ICS 03.180
A 18

CERACU

全国高等学校计算机教育研究会团体标准

T/CERACU 204—2020

计算机核心课程规范
程序设计

Specification for Core Courses of Computer Science—

Programming

2020-11-27 发布　　2021-05-01 实施

全国高等学校计算机教育研究会　发布

目　　次

前言 2
1 范围 3
2 概述 3
3 课程目标与总体要求 4
3.1 课程目标与教学要求 4
3.2 考核要求 4
4 基础知识点及要求 4
5 面向对象程程序设计扩展知识点及要求 7
6 泛型程序设计扩展知识点及要求 10
7 算法设计扩展知识点及要求 11
8 事件驱动的程序设计模式扩展知识点及要求 12

前　言

本文件按照GB/T 1.1－2020给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由全国高等学校计算机教育研究会（CERACU）提出并归口。

本文件起草单位：清华大学，吉林大学，北京航空航天大学，东北财经大学，天津师范大学，北京理工大学，深圳点猫科技有限公司，四川师范大学，西北农林科技大学，北京工商大学，北京工业大学，空军工程大学，滁州学院，清华大学出版社，华南师范大学，青海大学，首都医科大学，复旦大学，济南大学，南通理工学院，上海畅学教育科技有限公司，首都师范大学附属学校，长安大学，韩山师范学院。

本文件主要起草人：郑莉，徐明星，张睿，李莹，李超，陶永明，张珑，陈朔鹰，孙悦，李泽，廖雪花，耿楠，张珣，蔡越江，张红梅，嵩天，于春燕，刘洪，谢琛，邓富强，王洪江，刘晓静，夏翊，陈荣华，蒋彦，宋伟，佟钢，夏立，刘明非，吕进，郑耿忠。

计算机核心课程规范
程序设计

1 范围

本文件给出了高等学校计算机程序设计课程的知识点集合、知识体系结构，规定了课程的教学要求。

本文件适用于高等学校计算机程序设计课程、教材、教学过程和教学评价。

2 概述

本文件主要包括与具体计算机程序设计语言无关的基础知识点及要求、算法设计的扩展知识点及要求、面向对象程序设计的扩展知识点及要求、泛型程序设计的扩展知识点及要求、事件驱动的程序设计扩展知识点及要求。

知识体系按照分层定义，结构层级不超过三级，每个知识点可以容纳下级知识点，也可以是原子知识点。能力要求给出一级、二级、三级三个层次递进的具体要求。学时要求是应至少达到的学时下限，不设上限，包括总学时、理论学时（包括听课、自学等各种形式）、实践学时（包括实验、项目实训等各种形式），“其他”可根据不同语言知识点的需要进行扩展。

知识点属性如表1所示。

表1 知识点属性

属性项	数据类型	取值范围	可选（O）/必备（M）	说明
标识符	字符串	长度不超过20	M	在本文件中唯一标识知识点
名称	字符串	长度不超过40	M	应采用行业内共识的知识点名称
说明	字符串	长度不超过400	O	说明知识点内涵。非原子知识点O，原子知识点M
能力要求	容器类型	包含一级、二级、三级三个子项	M	
一级	字符串	长度不超过1000	O	说明某知识点“一级”要求的具体含义
二级	字符串	长度不超过1000	O	说明某知识点“二级”要求的具体含义
三级	字符串	长度不超过1000	O	说明某知识点“三级”要求的具体含义
学时要求	容器类型	包括理论、实践、其他三个子项	O/M	顶层知识点为M；非顶层知识点的为O
总学时	整数	大于0	O/M	O/M规则同“学时要求”。此项为生成项，等于理论、实践、其他三项之和
理论	整数	大于或等于0	O/M	O/M规则同“学时要求”。理论、实践、其他至少一项不为0，且三项总和与总学时相等
实践	整数	大于或等于0	O/M	O/M规则同“学时要求”。理论、实践、其他至少一项不为0，且三项总和与总学时相等
其他	整数	大于或等于0	O/M	O/M规则同“学时要求”。理论、实践、其他至少一项不为0，且三项总和与总学时相等

