# 《软件工程的形式化方法》教学大纲

**一、课程概述**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程名称** | 软件工程的形式化方法 | **英文名称** | Formal Method of Software Engineering |
| **课程性质** | 专业必修课 | **课程代码** | 22126021 |
| **总学时** | 48学时理论32学时+实验16学时 | **学分** | 2.5 |
| **开课学期** | 第七学期 | **先修课程** | 离散数学、程序设计基础、软件工程导论 |
| **适用专业** | 软件工程 | **开课单位** | 计算机与电气工程学院 |

**二、课程简介**

本课程是软件工程专业高年级开设的一门专业必修课。形式化方法是基于严格数学基础，对计算机软(硬)件系统进行形式规约、开发和验证的一系列技术，用以保障系统的可靠性。形式化方法与计算机科学理论密切相关，其发展与程序设计语言和程序理论的发展息息相连。在现代软件的开发过程中，形式化方法在不同的阶段、以不同的形式和程度得以应用，并形成了以模型检测、定理证明、进程代数等特色技术，近年来在操作系统、编译软件、航空控制软件等研发关键领域取得成效。

形式化方法理论体系复杂、技术方案多样，作为本科生课程，本课程主要讲授形式化方法最核心的原理和技术，如逻辑系统、自动机理论、程序语言理论，同时注重实践与应用，重点介绍当前流行的形式化验证工具的使用方法，如模型检测代表性工具UPPAAL、定理证明器Isabella等。通过本课程学习，有助于拓宽学生专业认知，能帮助学生正确认识计算机科学的本质，理解程序正确性乃至软件可靠性，有助于学生的后续学习和未来工作，使其成长为超越前人的新一代计算机工作者。

**三、课程目标**

**课程目标1：（理论）**具备基本的预备知识，如集合、字符串和语言、图、可计算性和计算复杂度等；能理解一阶逻辑系统及其逻辑公式的证明；能理解自动机模型，以及其与语言的对应性。（**支撑毕业要求1.1**）

**课程目标2：（方法）**能运用自动机理论开展问题建模，利用线性时序逻辑等提出规约，能指出模型检测、定理证明等方法的优缺点及适用范围。（**支撑毕业要求4.1**）

**课程目标3：（工具）**能根据实际问题，对比并选用合适的形式化验证工具；能运用模型检测工具对软件建模、规约并验证；能运用定理证明器对小规模软件系统开展分析和验证。（**支撑毕业要求5.2**）

**课程目标4：（探索）**熟悉形式化方法研究组织和机构；能找到核心的形式化方法领域文献来源；能通过自主学习了解软件工程研究前沿；培养富有专业热情和探索精神的新一代工程师。（**支撑毕业要求12.1**）

**四、课程目标对毕业要求指标点的支撑**

**表4-1 课程目标对毕业要求指标点的支撑**

| **毕业要求** | **毕业要求指标点** | **课程目标** |
| --- | --- | --- |
| **1** | **2** | **3** | **4** |
| 1.工程知识的应用 | 1.1能够用数理科学和软件工程的语言工具描述软件工程领域复杂工程问题的表达。 | H |  |  |  |
| 4.专业问题的研究 | 4.1能够基于科学原理，通过文献研究或相关方法，调研和分析工程领域的复杂工程问题的解决方案和实验方法。 |  | H |  |  |
| 5.现代工具的使用 | 5.2 能够开发、选择和使用恰当的技术、资源和工具对复杂软件工程问题进行分析、预测和模拟。 |  |  | H |  |
| 12.终身学习 | 12.1能认识到软件工程技术发展的快速性及应用领域的多样性和必要性，具有自主学习和终身学习的意识。 |  |  |  | H |

