第 11 周：成果展示与总结

展示你的 AI 阅卷助手

北京石油化工学院\人工智能研究院\王文通

北京石油化工学院
BEIJING INSTITUTE OF PETROCHEMICAL TECHNOLOGY

北京石油化工学院
人工智能研究院

2025-2026 学年

课程概览

本周内容：

- 项目展示重要性
- 技术演讲技巧
- Demo 演示方法
- **AI 辅助演讲准备**
- 项目总结复盘
- 成果展示实战

学期项目：AI 阅卷助手

1. 图像采集与预处理
2. 答题卡定位（Timing Marks）
3. 填涂检测与识别
4. 手写文字 OCR
5. 成绩统计与输出

本周时间分配（100 分钟 = 2 学时）

第 1 学时（45 分钟）：

00:15 背景知识铺垫（15min）

00:30 核心理论讲解（15min）

00:45 工具环境介绍（15min）

第 2 学时（55 分钟）：

01:15 Live Coding 实战（30min）

01:30 案例分析与互动（15min）

01:40 展示规则说明（10min）

时间控制提示

理论部分简洁，重点放在实战演练

展示顺序

抽签决定，确保公平公正

本周预习资源

课前 5 分钟视频（必须观看）

- **视频 1**：演讲技巧基础（开场、结构、结尾）
- **视频 2**：Demo 演示最佳实践
- **视频 3**：答辩技巧与应对策略

预习目标：

- 理解优秀演讲的基本结构
- 掌握 Demo 演示的关键技巧
- 学会应对答辩中的常见问题

预习检查：

- ☐ 能说出优秀演讲的三段式结构
- ☐ 能设计一个吸引人的开场 Hook
- ☐ 能预测评委可能问的问题

为什么项目展示很重要?

展示能力是工程师的核心软技能

对个人发展:

- **求职**:面试中展示项目经验
- **晋升**:向领导展示工作成果
- **影响力**:在团队中建立声誉
- **学习**:通过准备深化理解

对项目价值:

- **获得反馈**:发现不足和改进点
- **推广价值**:让他人了解项目
- **积累声誉**:建立个人品牌
- **总结经验**:系统化梳理收获

展示能力的长期价值

- 技术能力决定下限,展示能力决定上限
- 好的技术需要好的展示才能发挥价值
- 展示过程本身就是学习和成长

展示对个人职业发展的影响

从学生到工程师的必经之路

职业发展各阶段的展示需求：

阶段	展示场景	目的
求职	面试作品展示	证明能力，获得 offer
初级工程师	项目汇报	展示工作，获得认可
中级工程师	技术分享	传播知识，建立影响力
高级工程师	技术演讲	行业影响力，个人品牌
技术专家	会议演讲	引领方向，建立声誉

展示能力的复利效应：

- 一次好的展示 = 长期的专业声誉
- 展示能力 = 被看见的能力 = 更多机会
- 越早练习，竞争优势越大

如何在展示中脱颖而出

从合格到优秀的展示技巧

合格展示 vs 优秀展示：

合格展示	优秀展示
罗列功能清单	讲述问题和解决方案的故事
平铺直叙地演示	设计有起伏的演示流程
技术细节堆砌	聚焦核心价值和创新
被动回答问题	主动引导思考
单纯完成展示	激发兴趣和讨论