3 课程目标与总体要求

3.1 课程目标与教学要求

课程目标应服从各个学校、院系的培养方案。本文件不规定具体课程的教学要求，宜从本文件中选择适用的知识点和能力要求、学时要求组成一门具体课程的内容和要求，应包括本文件第4章规定的知识点80%以上，能力要求为“二级”、“三级”的应占所选知识点的80%以上。本文件中未包含的知识点应参照本文件规定的格式进行描述。课程目标与教学要求应坚持辩证唯物主义，坚持理论和实践的辩证统一，安排注重学生创新思维和实践动手能力培养。

3.2 考核要求

课程考核要求应服从各个学校、院系的培养方案和教学实际需求。

考核包括学习过程中的形成性考核与课程总评考核。考核知识点应覆盖所有被选择作为该课程内容的知识点，考题难度应与该课程所选的各知识点能力要求、学时要求相符。每道题目应明确对应到一个或多个知识点及能力要求。各知识点题目比例应与学时比例一致，含多个知识点的综合题目宜按学习次序靠后的知识点和难度较大的一个或多个知识点计。

考核形式应适合知识点的内容和能力要求。

4 基础知识点及要求

基础知识点及要求如表2所示。

表2 基础知识点及要求

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_FD_01	基础语法		能够看懂只包含基础语法的简单程序	能够应用基础语法编写简单程序		2	1	1	0
PROG_FD_01_01	顺序结构的程序		初步认识最简单的程序	能够仿照例题编写最简单的程序					
PROG_FD_01_02	字符集		知道字符集的概念和定义的语法	能够在程序中正确使用字符集					
PROG_FD_01_03	词法记号		知道词法记号的概念和定义的语法	能够在程序中正确使用词法记号					
PROG_FD_01_04	标准输入/输出	关联到标准设备的输入与输出	能够以默认格式进行标准输入/输出	能够控制标准输入/输出的宽度、精度等简单格式	能够应用语言规定的所有格式控制方式进行标准输入/输出；能够进行标准输入/输出重定向				
PROG_FD_01_05	语句		知道语句的语	能够正确使					

表2 基础知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
			法	用语句					
PROG_FD_02	数据类型		知道数据类型的概念和基本使用	能够按照设计要求定义数据的类型，并使用各种类型的数据	能够针对问题需求选择恰当的数据类型，并正确使用	1	0.5	0.5	0
PROG_FD_02_01	基本数据类型	程序语言语法内置的数据类型	知道基本数据类型的概念和基本使用方法	能够按照要求使用基本数据类型处理数据	能够根据需求选择使用恰当的数据类型处理数据				
PROG_FD_02_02	类型转换		知道各种数据类型之间的转换规则	能够按要求正确进行数据类型转换	能够通过数据类型转换解决实际问题				
PROG_FD_03	数据的存储与内置运算		知道常量、变量和内置运算的概念和语法	能够按照要求使用常量、变量和内置运算	能够根据需求选择使用常量、变量和各种内置运算符实现各种运算	2	1	1	0
PROG_FD_03_01	常量		知道常量定义和使用的语法	能够按照要求定义和使用常量	能够按照问题需求设计和使用常量				
PROG_FD_03_02	变量		知道变量定义和使用的语法	能够按照要求定义和使用变量	能够按照问题需求设计和使用变量				
PROG_FD_03_03	内置运算		知道各种内置运算符的优先级和结合性	能够按照要求使用内置运算符实现运算	能够按照问题需求使用内置运算符实现运算				
PROG_FD_03_04	表达式		知道表达式的构成语法	能够按要求使用常量、变量和内置运算符构成表达式	能够根据运算需求设计表达式				
PROG_FD_04	流程控制		知道选择结构与循环结构	能够按照要求使用选择与循环结构	能够根据问题需求设计与实现程序的流程控制	4	2	2	0
PROG_FD_04_01	选择语句	语法规定的选择语句和开关语句	知道选择结构的概念和语法	能够根据要求使用合适的选择结构编写程序	能够根据问题需求设计和实现多重选择结构的程序	1.5	1	0.5	0
PROG_FD_04_02	循环语句	语法规定的各种循环语句及其嵌套	知道循环结构的概念和语法	能够根据要求使用合适的循环结构编写程序	能够根据问题需求设计和实现嵌套循环结构的程序	2	1	1	0

