



基于任务驱动的高职软件开发类活页式教材设计研究

黄 涛

(武汉城市职业学院 计算机与电子信息工程学院,湖北 武汉 430064)

摘 要:任务驱动,高职,软件开发,活页式教材,这四个词组合起来就标明了基于任务驱动的高职软件开发类活页式教材的特殊性。在该类教材的设计研究过程中,分析了活页式教材和基于任务驱动的高职软件开发类教材特点,归纳基于任务驱动的高职软件开发类教材结构,融合任务驱动和活页式这两种教材编写方式,提出一个基于任务驱动的高职软件开发类活页式教材的设计方案。

关键词:任务驱动;高职;软件开发;活页式教材

中图分类号: TP311.52-43 文献标识码: A 文章编号: 1671-931X (2019) 06-0062-06

一、活页式教材介绍

活页式教材兼具“活页”和“教材”的双重属性,“活页”具备结构化、形式化、模块化、灵活性、重组性及趣味性等诸多符合教学和自主学习的特征,是其优秀的承载形式,而“教材”具备引导性、过程性、功能性、专业性、综合性等培训特质^[1]。

(一)活页式教材特点

活页式教材最大的特点就是使用灵活,受众面广,主要体现在两个方面。对于教材的各类使用者而言,可以根据自己的需要对教材进行个性化设置,即对教材已有内容进行编排上的调整操作。使用者进行个性化设置的依据是活页式教材提供的教材内容索引表。活页式教材本身所包含的内容应该是该教材所涉及领域里的内容全集,这样使用者才方便从全集中选择内容子集进行定制;对于教材的编者而言,如果后期要更新教材,也可以实现“按需操作”,即对旧教材进行“活页式更新”。

正是由于活页式教材的特点导致了教材编写难

度大,涉及内容多,且目前市面上可供参考的活页式教材极少。

(二)职业教育中使用活页式教材的提出

2019年2月,国务院正式发布了《国家职业教育改革实施方案》。方案中的第九条“坚持知行合一、工学结合”里首次提到了“遴选认定一大批职业教育在线精品课程,建设一大批校企‘双元’合作开发的国家规划教材,倡导使用新型活页式、工作手册式教材并配套开发信息化资源。”

2019年4月,教育部职业教育与成人教育司发布的《职业教育与继续教育2019年工作要点》中的第六点“推进‘三教’改革提高育人质量”里提到了“启动建设‘十三五’职业教育国家规划教材,倡导使用新型活页式、工作手册式教材并配套信息化资源。”

2019年5月,教育部、国家发展改革委、财政部、人力资源和社会保障部、农业农村部 and 退役军人部联合印发了《高职扩招专项工作实施方案》。在实施方案中也提到了“落实立德树人根本任务,加强高

收稿日期:2019-08-20

作者简介:黄涛(1982-),男,湖北武汉人,武汉城市职业学院讲师,研究方向:移动应用开发。

职业院校教师队伍建设,加快补充急需的专业教师,并开发适用于不同生源类型的新型活页式、工作手册式等教材,适应‘互联网+职业教育’发展需求,建好用好职业教育专业教学资源库,促进优质资源共享。”

由此可见,编写和使用活页式教材是今后职业教育的一个发展趋势。

二、基于任务驱动的高职软件开发类教材介绍

基于任务驱动的各类高职教材是现代职业教育中比较主流的一类教材,在教材的编写过程中运用了职业教育里常用的任务驱动教学法。任务驱动教学法是一种以解决问题、教会学生全面思考问题为主的交互式教学,通过每一个具体的任务完成来调动学生学习主动性^[2]。这一过程中,教师需要及时纠正和解决学生在新知识学习和实践中出现的各种问题^[3]。

(一)基于任务驱动的高职软件开发类教材特点

基于任务驱动的高职软件开发类教材就是在任务驱动的基础上为高职软件开发类课程而编写的教材。教材针对于课程,服务于课堂,此类教材的特点就是充分利用任务驱动教学法的优点,使其既能够满足高职学生软件开发类课程学习过程的要求,又提高了高职教师教授软件开发类课程的教学效率,具体体现在以下两个方面。

