能够进行交流、沟通,更好地搞好系统的建设开发工作。文档在系统的开发中具有重要地位,体现在系统建设的各个阶段,在系统总体规划阶段,在进行系统概要调查的基础上,要形成良好的用户需求文档,否则系统分析、设计工作就失去了可靠的基础。在系统的分析设计阶段,在系统总体规划指导下,在进行详细调查的基础上,要形成良好的系统分析、设计文档,否则,将造成信息系统项目开发过程的混乱、无序状态,影响系统开发的效率,甚至导致失败。在系统的试运行、运行维护阶段,也应建立良好的文档,防止对系统运行、维护带来不必要的麻烦和困难。因此,文档是信息系统的重要组成部分,是 MIS 建设的生命线;没有文档的信息系统,不能算是好的信息系统。

③ 文档的作用。文档在信息系统项目的开发建设和运行维护过程中,最重要的作用是进行沟通,它在用户和系统开发的各类人员之间起着桥梁和纽带的作用;还具有监理和审计的作用。每份文档都是项目建设过程中有关人员的一种书面承诺,绝大多数文档都有签名(表明其对所签署部分文档内容的认可),如果出现合同纠纷,文档可以作为重要的证据使用。统一思想,防止健忘和误解,是系统开发工作组内各类人员之间及组内外的通信依据;观察、控制、协调系统开发过程的依据。

④ 信息系统文档编制的要求和方法。如前所述,MIS 文档具有重要的沟通作用。信息系统建设过程在很大程度上是应用软件的开发过程,就软件的两大部分——程序和文档而言,程序必须完成,这往往导致开发人员误认为只要能够保证程序正确、满足系统需求就行了,而对于“软性”的文档,认为有些是可有可无的,有些也可以事后补充,从而造成盲目追求系统开发进度,忽略了一些文档资料的整理工作。另外,文档往往是提供给其他人看的,其作用在很多情况下是事后才能体现出来,也造成了系统开发人员缺乏书写文档的积极性和自觉性,一些不负责的人员只是应付性地把要求提供的文档赶写出来,从而造成文档质量不高。质量不高的文档,不仅会使读者难于理解,给使用造成不便,而且会削弱对信息系统的管理(项目负责人难于确认、评价开发工作),增加信息系统项目的开发成本,甚至造成操作失误等严重后果。

在文档编制过程中,根据不同的读者群,应编制出适合他们需要的文档。用户文档的读者主要是用户,在编制此类文档时,应尽量减少使用信息技术的专业术语,以免造成读不懂。

高质量文档的标准是:完整,任何一个文档都应当是完整、独立的,应该自成体系;清晰,文档编写应力求简明、扼要,在可能的情况下,适当插入图、表,以增加清晰性;易查阅,文档结构和文件装订应以方便查阅者进行查阅为宗旨;精确统一,文档的行文应当确切,不能出现二义性,同一个项目在不同文档中,描述的内容应协调一致,无矛盾冲突;可追溯性,在信息系统项目开发的各阶段,生成的文档与其对应阶段有紧密联系,前后两阶段生成的文档,应具有一定的继承性,即后一阶段的文档应是前一阶段文档随着开发工作的扩展,同一个项目各开发阶段提供的文档存在着可追溯关系;灵活性,对于不同的信息系统项目,依据其规模和复杂程度的不同,文档的简、繁也应做适当处理。

信息系统文档的编制原则是在编写过程中立足于读者,立足于实际需要,文字准确,简单明了。在文档的编排上,应采用由一般到具体的层次结构,在条件允许的情况下,可以采用词汇之间互相链接,图表与其解释、说明性文字就近编排,文档正文亦可以采用不同版式、不同字体以区别文档内容,使文档的编排有利于用户迅速查到所需的内容。

(2) 系统开发人员缺乏文档管理的原因

- ① 开发人员为了追求 MIS 的开发进度。
- ② 开发者往往只注重结果。
- ③ 开发者认为开发过程的“轨迹”不需要保留。
- ④ 文档的作用很多时候是在事后才体现出来。

(3) 文档管理的内容

- ① 文档标准与规范的制定。
- ② 文档编写的指导与督促。
- ③ 文档的收存、保管与借用手续的办理。

3.2 管理信息系统的开发方式

3.2.1 自行开发

由用户依靠自己的力量独立完成系统开发的各项任务。适合有较强专业开发分析与设计队伍和程序设计人员系统维护使用队伍的组织和单位,如大学、研究所、计算机公司、高科技公司等单位。