表2 基础知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_FD_04_03	其他控制语句		知道其他控制语句的语法	能够根据问题要求恰当选择使用		0.5	0.5	0	0
PROG_FD_05	功能模块	表达程序功能的基本单位。例如：函数、方法	知道功能模块的定义和使用方法	能够按照要求定义和使用功能模块	能够根据问题的需求设计和实现模块，并构建模块间的关系	4	2	2	0
PROG_FD_05_01	定义		知道功能模块的定义语法	能够按照要求定义功能模块	能够根据问题的需求设计功能模块并实现	1	0.5	0.5	0
PROG_FD_05_02	调用		知道功能模块的调用语法	能够按照要求调用功能模块	能够根据问题的需求进行功能模块调用，以及模块的嵌套和递归调用	2	1	1	0
PROG_FD_05_03	功能模块间的数据共享	例如函数调用时的数据传递	知道功能模块间数据共享的语法	能够按照要求实现功能模块间数据共享	能够根据问题的需求设计和实现功能模块间的数据共享	1	0.5	0.5	0
PROG_FD_06	数组		知道数组的概念和语法	能够按照要求定义和使用数组	能够根据问题的需求设计和使用数组	4	2	2	0
PROG_FD_07	地址与指针(C/C++)		知道地址类型和指针的概念和语法			3	1.5	1.5	0
PROG_FD_07_01	地址运算		知道地址运算语法	能够按照设计要求使用地址运算	能够根据问题的需求恰当使用地址运算				
PROG_FD_07_02	指针类型数据的定义		能够定义指针类型常量和变量	能够按照设计要求定义和使用指针	能够根据问题的需求恰当使用指针				
PROG_FD_07_03	指针运算	包含指针的算术运算、关系运算和赋值运算	知道指针运算语法	能够按照设计要求使用指针运算	能够根据问题的需求恰当使用指针运算				
PROG_FD_08	引用（C++/Java）		知道引用的语法	能够按照设计要求使用引用	能够根据问题的需求恰当使用引用	1	0.5	0.5	0
PROG_FD_09	字符串		知道字符串存储与表示的语法	能够按照要求使用字符串	能够根据问题的需求定义和使用字符串	2	1	1	0
PROG_FD_10	输入/输出	实现数据的输入/输出，例如：文件操	知道输入/输出概念和处理方法	能够根据要求实现输入/输出与文件	能够理解问题需求，设计和实现合适的输入/输	4	2	2	0

表2 基础知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
		作、数据持久化		操作	出与文件操作				
PROG_FD_10_01	输入/输出的基本概念和语法		知道输入/输出的概念、语法和标准库基本操作	能够根据要求实现输入/输出	能够理解问题需求，设计和实现合适的输入/输出	1	0.5	0.5	0
PROG_FD_10_02	文件操作		知道文件处理的语法和标准库基本操作	能够根据要求实现文件操作	能够理解问题需求，设计和实现合适的文件操作	3	1.5	1.5	0
PROG_FD_11	标准库	程序语言标准/规范所规定的库	知道标准库的结构和调用方法	能够根据要求使用标准库中的预定义模块	能够理解问题需求，在解决方案中有效利用标准库中的预定义模块	3	1.5	1.5	0
PROG_FD_11_01	标准库组成	库的主要组成部分	知道标准库的组成结构	能够根据要求选择使用标准库中的相应模块		0.5	0.5	0	0
PROG_FD_11_02	标准库调用	使用标准库的语法	知道语法上如何使用标准库中的预定义模块	能够根据要求在程序中使用标准库中的预定义模块	能够理解问题需求，有效选择和利用标准库中的预定义模块，设计恰当有效的解决方案	2.5	1	1.5	0
PROG_FD_12	程序开发过程与工具环境	程序的编辑、编译链接（或解释）等过程，以及相应的开发环境/工具	知道程序的开发过程，知道至少一种开发环境及其中的工具集	能够使用至少一种开发环境及其中的基本工具集，完成简单程序的开发过程	能够根据需要，选择合适的开发环境，精通多种工具，完成复杂程序的开发和调试	1	0	1	0
PROG_FD_12_01	程序的开发过程	程序的编辑、编译链接（或解释）等过程	知道程序的开发过程	能够完成简单程序的开发和调试	能够完成复杂程序的开发和调试	0.5		0.5	0
PROG_FD_12_02	开发环境/工具		知道至少一种开发环境及其中的工具集	能够使用至少一种开发环境及其中的基本工具集	能够根据需要，选择合适的开发环境，精通多种工具	0.5		0.5	0