一方面是学生的学。相比于普通本科院校学生,高职学生普遍基础比较差,对于基础操作易于上手,模仿能力强,对于较复杂知识的学习有一定困难^[4]。而软件开发类课程相比于其它课程又有一定难度,需要学生具有一定的英语基础、记忆力、抽象思维能力、逻辑判断推理能力等综合素质,因此高职学生学习软件开发的一般过程是“模仿-领会-创新”。“模仿”指的是由于软件开发类课程对于高职学生而言普遍具有一定难度,学生在学习的过程中只能先通过模仿代码的编写来进行感性的认识,在模仿编写代码的过程中对代码的语法,规则,技巧等进行“领会”,这是高职学生学习软件开发基础知识的前期过程。在软件开发基础打好的前提下,高职学生能够熟练使用前期学习的软件开发技术实现自己想做的任何功能,这就是“创新”,也就是软件开发类课程的最终目标,让学生能够通过编程自主解决实际问题的能力。通过教材中各种任务的设置正好可以满足学生学习的这三个环节需求,首先通过实操任务进行“模仿”,其次通过完成任务进行“领会”,最后通过进阶任务进行“创新”。

另一方面是教师的教。根据高职学生学习特点,教师在教授过程中必须按照循序渐进,由易到难的方式进行授课,以学生为主导,强调实际操作能力。通过教材中各种任务的设置,可以将枯燥的课程知识点融入到任务中,任务按照由易到难进行安排,让

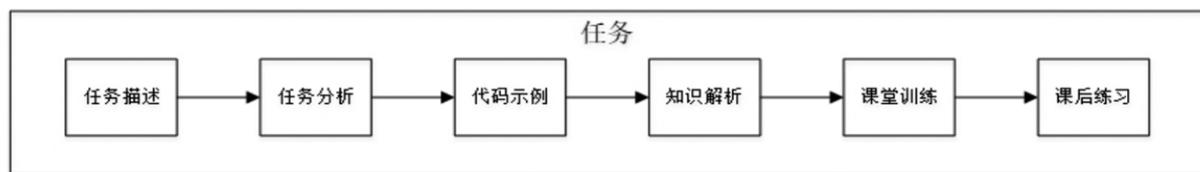


图1 基于任务驱动的高职软件开发类教材结构图

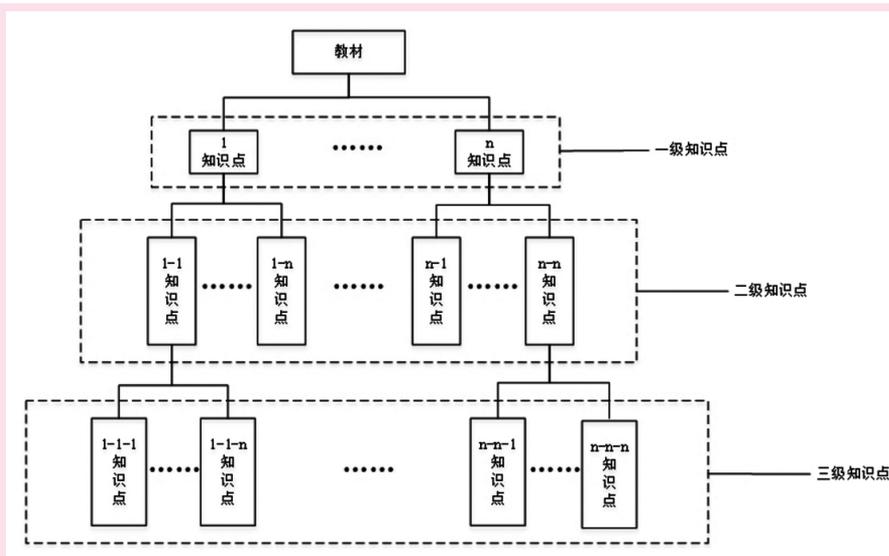


图2 教材知识点划分图

学生主导完成任务，在完成任务的过程中教师进行必要的引导、指导、答疑等工作，这样不仅可以配合学生的软件开发类课程学习过程，教师还可以实时掌握学生的学习情况，及时发现并解决各种共性和个性的问题，提高教学效率。