1. 优点

开发费用少,容易开发出适合本单位需要的系统,方便维护和扩展,有利于培养自己的系统开发人员。

2. 缺点

- (1) 容易受业务工作的限制,系统整体优化不够,开发水平较低。

(2) 系统开发时间长,开发人员调动后,系统维护工作没有保障。

3. 注意事项

(1) 需要大力加强领导,实行“一把手”原则。

(2) 向专业开发人士或公司进行必要的技术咨询,或聘请他们作为开发顾问。

3.2.2 委托开发

由使用单位(甲方)委托通常是有丰富开发经验的机构或专业开发人员(乙方),按照用户的需求承担系统开发的任务。适合情况:使用单位(甲方)没有 MIS 的系统分析、系统设计及软件开发人员或开发队伍力量较弱、但资金较为充足的单位。

1. 优点

省时、省事,开发的系统技术水平较高。

2. 缺点

费用高、系统维护与扩展需要开发单位的长期支持,不利于本单位的人才培养。

3. 注意事项

(1) 使用单位(甲方)的业务骨干要参与系统的论证工作。

(2) 开发过程中需要开发单位(乙方)和使用单位(甲方)双方及时沟通,进行协调和检查。

3.2.3 合作开发

由使用单位(甲方)和有丰富开发经验的机构或专业开发人员(乙方),共同完成开发任务。双方共享开发成果,实际上是一种半委托性质的开发工作。适合情况:使用单位(甲方)有一定的 MIS 分析、设计及软件开发人员,但开发队伍力量较弱,希望通过 MIS 的开发建立、完善和提高自己的技术队伍,便于系统维护工作的单位。

1. 优点

相对于委托开发方式比较节约资金,可以培养、增强使用单位的技术力量,便于系统维护工作,系统的技术水平较高。

2. 缺点

双方在合作中沟通易出现问题,因此需要双方及时达成共识,进行协调和检查。

根据《计算机软件保护条例》第 10 条的规定,计算机软件著作权归属软件开发者。因此,确定计算机著作权归属的一般原则是“谁开发谁享有著作权”。软件开发者指实际组织进行开发工作,提供工作条件完成软件开发,并对软件承担责任的法人或者非法人单位,以及依靠自己具有的条件完成软件开发,并对软件承担责任的公民。

3.2.4 利用现成软件包开发

利用现成的软件包开发 MIS, 可购买现成的应用软件包或开发平台, 如财务管理系、小型企业 MIS、供销存 MIS 等。应用软件包是预先编制好的、能完成一定功能的、供出售或出租的成套软件系统。它可以小到只有一项单一的功能, 比如打印邮签, 也可以是有 50 万行代码的、400 多个模块组成的复杂的运行在主机上的大系统。适合于功能单一的小系统开发颇为有效。但不太适用于规模较大、功能复杂、需求量的不确定性程度比较高的系统的开发。

1. 优点

能缩短开发时间, 节省开发费用, 技术水平比较高, 系统可以得到较好的维护。

2. 缺点

功能比较简单, 通用软件的专用性比较差, 难以满足特殊要求, 需要有一定的技术力量根据使用者的要求做软件改善和编制必要的接口软件等二次开发的工作。

3.3 管理信息系统的开发方法

管理信息系统从产生到现在已经发展了许多开发方法, 其中生命周期法 (Life Cycle Approach)、结构化方法 (Structured Approach)、原型法 (Prototyping Approach) 和面向对象的开发方法 (Object-Oriented Developing Approach) 在 MIS 开发实现中产生了重要的影响。这些方法既有区别又有联系, 可以根据系统规模选择或组合使用。

3.3.1 结构化系统开发方法

1. 结构化开发方法概念

用系统的思想, 系统工程的方法, 按用户至上的原则, 结构化、模块化、自顶向下地对信息系统进行分析与设计; 将管理信息系统的开发分为系统分析、系统设计、系统实施三个阶段。

2. 结构化开发方法的特点与局限性

结构化开发方法的假设是预先定义需求的策略, 结构化开发方法特别适合于开发那些能够预先定义需求、结构化程度又比较高的大型事务型系统 (TPS 即 EDPS) 和管理信息系统 (MIS)。它也有比较明显的缺点: 开发过程复杂烦琐, 周期长、系统难以适应环境的变化。

结构化开发方法不适合于开发信息需求不明确的系统。

3.3.2 原型法

随着时间的推移、技术的进步, 生命周期法和结构化方法的弊端逐渐暴露出来; 开发过

程烦琐复杂,灵活性较差,系统开发周期长,系统难以适应内外环境变化等。原型法是计算机软件技术发展到一定阶段的产物。原型方法是 20 世纪 80 年代随着计算机软件技术的发展,特别是在关系数据库系统(Relational Data Base System, RDBS)、第四代程序生成语言(4th Generation Language, 4GL)和各种系统开发生成环境产生的基础上,提出的一种从设计思想到工具、手段都是全新的系统开发方法。与结构化系统开发方法不同,原型法不注重对管理系统进行全面、系统的调查与分析,而是本着系统开发人员对用户需求的理解,先快速实现一个原型系统,然后通过反复修改来实现管理信息系统。