5 面向对象程程序设计扩展知识点及要求

面向对象程程序设计扩展知识点及要求如表3所示。

表 3 面向对象程程序设计扩展知识点及要求

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_OO_01	类与对象		知道类与对象的概念、语法和用途	能够按照要求定义和使用类和对象	能够对问题进行抽象，设计类结构和定义类，根据需求定义和使用对象	3	1.5	1.5	0
PROG_OO_01_01	类的概念和定义	类定义的语法，以及抽象和封装的思想	知道类的概念和类定义的语法	能够按照要求定义和使用类	能够对问题进行抽象，设计类结构、定义和使用类				
PROG_OO_01_01_01	类的成员	包括数据成员或属性、函数成员或方法	知道类成员定义的语法	能够按照要求定义和使用类的成员	能够对问题进行抽象，设计、定义和使用成员				
PROG_OO_01_01_02	成员的访问方式和权限	类定义时规定的可访问成员的方式和在不同范围的访问权限	知道访问成员的方式和访问权限定义的语法	能够按照要求定义成员的访问方式和权限	能够对问题进行抽象，设计和定义成员的访问方式和权限				
PROG_OO_01_01_03	函数/方法重载	在同一作用域范围内，定义名字相同，但参数表不同的函数，包括全局函数重载和类成员函数重载	知道函数/方法重载的概念和语法	能够按照要求定义重载函数/方法	能够根据问题需求定义和使用重载函数/方法				
PROG_OO_01_02	对象的概念和定义	对象定义的语法，对象与类的关系，对象在程序中的作用	知道对象的概念、定义和使用	能够按照要求定义和使用对象	能够根据问题需求定义和使用对象				
PROG_OO_01_03	对象的初始化和删除	对象初始化的意义和作用，初始化方法，对象不再使用时被删除的方法及其意义	知道对象可以初始化和删除	能够按照要求对对象进行初始化和删除	能够根据问题需求，设计和实现对象的初始化和删除				
PROG_OO_01_04	对象的复制		知道对象复制的概念和方法	能够按照要求复制对象		0.5	0.25	0,25	0
PROG_OO_01_05	对象在函数或方法间共享	通过类的封装、函数或方法参数和返回值等，在内存中共享对象	知道对象在函数或方法间共享的语法规定	能够按照要求定义共享机制	能够根据问题需求，设计和实现对象在函数或方法间的共享				
PROG_OO_02	类的重用	利用已有的类设计新类	知道类重用的基本概念和语	能够按照要求定义类的重用关系	能够根据问题需求，设计和实现类的重用	8	4	4	0