(二) 基于任务驱动的高职软件开发类教材结构

根据对基于任务驱动的高职软件开发类教材特点的详细论述，此类教材的编写可以遵循图 1 的结构。

上述结构以任务为组织单元，基于任务从前到后依次分为“任务描述”“任务分析”“代码示例”“知识解析”“课堂训练”和“课后练习”这几个要素。其中“任务描述”用于提出任务需求；“任务分析”与“代码示例”是用于学生的“模仿”阶段，教师起引导作用；“知识解析”是用于学生的“领会”阶段，教师起指导作用；“课堂训练”与“课后练习”是用于学生的“创新”阶段，教师起答疑作用。

三、基于任务驱动的高职软件开发类活页式教材的设计

基于任务驱动的高职软件开发类活页式教材在已有基于任务驱动的高职软件开发类教材的基础上

进一步融合了活页式教材的优点，是现阶段最符合职业教育要求的教材类型。此类教材的设计主要分为以下两个阶段。

(一) 准备阶段

在教材设计的准备阶段需要依次完成以下 3 个步骤。

第一，确定教材中包含的所有知识点。考虑到软件开发类课程中包含的知识点具有“多而细”的特点，将知识点从上到下分为三级，便于分层管理。一级知识点指的是纲领性知识点，以开发《Java 面向对象程序设计》教材为例，“类和对象”“基本工具类”“接口、继承与多态”等这些知识点就是一级知识点。二级知识点是在一级知识点的基础上进行细化，比如在“基本工具类”这个一级知识点中包含“包装类”“数字处理类”“其它工具类”等二级知识点。三级知识点则是属于二级知识点中包含的原子性知识点，

表 1 知识点与单元 & 任务对应表

知识点	单元 & 任务	对应关系
一级知识点	单元	一对一
二级知识点	一级任务	一对一
三级知识点	二级任务	一对一、多对一、一对多

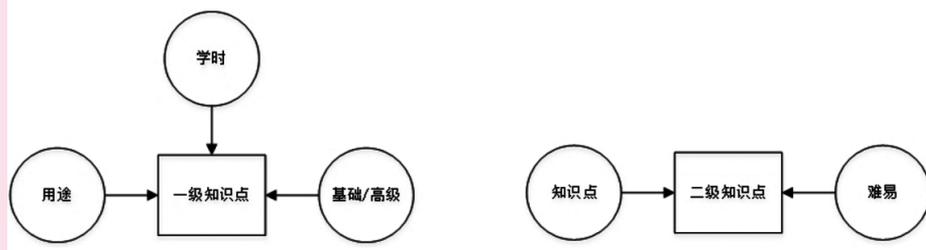


图 3 知识点元数据项

表 2 知识点与单元 & 任务对应关系样例表

教材	单元(一级知识点)	一级任务(二级知识点)	二级任务(三级知识点)
Java 面向对象程序设计	2 基本工具类(2 基本工具类)	2-1 包装类的使用(2-1 包装类)
		2-2 数字处理类的使用(2-2 数字处理类)	2-2-1 使用 DecimalFormat 类格式化整数、浮点数、科学计数法、百分数(2-2-1 数字格式化)
			2-2-2 使用 Math 类进行最大值、最小值、绝对值、平方根和立方根计算(2-2-2 数学运算)
			2-2-3 使用 Math.random 方法产生两数之间的随机偶数和奇数(2-2-3 随机数)
			2-2-4 使用 Random 类产生各种类型的随机数(2-2-4 随机数)
		2-2-5 使用 BigDecimal 和 BigInteger 类构建大数字对象并进行加减乘数运算(2-2-5 大数字运算)	
	2-3 其它工具类的使用(2-3 其它工具类)