脱颖而出的关键：

1. **故事化**：用故事串联技术内容
2. **价值导向**：始终聚焦"解决了什么问题"
3. **真诚**：诚实面对不足，展示真实过程

常见展示问题分析

避免这些常见错误

内容层面：

- × 内容过多，信息过载
- × 缺乏主线，逻辑混乱
- × 过度技术，听众跟不上
- × 没有故事，像念说明书

表达层面：

- × 声音太小，后排听不清
- × 语速过快，没有停顿
- × 照屏念稿，没有交流
- × 肢体僵硬，缺乏自然

Demo 层面：

- × 准备不足，现场翻车

技术成果展示的多种形式

不同场景选择不同的展示方式

Demo 演示

- 实时操作
- 直观效果
- 互动性强
- 风险较高

技术演讲

- 系统讲解
- 深度分析
- 理论结合
- 适合分享

海报展示

- 一目了然
- 互动交流
- 时间灵活
- 学术场合

视频展示

- 可控性强
- 精心剪辑
- 可重复播放

在线项目

- 可交互体验
- 随时访问
- 持续展示

选择合适的展示形式

根据场景和目的选择

场景	推荐形式	原因
课程项目展示	Demo + 演讲	展示功能 + 说明技术
求职面试	在线项目 + Demo	可体验 + 可验证
技术大会	演讲 + 视频	可控 + 传播性
学术会议	海报 + 论文	深度交流 + 学术规范
产品发布	视频 + Demo	精彩呈现 + 真实体验
团队分享	演讲 + 讨论	知识传递 + 互动

组合使用多种形式：

- 核心展示：Demo/演讲
- 辅助材料：视频/海报
- 后续访问：在线项目

Demo 演示的特点与技巧

最直观的展示方式

Demo 的优势：

- **真实性**：眼见为实，最有说服力
- **互动性**：可以根据观众问题调整
- **感染力**：现场演示更有冲击力

Demo 的挑战：

- **环境依赖**：硬件、网络、数据
- **不可控因素**：运行时的问题
- **时间压力**：实时操作不可重来

成功 Demo 的要素：

1. 充分的准备和测试
2. 设计好的演示脚本
3. 准备备用方案

技术演讲的特点与技巧

系统化的知识传递

技术演讲的结构：

- **问题引入**：为什么这个问题重要？
- **解决方案**：我们是如何解决的？
- **技术细节**：核心思想和实现
- **效果验证**：数据、对比、演示
- **总结展望**：收获和未来方向

技术演讲的平衡：

- 深度 vs 广度：深度讲清楚核心，广度留延伸
- 理论 vs 实践：理论支撑实践，实践验证理论
- 技术 vs 故事：技术是骨架，故事是血肉

技术演讲的最高境界

项目总结的意义

总结是学习闭环的关键环节

学习闭环：

$$\textbf{实践} \rightarrow \textbf{反思} \rightarrow \textbf{总结} \rightarrow \textbf{提升} \rightarrow \textbf{新实践}$$

不做总结的后果：

- 经验碎片化，无法复用
- 同样的错误重复犯
- 能力成长缓慢
- 项目价值流失

有效总结的价值：

- **系统化**：将零散经验体系化
- **可复用**：形成可迁移的方法论
- **可见性**：让成果和经验可见

如何进行有效的项目复盘

系统化的复盘方法

复盘四步法详解：

1. **目标回顾**
   - 当初的目标是什么？
   - 期望达成什么结果？
   - 目标是否合理？
2. **过程分析**
   - 实际发生了什么？
   - 哪些做得好？为什么？
   - 哪些不到位？为什么？
   - 有哪些意外？如何应对？
3. **结果评估**
   - 目标完成度如何？
   - 与预期的差距在哪？
   - 有哪些意外收获？

从失败中学习

失败是宝贵的经验来源

正确看待失败：

- **失败是常态**：开发中失败不可避免
- **失败有价值**：每次失败都是学习机会
- **分享失败**：让团队从你的失败中学习

从失败中学习的方法：

1. **承认失败**：不要回避或掩饰
2. **分析原因**：找出根本原因
3. **总结教训**：形成可复用的经验
4. **制定改进**：避免重复犯错

失败复盘模板

问题：什么失败了？ → **原因**：为什么会失败？ → **教训**：学到了什么？ → **改进**：下

持续改进的方法

让每个项目都成为成长的阶梯

持续改进的循环：

$$\textbf{Plan} \rightarrow \textbf{Do} \rightarrow \textbf{Check} \rightarrow \textbf{Act} \quad \text{(PDCA 循环)}$$

持续改进的实践：

- **记录技术债务**：
  - 已知问题列表
  - 待优化项
  - 待探索方向
- **建立最佳实践**：
  - 可复用的代码模板
  - 可遵循的工作流程
  - 可参考的设计模式
- **知识沉淀**：

项目总结的输出物

有形的总结成果

项目总结应该包含：

代码层面：

- 清理的代码仓库
- 完善的 README
- 详细的 API 文档
- 清晰的代码注释

文档层面：

- 项目总结报告
- 技术亮点说明
- 问题与解决方案
- 经验教训清单

展示层面：

- 演示幻灯片
- 演示视频
- 技术分享材料

好的总结

技术演讲的重要性

为什么展示能力很重要?