表3 面向对象程程序设计扩展知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
			法						
PROG_00_02_01	类的组合	利用部件类的对象作为成员设计组合类	知道类组合的基本概念和语法	能够按照要求定义类的组合关系	能够根据问题需求，设计和实现类的组合	2	1	1	0
PROG_00_02_01_01	组合类的定义	定义含有对象成员的类	知道组合类定义的语法	能够按照要求实现组合类的定义	能够根据问题需求，设计和实现类的组合				
PROG_00_02_01_02	组合类对象的初始化和删除	组合类对象初始化和删除时基本类型成员和对象成员的处理（不需要考虑初始化或删除的语言除外）	知道组合类对象初始化时要对成员对象进行初始化	能够按照要求在类定义时设计和实现基本类型成员和对象成员的初始化和删除	能够根据问题需求，设计和实现组合类对象的多种初始化和删除方法				
PROG_00_02_02	类之间的引用	在类成员函数或方法参数表、局部对象等处引用其他类	知道类之间引用的语法	能够按照要求实现类之间的引用关系	能够根据问题需求，设计和实现类之间的引用关系	0.5	0.25	0.25	0
PROG_00_02_03	类的继承	在继承基类或超类的基础上设计新的类（派生类或子类）	知道类继承的概念、语法和应用	能够按照要求实现类之间的继承关系、对象的初始化和删除	能够根据问题需求，设计和实现类之间的继承关系、对象的初始化和删除	3	1.5	1.5	0
PROG_00_02_03_01	继承关系的定义	定义类与类之间的继承关系	知道类继承的语法	能够按照设计要求设计和实现类之间的继承关系	能够根据问题需求，设计和实现类之间的继承关系				
PROG_00_02_03_02	派生类对象的初始化和删除	在对象诞生和消亡时进行初始化和删除（不需要考虑初始化或删除的语言除外）	知道对象诞生时需要初始化、消亡时需要删除	能够按照要求实现对象初始化和删除	能够根据问题需求，设计和实现对象初始化和删除				
PROG_00_02_04	多态性	包括静态多态性和动态多态性	知道多态性的概念、动态多态性语法和用途	能够按照要求实现具有动态多态性的程序	能够根据问题需求，设计和实现具有动态多态性的程序	2.5	1.25	1.25	0
PROG_00_02_04_01	多态性的概念和语法	多态性的概念含义、语法	知道多态性的概念和语法	能够按照要求实现具有多态性的程序	能够根据问题需求，设计和实现具有多态性的程序				
PROG_00_02_04_02	多态性的应用	多态性的应用场合、解决的问题和效果	知道多态性的用途	能够编写程序，实现设计中的多态性意图	能够理解问题需求，通过多态性使解决方案更优				
PROG_00_03	异常处理	在程序设计语言中用于描述	知道异常处理的概	能够按照要求实现异常处理	能够理解问题需求，通过异	2	1	1	0

表3 面向对象程程序设计扩展知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
		异常与异常处置而用的语言机制	念和语法		常处理机制增强程序的容错能力				
PROG_OO_03_01	异常处理的基本思想	异常处理的目的、意义和用途	知道异常处理的基本概念和使用	能够理解设计中的异常处理要求	能够理解问题需求，识别对异常处理的需求	0.5	0.5	0	0
PROG_OO_03_02	异常处理的语法	程序设计语言中对异常处理机制的规定	知道异常处理的语法	能够根据设计中的异常处理要求实现异常处理	能够理解问题需求，识别对异常处理的需求并实现异常处理	1.5	0.5	1	0

6 泛型程序设计扩展知识点及要求

泛型程序设计扩展知识点及要求如表4所示。

表4 泛型程序设计扩展知识点及要求

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_GN_01	泛型程序设计		知道泛型程序设计的概念、语法和使用	能够按照要求定义和使用泛型	能够根据问题需求，设计和实现泛型程序	2	2	1	0
PROG_GN_01_01	函数模板的概念和定义	函数模板的语法，类型参数化的思想	知道函数模板的概念和函数模板定义的语法	能够按照要求定义和使用函数模板	能够根据问题需求，设计和实现函数模板				
PROG_GN_01_02	类模板的概念和定义	类模板定义的语法，对不同数据类型的类进行抽象	知道类模板的概念和定义、使用的语法	能够按照要求定义和使用类模板	能够根据问题需求，设计和实现类模板				
PROG_GN_01_03	模板的实例化	实例化的意义；隐式实例化与显式实例化的区别	知道模板需要实例化后才能使用	能够区分隐式实例化与显式实例化	能够根据问题需求，在多文件下组织模板，并实例化				
PROG_GN_01_04	模板的特殊实现	模板的特化、类模板的偏特化、函数模板的重载的定义与语法	知道可以实现模板特化，了解其作用与意义	能够根据要求实现类模板的偏特化、函数模板的重载	能够根据问题需求，设计和实现特殊模板				
PROG_GN_02	泛型应用					3	2	1	0
PROG_GN_02_01	容器的概念与定义	容器的定义和使用的语法；提供存储和组织元素的方法	知道容器是一组元素的对象；知道容器的分类与作用	能够按照要求定义和使用容器	能够根据问题需求，选择和使用合适的容器				

表4 泛型程序设计扩展知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_GN_02_02	迭代器的概念与定义	迭代器的定义和使用的语法；提供访问元素方法	知道迭代器提供访问元素的方法；知道迭代器的分类与作用	能够按照要求定义和使用迭代器	能够根据问题需求，选择和使用合适的迭代器				