比如“数字处理类”这个二级知识点中包含“数字格式化”“数学运算”“随机数”“大数字运算”等众多三级知识点。教材知识点的划分如图2所示。

图2中的一级知识点需根据“基础/高级”按照从基础到高级的顺序进行排序，二级和三级知识点需根据“难易”按照从易到难的顺序进行排序。为了便于教材知识点划分的理解 and 操作，在图2中加入了按照层级依次递增的数字序列，用于唯一标识各级知识点。数字序列中的“n”代表所有自然数，文中出现的所有“n”都是此意。

第二，对一级知识点与二级知识点进行元数据填写。一级知识点进行“用途”、“学时”和“基础/高级”的内容填写，二级知识点进行“知识点”和“难易”的内容填写。这里要注意的是二级知识点中的“知识

点”内容指的是该二级知识点所包含的三级知识点，一级知识点中的“基础/高级”内容和二级知识点中的“难易”内容在前面步骤1中已经确定。具体需填写的知识点元数据项如图3所示。

以教材《Java面向对象程序设计》中的一级知识点“基本工具类”为例，其用途为实现数据类型转换、数字处理、字符串操作、运行时处理、日期操作等功能，学时为4学时，基础/高级为基础；以其中包含的二级知识点“数字处理类”为例，其所含知识点（三级知识点）为数字格式化、数学运算、随机数和大数字运算，难易为易。

第三，根据表1，对知识点设计出单元以及单元中包含的各种任务。

从表1中可以看出，一级知识点与单元一一对

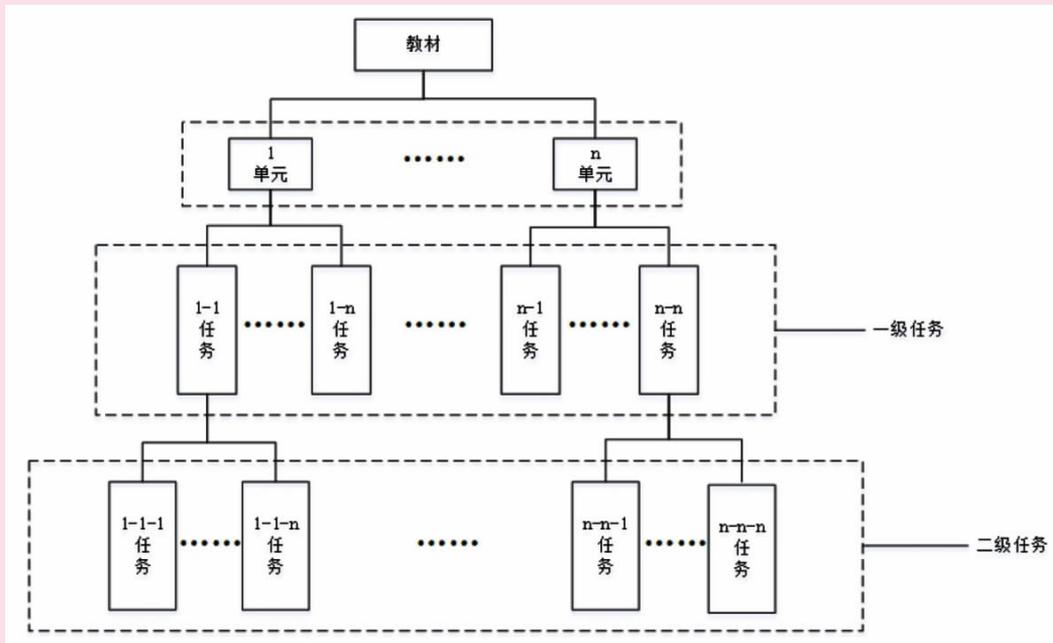


图4 教材单元 & 任务划分图

内容索引表	
单元1 单元名称	} 举例
任务1 任务名称	
任务2 任务名称	
.....	
任务n 任务名称	
课后练习1	
单元2 单元名称	
任务1 任务名称	
任务2 任务名称	
.....	
任务n 任务名称	
课后练习2	
.....	
单元n 单元名称	
任务1 任务名称	
任务2 任务名称	
.....	
任务n 任务名称	
课后练习n	