职业发展必备技能

- **求职面试**：向面试官展示你的能力
- **技术分享**：在团队中分享知识
- **产品发布**：向客户展示价值
- **项目汇报**：向管理层汇报进展

优秀技术演讲的价值:

- 展示专业能力
- 建立个人品牌
- 传播技术价值
- 获得反馈成长

技术演讲的标准结构

经典的三段式结构

开场	主体	结尾
Hook 听众	**传达内容**	**留下印象**
• 引人入胜	• 清晰逻辑	• 总结要点
• 明确主题	• 充实内容	• 呼吁行动
• 建立期待	• 生动演示	• 感谢听众

优秀演讲 = 好故事 + 清晰逻辑 + 生动的 Demo

开场技巧：抓住听众注意力

如何设计一个精彩的开场？

开场 Hook 的常见方式：

1. **提问式开场**
   - ” 你是否曾为期末阅卷的繁琐而烦恼？ ”
   - ” 想象一下，如果阅卷只需几秒钟...”
2. **数据式开场**
   - ” 一个老师平均每周需要花费 10 小时阅卷...”
   - ” 传统阅卷的误差率高达 5%...”
3. **故事式开场**
   - ” 上个月，我帮老师改了 500 份试卷...”
   - ” 我们的项目始于一个简单的想法...”
4. **演示式开场**
   - 先展示最终效果
   - 让结果说话

主体结构：清晰的逻辑组织

如何组织演讲的主体内容?

常见的逻辑结构:

问题-解决方案型

1. 提出问题
2. 分析原因
3. 展示方案
4. 验证效果

时间顺序型

1. 项目背景
2. 开发过程
3. 遇到挑战
4. 最终成果

内容组织原则:

- **金字塔原则**：结论先行，再展开细节
- **MECE 原则**：相互独立，完全穷尽
- **故事化表达**：用故事串联技术内容

结尾技巧：留下深刻印象

如何设计一个有力的结尾?

有效结尾的要素:

1. **总结要点**:
   - " 今天我们展示了..."
   - " 核心功能包括..."
2. **强化价值**:
   - " 这个系统能节省 80% 的阅卷时间..."
   - " 准确率媲美人工阅卷..."
3. **展望未来**:
   - " 未来我们计划支持主观题..."
   - " 这个技术可以应用到更多场景..."
4. **呼吁行动/致谢**:
   - " 欢迎大家试用我们的系统..."
   - " 感谢老师的指导，感谢团队的努力..."

幻灯片设计：视觉设计原则

优秀的幻灯片设计让演讲更有效

核心设计原则：

1. **简洁原则**：
   - 一页一个核心观点
   - 去除不必要的装饰
   - 留白让内容呼吸
2. **视觉原则**：
   - 用图表代替文字
   - 用截图代替描述
   - 用动画展示流程
3. **一致性原则**：
   - 统一的字体和字号
   - 统一的配色方案
   - 统一的排版风格
4. **层次原则**：

幻灯片设计：信息层次与排版

如何组织信息层次?

层次化的信息组织:

层级	字号	用途
一级标题	36-44pt	幻灯片标题，最醒目
二级标题	28-32pt	章节标题，次醒目
正文内容	18-24pt	核心内容，易读
辅助说明	14-16pt	注释、来源

排版技巧:

- **对齐**：左对齐为主，保持视觉流线
- **间距**：行距 1.2-1.5 倍，段落间距明显
- **分组**：相关内容靠近，用框架定边界
- **对比**：用大小、颜色突出重点

幻灯片设计：图表与可视化

让数据说话，让图表表达

常用图表类型：

展示比较：

- 柱状图：数量对比
- 条形图：排名对比
- 对比图：前后对比

展示趋势：

- 折线图：时间变化
- 面积图：累积变化

展示占比：

- 饼图：部分与整体
- 环形图：现代感的饼图

展示关系：

- 流程图：步骤关系
- 架构图：系统结构
- 思维导图：概念关系

技术演讲必备的可视化：

- 系统架构图
- 数据流程图

幻灯片设计：动画与过渡

恰当使用动画增强表达

动画使用原则：

1. **有目的性**：
   - 逐步展示内容，避免信息过载
   - 强调重点，吸引注意力
   - 展示流程，说明顺序
2. **克制使用**：
   - 不要为了动画而动画
   - 避免花哨的转场效果
   - 保持一致的动画风格
3. **适度简洁**：
   - 淡入淡出最安全
   - 从左/右飞入适合列表
   - 缩放适合强调重点