注：分别用“H、M、L”对应表示“高、中、低”支撑。

**五、教学内容及实施手段**

**表5-1教学内容与进度要求**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **章号** | **小节内容** | **要求** | **具体要求** | **学生成果** | **课程****目标** | **学时** |
| 一、形式化方法概貌 | (1)形式化方法起源和发展  | 认知 | 通过经典文献带读，了解形式化方法的起源和发展，尤其近年来重要的应用成果 | 能意识到关键领域软件的可靠性不能依靠测试来解决，形式化方法具有独到作用 | 4 | 2 |
| (2)形式化方法的理论基础 | 介绍形式化方法的理论基础，主要建立在数理逻辑基础上，涵盖众多图灵奖得主的贡献，如霍尔逻辑、时序逻辑、模型检测理论等 |
| (3)形式化方法的技术和工具 | 主要介绍当前成熟的模型检测理论及代表性工具UPPAAL、SPIN等，定理证明器工具Isabella、coq等，进程代数CCS、CSP等 |
| 二、预备知识 | (1)集合论、关系、函数、图、字符串和语言、布尔逻辑 | 理解 | 介绍集合、关系和图的定义和运算规则等 | 能熟练使用集合的语言描述简单问题模型 | 1 | 2 |
| (2)自动机与语言 | 理解 | 介绍有穷自动机的形式化定义及举例 | 能运用有穷自动机对简单问题建模 | 1 | 2 |
| 介绍正则语言，以及与有穷自动机的等价性 |
| (3)可计算性和计算复杂度 | 认知 | 通过图灵机模型介绍可计算性问题，揭示计算机的本质 | 能理解可计算问题 | 1 | 2 |
| (4)λ演算和函数式语言 | 理解 | 介绍λ演算模型以及由此演化出来的函数式编程语言，阐述函数式语言的特点，如柯里化、高阶函数在编程中的应用等 | 能区分不同范式的编程语言，能说出主流编程语言的源流 | 4 | 2 |
| 三、逻辑和定理证明 | (1)命题逻辑和一阶逻辑 | 理解 | 介绍命题逻辑形式化定义及其证明 | 能理解简单形式化逻辑系统及其证明 | 1 | 2 |
| 介绍一阶逻辑形式化定义及其证明 |
| (2)程序证明简介 | 理解 | 通过对比程序证明和软件测试凸显程序证明的重要性，介绍程序证明的真实应用 | 能理解关键领域软件可靠性问题 | 2 | 2 |
|  | (3)函数式程序证明及其工具 | 理解 | 通过主流定理证明器软件的介绍，引入函数式程序证明，如Isabella、Coq等 | 能操控一种定理证明软件GUI，明白其运行机制 | 3 | 2 |
| 四、软件系统建模 | (1)顺序系统、并发系统和反应式系统简介 | 理解 | 通过常用软件系统的介绍，理清主要的软件模型及其建模方法  | 能识别软件模型的类别及特点 | 2 | 2 |
| (2)形式化建模工具 | 通过主流的形式化验证工具，如UPPAAL等，针对实例进行建模演示 | 能运用模型检测工具针对实际问题开展建模 | 3 | 2 |
| 五、形式化规约 | (1)线性时序逻辑及其示例 | 理解 | 介绍线性时序逻辑及其公理化方法 | 能利用模型检测工具针对实例创建规约，并有效描述其意义 | 2 | 2 |
| (2)模型检测中的规约 | 利用主流模型检测工具，针对实例，创建形式化规约 | 3 | 2 |
| 六、自动验证 | 模型检测原理及范例 | 认知 | 介绍模型检测原理 | 能利用模型检测工具对软件建模并开展自动验证 | 2 | 2 |
| 通过模型检测工具对实例开展自动化验证，分析其利弊 |
| 七、演绎式软件验证 | (1)程序证明的逻辑基础 | 认知 | 对比介绍希尔伯特系统以及自然演绎系统 | 能利用定理证明器分析简单的逻辑系统 | 4 | 2 |
| 通过引入主流定理证明器，介绍其逻辑框架  |
| (2)证明方法和语言 | 综合 | 通过引入主流定理证明器介绍基本证明方法和自动化证明方法 | 能利用定理证明器分析基本的证明方法 | 3 | 2 |
| (3)函数式数据结构、算法和证明 | 综合 | 介绍基本的函数式数据结构，分析其算法，利用主流定理证明工具完成若干种数据结构算法的构造和证明 | 能利用定理证明工具验证主流的数据结构算法及其正确性 | 3 | 2 |