1. 原型法的流程

其流程图如图 3-3 所示。

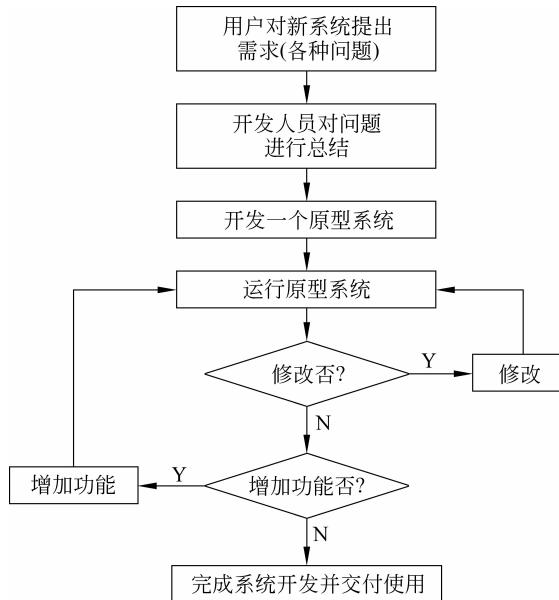


图 3-3 原型法的流程

2. 原型法的特点

应用原型法进行系统开发,有利于用户及早参与开发过程,实实在在的系统原型能给用户和开发人员一个直观的对象,让用户在开发之初就看到系统雏形,了解管理信息系统,便于在系统开发的早期较全面认识和评价系统,从而打破使用与开发的分割状态。也可以使用户培训工作同时启动,有利于系统今后顺利交接和运行维护;构造原型快速,成本较低;开发进程加快,周期缩短,反馈及时。但原型法的应用也存在一些问题:①对于大型系统或复杂性高的系统,没有充分的系统需求分析,很难构造出原型;②开发进程管理复杂,要求用户和开发人员的素质高,配合默契,如果用户合作不好,盲目纠错,就会拖延开发过程;③必须依赖强有力的支撑环境,否则无法进行;④该方法的另一不足是每次反复都要花费人力、物力。

随着面向对象和基于构件的软件开发技术的发展,原型化方法的策略更多地利用构件组合构成,利用各种工具自动生成文档,在许多软件平台上建立交互式原型;更多地运用混

合型原型化策略,多次迭代;更重视多次的评估,请专家、用户、开发者共同完成。

3. 原型法在应用中的问题

作为一种具体的开发方法,原型化方法也有其局限性,在使用时应注意以下几点。

(1) 应将原型法与结构化方法有机结合

在具体的开发中,为了得到有效的开发软件,在整体上可使用结构化方法,以弥补原型法的不足。系统规范化是管理信息系统开发的关键,开发应当做到完整一致和准确。可把原型作为需求描述的补充和量化,以代替传统的数字审核与确认,提高需求描述的质量,可把系统分析设计和建造原型结合起来,在分析的同时考虑设计的要求和目标。

(2) 应当重视开发过程的控制

由于原型化方法缺乏统一规划和对系统开发的分析设计,只是按照“构造原型—修改—再修改”等粗略过程反复迭代,没有约束原型完成和资源分配的标准,常常使开发过程难以控制,项目的管理和系统的维护比较困难。为此,用户和开发者不仅需要达成一个具体的开发协议,规定一些开发的标准和目标,还要建立完整准确的文字档案;特别在每次原型的改进、完善中都必须做相应的文档记录和整理。

(3) 应当充分了解原型化方法的使用环境和开发工具

原型法对开发环境有很高的要求。开发环境包括软件环境、硬件环境和开发人员,最主要的是软件环境。尤其需要支持开发过程中主要步骤的工程化软件支撑环境,以解决原型的快速构造,以及从原型系统到最终系统形成的各种变换以及这些变换的一致性。一般认为,第四代语言(4GL)和软件开发工具是支持原型开发的有力工具,如数据库语言、图形语言、决策支持语言、报表生成器和应用程序生成器等,以及支持软件开发各个阶段的工具系统,如计算机辅助软件工程,这些工具的使用可提高系统开发的效率。