演讲表达：语言与声音

让你的表达更有感染力

语言表达技巧：

语速控制

- 正常：120-150 字/分钟
- 重点处：放慢
- 过渡处：加快
- 关键数字：重复强调

停顿艺术

- 章节之间停顿
- 重点之前停顿
- 提问之后停顿
- 让听众思考

声音控制要点：

- **音量**：足够大，后排能听清
- **语调**：有起伏，避免单调
- **清晰**：发音清楚，避免吞字
- **热情**：传递自信与能量

演讲表达：肢体语言

身体语言强化表达效果

站姿与移动：

- **站姿**：双脚与肩同宽，挺胸抬头
- **移动**：适度移动，不要晃动
- **面向**：面向观众，不要背对观众

手势运用：

手势类型	适用场景
开放式手势	欢迎观众、展示开放态度
指示性手势	引导观众看屏幕、强调重点
切分式手势	列举要点、区分不同内容
强调式手势	配合重点、加强语气

演讲表达：眼神交流

建立与观众的联系

眼神交流技巧：

1. **扫视全场**：
   - 不要盯着屏幕或地面
   - 定期扫视全场观众
   - 确保各个区域都被覆盖
2. **短暂对视**：
   - 与单个观众短暂眼神接触（3-5 秒）
   - 传递信心和真诚
   - 获得观众反馈
3. **关注反馈**：
   - 观察观众的反应
   - 看到困惑时放慢节奏
   - 看到兴趣时展开讲解

演讲表达：应对紧张

紧张是正常的，学会管理它

演讲前的准备：

- **充分准备**：准备越充分，越有信心
- **提前到场**：熟悉环境，测试设备
- **深呼吸**：上场前做几次深呼吸
- **积极想象**：想象成功演讲的场景

演讲中的应对：

- **承认紧张**：说出来反而会放松
- **放慢语速**：给自己思考的时间
- **记住内容**：关注内容而非自己
- **寻找支持**：从友善的观众获得鼓励

记住

QA 环节：回答问题的技巧

问答环节展示应变能力

回答问题的标准流程：

1. **倾听**：
   - 认真听完问题
   - 必要时重复确认
   - 不要急于打断
2. **思考**：
   - 花几秒组织答案
   - 确保理解问题核心
   - 决定回答的深度
3. **回答**：
   - 先给出直接答案
   - 再展开解释说明
   - 必要时举例说明
4. **确认**：

QA 环节：处理不同类型的问题

针对不同问题采用不同策略

问题类型	应对策略	示例
技术细节	解释思路，不必展开代码	"我们采用 X 方法，核心是..."
质疑效果	用数据说话，诚实对待	"准确率 95%，基于 100 张测试..."
超出范围	说明限制，聚焦核心	"这超出我们范围，但..."
多个问题	分别回答，控制时间	"关于 A... 关于 B..."
不太确定	诚实承认，提供思路	"我不太确定，但可能是..."

万能回答框架：

- "这是个好问题，我们确实考虑过..."
- "这个问题的核心在于..."
- "我们从 XX 角度来解决..."

QA 环节：处理刁钻问题

面对挑战性问题保持专业

常见刁钻问题与应对：

"你们的项目有什么创新？"

- 避免夸大
- 说明改进点
- 强调应用场景

"准确率这么低？"

- 说明测试条件
- 解释误差来源
- 说明改进方向

"这个技术很简单啊"

- 承认技术基础
- 强调工程实践
- 说明集成价值

"这个不是你们写的吧"