**表5-2 实验/上机部分教学内容与进度要求**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **实验/上机项目** | **实验内容与方法** | **实验****类型** | **学时** | **必做/选做** | **课程目标** |
| 1 | 程序正确性证明 | 基于霍尔逻辑，对C程序代码建立前后断言，以确保程序的正确性 | 设计性 | 2 | 必做 | 1、4 |
| 2 | 模型检测工具的运用 | 利用模型检测主流工具，如UPPAAL，对实例进行建模、规约和自动验证 | 设计性 | 2 | 必做 | 2、3 |
| 3 | 基于定理证明器的程序设计 | 安装一种主流定理证明器，如Isabella，熟悉其程序语言，基于其基本类型和表达式构造简单程序 | 综合性 | 2 | 必做 | 2、3 |
| 4 | 函数式数据结构设计和证明 | 利用定理证明器设计指定函数式数据结构的算法，并对其正确性开展证明 | 综合性 | 2 | 必做 | 2、3 |
| 5 | 查找算法的设计和证明 | 利用定理证明器设计特定的查找算法，并对其正确性进行证明 | 综合性 | 2 | 必做 | 2、3 |

**表5-3 课程实施手段**

|  |  |  |
| --- | --- | --- |
| **序号** | **采用方法和手段** | **具体目标** |
| 1 | 讲授法：课堂教学采用多媒体课件、电子教案、传统教学方法和线上教学辅助结合 | 帮助学生掌握领域内常识，掌握学习路径 |
| 2 | 练习法：主要用在课堂作业和实验中，通过设置与教学进度适应的习题 | 了解各种场景下形式化工具的运用，熟悉典型的模型检测和定理证明形式化验证工具的使用 |
| 3 | 任务驱动法：主要用于实验和课后作业，通过布置探究性的学习任务 | 让学生自主分工、查阅资料、研究分析与制作汇报报告，培养学生提出问题、分析问题和解决问题的能力 |

**六、课程思政**

**表6-1 课程思政具体案例示例**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **所属章节/****案例名称** | **案例教学目标** | **思政元素** |
| 1 | 1.2形式化方法的理论基础 | 理解形式化方法的理论来源，了解主要的计算机科学家及其理论贡献 | **勤勉、钻研**：图灵奖是计算机领域的最高奖项，图灵奖得主中多达十几名所做贡献与形式化方法直接相关，通过前辈科学家的事迹鼓励学生热爱专业、勇于探索。 |
| 2 | 2.3可计算性和计算复杂度 | 理解可计算性问题，了解哥德尔、图灵、丘奇等计算机先驱所做的贡献 | **务实、乐观**：计算理论发展史上一个重要事件就是希尔伯特计划的破产，通俗说就是逻辑学家们建立完美数学系统是不可能实现的，这就是著名的哥德尔不完备定理，通过理论学习引导学生脚踏实地、乐观面对学习和生活中的各种问题。 |

**七、考核及成绩评定**

1、考核方法

本课程考核采用平时成绩+实验成绩+期末考试的综合考核方式，即：

**总成绩=平时成绩\*20%+实验成绩\*20%+期末考试成绩\*60%**

平时成绩分为4部分：作业（10%）、课堂表现（10%）。

各考核环节及权重如表7-1所示。

**表7-1 考核环节及权重表**

|  |  |  |
| --- | --- | --- |
|  **项目名称** **课程目标** | **考核环节及成绩占比（%）** | **课程目标权重****（%）** |
| **课堂****表现****（1）** | **课后****作业****（2）** | **实验****成绩****（3）** | **期末****考试****（4）** |
| **课程目标1** | - | 4 | 2 | 10 | 16 |
| **课程目标2** | - | 2 | 8 | 20 | 30 |
| **课程目标3** | - | 2 | 8 | 20 | 30 |
| **课程目标4** | 10 | 2 | 2 | 10 | 24 |
| **成绩合计（%）** | 10 | 10 | 20 | 60 | 100 |