3.3.3 面向对象的开发方法

结构化程序设计的主要缺陷是问题的模型与求解的模型不一致。在客观世界中,实体的内部状态(数据)和运动规律(对数据的操作)是密不可分的,但结构化程序设计缺乏将两者“封装”的机制。这就造成人为的求解空间与客观问题空间的偏离,增加程序设计的复杂性和难度。随着软件不断增大的趋势,矛盾也愈加突出。在此背景下,产生了面向对象的程序设计思想。面向对象的开发方法起源于程序设计语言,但远远超出程序设计的范畴,发展成包括面向对象的系统分析(00A)、面向对象的系统设计(OOD)、面向对象的程序设计(OOP)的方法体系。

面向对象的思想最初出现于仿真语言 Simula。Smalltalk 是 20 世纪 70 年代起源于 Simula 语言的第一个真正面向对象的程序设计语言。C++ 是一种比 Smalltalk 更接于机器,比 C 语言更接近于问题的面向对象的程序设计语言,发展成为标准化的面向对象程序设计语言 Visual C++。其后的集成开发工具,都提供面向对象的开发环境。

1. 面向对象方法的基本概念及特征

面向对象是一种认识客观世界的 worldview,从结构组织角度模拟客观世界的一种方法论。在认识和理解现实世界的过程中,普遍运用以下 3 个构造法则:区分对象及其属性;区分整

体对象及其组成部分;不同对象类的形成及区分。客观世界可以看成由许多不同种类的对象构成,每个对象都有自己的内部状态和运动规律,不同对象间的相互联系和相互作用构成完整的客观世界。

(1) 基本概念

① 对象(Object)。客观世界中(问题空间)任何一个事物都可以看成一个对象。对象是构成系统的基本单位,对象的种类繁多、可大可小。用计算机(求解空间)所创建的对象来对客观事物进行表达和模拟,对象是包括一个事物的状态数据和行为特征的数据模型;即一个对象与一个客观实体相对应,对象是把由实体抽象的状态数据和对数据进行的各种操作封装在一起构成的封装体。例如,一辆汽车是一个对象,它包含了汽车的信息(如颜色、型号、载重量等)及其操作(如启动、刹车等)。一个窗口是一个对象,它包含了窗口的信息(如大小、颜色、位置等)及其操作(如打开、关闭等)。

② 类(Class)。类是具有共同属性、方法、事件的对象的集合;类代表了某一批对象的共性和特征。而一个具体的对象则是其对应类的一个实例(Instance),每个对象都属于一个特定的类。如在 C++ 中,是先声明一个类类型,然后用它去定义若干个同类型的对象。对象就是类类型的一个变量。可以说类是对象的模板,是用来定义对象的一种抽象类型。

如果对某一个类的定义进行修改,如增加一些属性或修改一些方法,就得到一个新的类,而原先的类就是新类的父类。例如,在窗口类的定义上,另外定义它还包含两个按钮,一个标题是“确定”,一个标题是“取消”,并且在分别按下这两个按钮后,执行 OK 与 Cancel 两个事件。这样就定义了一个新类,可以称其为“选择窗口”类,并保存它,这样就可以直接由“选择窗口”生成窗口实例。

在客观世界中存在很多类,这些类之间有一定的结构关系。类有明显的层次结构,通常上层类称为父类或超类,下层类称为子类或派生类。一个类可派生多个子类,父类层有的数据可被多次重用,子类亦可扩展自身的属性与方法。如果子类只有一个超类,则称为单继承性。如果一个子类具有多个超类,则称为多继承性,这时该类共享多个超类的属性及操作。类和继承性是现代软件工程中的重要概念,重用就是通过继承类中的属性和操作而实现的。

③ 消息(Message)。对象之间相互作用和相互协作的一种机制。对象之间的相互操作、调用和应答多是通过发送消息到对象的外部接口来实施的。消息是为完成某些操作而向对象所发送的命令和命令说明。

④ 继承。相关对象类层次之间的一种数据和操作(程序代码)的共享机制。如果类 B 继承了类 A,则在 A 中所定义的数据和操作也将成为 B 的组成部分。

(2) 对象的特征

① 封装性(Encapsulation)。将自由数据与操作(方法)封闭在一起(即放于同个对象中)使自身的状态、行为局部化(对数据的操作只通过该对象本身的方法来进行)。

② 继承性(Inheritance)。通过对类继承可以弥补由封装对象而带来的诸如数据或操作冗余的问题。通过继承支持重用,实现软件资源共享、演化以及增强扩充。

③ 多态型(Polymorphism)。同样的消息为不同的对象接受后,会因不同对象所含操作的不同,而导致完全不同的行动,使软件开发设计更便利,编码更灵活。

④ 可维护性。由于面向对象的抽象封装使对象信息隐藏在局部,当对象进行修改,或对象自身产生错误的时候,由此带来的影响仅仅在对象内部而不会波及其他对象乃至整个