图5 教材目录结构图

应,任务分为两级,其中二级知识点与一级任务一一对应,三级知识点与二级任务的对应关系则比较灵活,可以一个二级任务包含一个或多个三级知识点,也可以多个二级任务包含一个三级知识点,这主要取决于三级知识点包含内容的多少。

根据表 1 中的对应关系,将图 2 转换成单元和任务形式,如图 4 所示。

为了便于理解和操作教材单元和任务的划分,在图 4 中加入了按照层级依次递增的数字序列,用于唯一标识单元和各级任务。还是以教材《Java 面向对象程序设计》中的一级知识点“基本工具类”和其包含的二级知识点“数字处理类”为线索,根据表 1,其知识点与单元 & 任务对应关系如表 2 所示。假设“基本工具类”为第 2 个单元和第 2 个一级知识点,后面也都按此假设进行。

(二)实施阶段

基于准备阶段 3 步骤中完成的内容,在教材设计的实施阶段需要依次完成以下 4 个框架性内容。

1.设计教材目录结构

综合前面讨论,结合图 1 和图 4,融合了任务驱动与活页式的软件开发类教材目录结构,如图 5 所示。

图 5 中左边部分列出的单元与任务(一级任务)与图 4 中的结构对应,右边部分展示了前面讨论的单元 2“基本工具类”的目录结构样例。

2.设计教材页码编排方式

为了更加方便地在教材中增删和替换内容,特别是方便编者后期对教材进行更新操作,结合教材目录结构,教材页码采用“单元编号—一级任务编号—页码号”三级编排形式。比如:“1-1-1”代表单元 1 中任务 1 的第 1 页。这里需要特殊处理的是教材目录结构中的内容索引表和每个单元包含的课后练习的页码编排方式,其中内容索引表可以看成是特殊的单元,使用“0-0-页码号”方式编排,每个单元的课后练习可以看成是特殊的一级任务,使用“单元编号—X-页码号”方式编排。

3.设计教材内容索引表

需要注意的是表 3 中的“任务”指的是一级任务,“知识点”指的是三级知识点。“单元”,“任务”和“知识点”三者之间的关系是依次从上往下一对多的关系,即一个单元包含多个任务,一个任务包含多个知识点。结合表 1 的对应关系,表 3 中除了“单元页码”和“任务页码”外,其它所有表项内容在准备阶段的第 2 步中已经完成。而“单元页码”和“任务页码”表项内容的填写参考教材页码编排方式即可。表 3 中已有单元 2“基本工具类”的填写样例。

表 3 是活页式教材特有的,目的是方便教师、学生、其他自学者等各类使用者根据表 3 内容进行个性化授课或者学习设置。比如计算机相关的不同专业,在同一门软件开发类课程中的侧重点会有不同,

单元2 基本工具类	
知识目标	
能力目标	
任务1 包装类的使用	
.....	
任务2 数字处理类的使用	
任务2-1 使用 DecimalFormat类格式化整 数、浮点数、科学计数法、 百分数	
任务描述	
任务分析	
代码示例	
任务2-2 使用Math类进行最 大值、最小值、绝对值、平 方根和立方根计算	
任务描述	
任务分析	
代码示例	
.....	
任务2-5 使用BigDecimal和 BigInteger类构建大数字对 象并进行加减乘数运算	
任务描述	
任务分析	
代码示例	
知识解析	
课堂训练	
任务3 其它工具类的使用	
.....	
课后练习2	

图 6 教材单元内容结构图

表 3 教材内容索引表

单元	单元页码	用途	学时	基础/高级	任务	任务页码	知识点	难易
.....
单元 2 基本工 具类	2-0-1	实现数据类型 转换、数字处 理、字符串操 作、运行时处 理、日期操作 等功能	4	基础	任务 1 包装类的使用	2-1-1
					任务 2 数字处理类的使用	2-2-1	数字格式化	易
							数学运算	易
							随机数	易
						大数字运算	易	
					任务 3 其它工具类的使用	2-3-1
.....