2、考核内容及评价标准

① 平时成绩：占总成绩的20%，其中课堂表现占10%，课后作业占10%。课堂表现主要考查学生到课情况、课堂参与积极度等。课后作业不少于五次，主要以教师出课题引导学生进行探究性学习为主，学生通过文献调研、实验等方式完成，主要考查学生学习主动性、完成的准确性等。

② 实验成绩：占总成绩的20%。课程实验不少于五次，主要考查实验态度、实验过程的正确性、实验的完成度等。

③ 期末考试：占总成绩60%。一般以实践类大作业、调研报告或课程论文形式进行，由教师提供考查要求，限时学生完成相关任务并提交考查文档。

3、按照工程教育认证标准和学校人才培养要求，考核以学生能力是否有效达成为基准。为保障学生课程培养能力的达成，规定期末考试卷面成绩应高于一定分数。

4、考核周期为一个学年。为使评价结果尽快反馈给各个教学环节，促使各个教学环节尽快改进，保证教学效果的快速提升，课程考核成绩评价每学年进行1次。

5、考核依据《计算机与电气工程学院课程目标达成评价实施办法》文件进行。

**八、课程质量评价和持续改进**

课程结束后由课程责任人以定量和定性评价方法，针对具体课程目标形成文字或图表形式的报告，针对学生个体和整体的学习成果评价并对相关问题进行分析；课程目标达成与课程在培养学生解决复杂工程问题能力的具体环节任务的达成相关性分析；对以上各薄弱环节进行原因分析，提供持续改进建议，并由学院教学指导委员会进行审核。针对学生个体和整体的课程目标评价方法如下：

1、课程考核成绩算分评价法：课程目标达成度算分评价法是以参加课程学习的所有学生获得课程成绩为样本，对支撑毕业要求中各个指标点对应的课程目标进行达成情况评价，要由任课教师、课程负责人进行评价。课程目标达成情况评价值计算按下面公式进行：

课程目标达成评价值 $\sum\_{i=1}^{k}\frac{Di}{Zi}×Pi$

上式中k是该课程目标评价环节数，评价环节有平时过程考核、课后作业、大作业、实验、期末考试等（具体的依据考核审核表确定）; z𝑖 是第𝑖个评价方式的总分值，𝐷𝑖 是为学生在第𝑖个评价方式上的得分，𝑃𝑖 是第𝑖个评价方式在该课程目标评价中的占比。

2、针对学生课程目标未达成者，通过优秀学生与其沟通交流及任课教师进一步专题辅导改进；整体达成度较差部分，任课教师通过抽查与学生交流、分析问题，作进一步教学内容及方法的改善。

**九、教材与主要参考资料**

**1.教材：**Doron A.Peled. 软件可靠性方法[M]. 机械工业出版社. 2012.3

**2.教学参考书目：**

[1] Kenneth H.Rosen. 离散数学及其应用[M]. 机械工业出版社. 2020.1

[2] Michael Hush,Mark Ryan. 面向计算机科学的数理逻辑[M]. 机械工业出版社. 2020.9

[3] 张广泉. 形式化方法导论[M]. 清华大学出版社，2015.12

[4] 王戟,詹乃军,冯新宇,刘志明.形式化方法概貌[J].软件学报,2019,30(1):33-61

[5] Robert W.Sebesta. 程序设计语言原理[M]. 机械工业出版社. 2022.1

**十、教学团队**

|  |  |  |
| --- | --- | --- |
| **姓名** | **职称** | **承担的教学工作** |
| 李果 | 讲师 | 课程负责人、主讲教师 |
| 何青 | 讲师 | 主讲教师 |
| 巢湘萍 | 讲师 | 主讲教师 |
| 江伟 | 讲师 | 主讲教师 |

执笔人：李果

系（室）审核机构：软件工程教学大纲审核小组 组长：江伟

审核执行人（签字）胡千红

2022年12月8日

教学院审核机构：

计电学院教学大纲审核小组 组长李建英

审核执行人（签字）沙伯海

2023年2月27日