7 算法设计扩展知识点及要求

算法设计扩展知识点及要求如表5所示。

表5 算法设计扩展知识点及要求

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_AL_01	逻辑推理	基于关系运算、逻辑运算，通过枚举进行逻辑推理问题求解	知道枚举思想的可行性	能用循环语句 来对具体问题的各种可能性进行枚举	能用关系运算和逻辑运算表达具体问题中的各种条件，能用循环语句列举和测试所有可能性	3	2	1	
PROG_AL_02	筛法	高效找出一定范围内质数的算法	知道筛法的数学原理	理解筛法实现的步骤	能将筛法运用于类似找质数的场景	2	1	1	
PROG_AL_03	递推	通过递推公式对问题建模和求解	知道递推求解的实现方法	能分析出问题中明显存在的递推关系，写出递推的数学公式	能分析出复杂的递推关系，能以循环实现递推求解	2	1	1	
PROG_AL_04	递归	通过函数自调用的方式，不断缩小问题规模直到问题可以直接得到解决	知道递归算法的基本形式，理解递归过程中问题规模的变化	能对经典递归问题进行分析，写出正确的函数接口与功能定义	能分析问题是否具能递归求解，理解递归运行的机制，能用递归思想解决复杂的递归问题	9	6	3	
PROG_AL_05	贪心法	通过一系列局部最优的决策来计算整个问题的最优解	知道贪心法的求解过程、算法特点	对于适用贪心法的问题，能用贪心法求解	能分析判断问题是否适用贪心法，能用贪心法求解问题	2	1	1	
PROG_AL_06	动态规划	一种通用的算法技术，用于解决由交叠子问题	知道动态规划算法的基本形式和算法特点	能从问题场景出发，推导出递推关系式并实现	能以递推和递归两种形式实现动态规划算法	3	2	1	

表5 算法设计扩展知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
		构成的问题		算法					
PROG_AL_07	宽度优先搜索	常见解题思路，在每一步尝试可能选项时，把所有当前合法可行的选项都尝试一下	知道宽度优先搜索的含义和算法特点	能基于队列来实现宽度优先算法	能深入发现问题特点以判断是否适用宽度优先搜索；能高效实现宽度优先搜索	3	2	1	
PROG_AL_08	深度优先搜索	常见解题思路，在每一步尝试时，根据一定准则挑选一个选项进行尝试	知道深度优先搜索的含义和算法特点	能基于递归思想实现深度优先算法	能根据问题特点来判断是否适用深度优先搜索；能高效实现深度优先搜索	3	2	1	

8 事件驱动的程序设计模式扩展知识点及要求

事件驱动的程序设计模式扩展知识点及要求如表6所示。

表6 事件驱动的程序设计扩展知识点及要求

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
PROG_EV_01	事件驱动程序设计	事件的基本概念、事件源、事件对象、事件处理器、监听与广播模式、事件响应机制的实现	知道事件的基本概念和使用	能够根据要求，实现事件源、事件对象、事件处理器的基本应用	能够根据问题需求，设计和实现监听与广播模式、事件响应机制	2	2	0	0
PROG_EV_01_01	监听与广播机制	监听与广播机制的概念以及应用	知道监听与广播机制的基本概念和使用	能够根据要求，实现监听与广播机制的典型应用	能够根据问题需求，设计和实现监听与广播机制				
PROG_EV_01_02	事件	事件的概念以及应用	知道事件的基本概念和使用	能够根据要求，实现事件的定义和使用					
PROG_EV_01_03	事件源	事件源与事	知道事件源	能够根据要	能够根据问				

表6 事件驱动的程序设计扩展知识点及要求（续）

标识符	名称	说明	能力要求			学时要求			
			一级	二级	三级	总	理论	实践	其他
	与事件监听器	件监听器的概念以及应用	与事件监听器的基本概念和使用	求，实现事件源与事件监听器	题需求，设计和实现事件源与事件监听器机制				
PROG_EV_02	事件驱动应用					3	2	1	0
PROG_EV_01_01	委托		知道委托的基本概念以及与事件的联系	能够根据要求，实现委托	能够根据问题需求，设计和实现委托				
PROG_EV_01_02	组件交互		知道组件交互的概念和应用	能够根据要求，实现基于事件驱动的组件交互	能够根据问题需求，设计和实现基于事件驱动的组件交互				