此时可以参考表 3 中的“用途”和“知识点”表项进行个性化设置；如果对课程的课时有要求，则可以参考“学时”和“基础/高级”表项进行个性化设置；如果要考虑班级学生的整体水平，则可以参考“难易”表项进行个性化设置。

4. 设计教材单元内容结构

综上所述，结合图 1 和图 4，以单元 2“基本工具类”为例，融合了任务驱动与活页式的软件开发类教材单元内容结构如图 6 所示。

在图 6 结构中，每个独立单元里首先给出本单元的“知识目标”与“能力目标”，用于提前告知使用者学习完本单元后所具有的知识与能力。每个单元会包含 1 到 n 个一级任务，而每个一级任务又会包含 1 到 n 个二级任务，这与图 4 中的结构对应。每个单元的一级任务都会配套“知识解析”与“课堂训练”，其中“知识解析”包含该一级任务中的所有二级任务涉及的知识点，“课堂训练”则是针对这些知识点设置的课堂任务，用于检测知识点的掌握情况。在每个单元的最后会有总的“课后练习”，用于对整个单元的知识点进行检测。每个单元的二级任务独立拥有“任务描述”“任务分析”“代码示例”。在这个结构中，使用者参照表 3 对教材进行个性化设置的最小单位是单元中的一级任务，编者更新教材时同样也是。

有了以上教材中的框架性内容，剩下具体的任务内容、知识点、习题等，则根据教材对应的软件开

发课程进行填充即可。在填充工作开始之前，需要根据设计的框架性内容收集与整理各种素材。这些是最基础最普通的教材编写工作，在此不再展开讨论。

四、结束语

本文简要探讨了如何进行基于任务驱动的高职软件开发类活页式教材的设计，提出了一种设计方案，其中重点阐述了教材中框架性内容的设计过程。目前此类教材的编写尚处于探索期，在高职软件开发类教材中由于同时融合了任务驱动和活页式这两种教材编写方式，对教材编写者的要求较高，如何进行更优的设计来真正发挥此类教材的优势，更好地服务于高职软件开发类课程，是需要持续研究的内容。

参考文献：

- [1] 吴振东. 试论“活页教材+活页笔记+功能插页”三位一体自主思维模式的构建[J]. 新课程研究, 2018, (9): 62-66.
- [2] 沈爱凤, 韩学芹. 职业教育中“任务驱动式”教学模式的探讨与应用[J]. 职教论坛, 2016, (2): 46-49.
- [3] 张萍. 基于任务驱动理念的高职计算机基础教学探讨[J]. 教育与职业, 2015, (6): 159-160.
- [4] 黄涛. 高职软件开发实训类课程的设计研究[J]. 武汉职业技术学院学报, 2016, (6): 78-81.

[责任编辑：向 丽]

Research on Design of Loose-Leaf Textbook about Software Development for Higher Vocational College Based on Task-Driven

HUANG Tao

(School of Computer and Electronic Information Engineer, Wuhan City Polytechnic, Wuhan 430064, China)

Abstract: This paper analyses the features of loose-leaf textbook and software development series textbook for higher vocational college based on task-driven, generalizes the structure of task-driven higher vocational software development textbooks, and integrate task-driven and loose-leaf textbooks, proposes a design of loose-leaf textbook about software development for higher vocational college based on task-driven.

Key words: task-driven; higher vocational college; software development; loose-leaf textbook