- 诚实说明引用
- 强调集成工作
- 说明改进部分

QA 环节：不知道答案怎么办

诚实是最专业的选择

不知道答案时的应对策略：

1. **诚实承认**：
   - ”这个问题很有意思，但我目前没有相关数据”
   - ”我不太确定，需要进一步研究”
2. **提供思路**：
   - ”从 XX 角度看，可能是...”
   - ”类似情况下，通常...”
3. **转移焦点**：
   - ”我们主要关注的是...”
   - ”在我们的应用场景下...”
4. **承诺跟进**：
   - ”我可以会后研究一下，再和您交流”
   - ”这是个很好的方向，我们可以后续探讨”

QA 环节：时间管理

在有限时间内高效回答

时间控制技巧：

- **快速判断**：评估问题重要性，决定回答深度
- **简洁回答**：先给核心答案，再根据时间展开
- **合并回答**：多个相似问题合并回答
- **延后讨论**：复杂问题建议会后讨论

结束 QA 的时机：

- 预定时间快到时
- 回答了 3-5 个主要问题后
- 问题开始重复或跑题时

优雅结束 QA：

- ”由于时间关系，我们再回答最后一个问题”
- ”感谢大家的提问！感兴趣的同学可以会后继续交流”

Demo 准备：演示流程设计

精心设计的演示流程让 Demo 更流畅

Demo 流程设计原则：

1. **完整但不冗长**：
   - 展示完整的用户流程
   - 跳过重复性操作
   - 聚焦核心功能
2. **有故事线**：
   - 从问题出发
   - 展示解决方案
   - 证明有效可行
3. **突出亮点**：
   - 提前规划展示哪些功能
   - 准备对比场景
   - 设计”wow 时刻”
4. **有备选方案**：

Demo 准备：场景与数据准备

合适的测试数据让演示更有说服力

测试数据准备：

正常场景	边缘场景
• 标准答题卡	• 轻微倾斜
• 清晰填涂	• 光照不均
• 典型使用场景	• 部分填涂

数据准备检查清单：

- ☐ 准备至少 5-10 组测试数据
- ☐ 覆盖不同场景
- ☐ 提前验证能正确处理
- ☐ 准备"对比"数据（前后对比）
- ☐ 准备"失败"案例展示处理能力

Demo 准备：备用方案（Plan B）

做好最坏打算，确保演示顺利进行

常见故障与备用方案：

可能的故障	备用方案
代码运行报错	切换到备份代码分支
网络连接失败	使用本地数据/离线模式
处理速度太慢	使用预处理好的数据
识别效果不佳	选择最理想的测试数据
演示环境异常	切换到录制的演示视频
投影设备问题	使用截图 PPT 继续讲解

备用方案准备清单：

- □ 录制完整的演示视频
- □ 准备截图 PPT
- □ 准备备用测试数据

Demo 准备：环境检查清单

提前检查，避免现场出问题

演示前 30 分钟检查清单：

硬件检查：

- ☐ 笔记本电量
- ☐ 电源适配器
- ☐ 投影连接测试
- ☐ 音频输出测试
- ☐ 鼠标/翻页器

软件检查：

- ☐ 代码已拉取最新
- ☐ 虚拟环境激活
- ☐ 依赖包安装完整
- ☐ 测试数据就位
- ☐ 演示脚本准备

网络检查：

- ☐ WiFi 连接状态
- ☐ API 服务可用性
- ☐ 准备离线模式

Demo 执行：开场介绍

好的开场让 Demo 更有意义

Demo 开场三要素：

1. **场景介绍**：
   - ” 我们将演示如何处理这张答题卡...”
   - ” 这是一个典型的 XX 场景...”
2. **预期结果**：
   - ” 系统将自动识别出答案...”
   - ” 这个过程大约需要 X 秒...”
3. **流程预告**：
   - ” 演示将分为三个步骤...”
   - ” 首先我们做 X，然后 Y，最后 Z...”

Demo 开场示例

” 现在我来演示系统的核心功能。这是一张数学答题卡，包含 20 道选择题。我们的系统将

Demo 执行：功能展示顺序

合理的展示顺序让演示更流畅

推荐的 Demo 展示顺序：

1. **端到端演示**（30 秒）：
   - 先展示完整流程
   - 让观众看到最终效果
   - 建立整体印象
2. **分步骤演示**（2 分钟）：
   - 慢速展示关键步骤
   - 解释每个环节的作用
   - 展示中间结果
3. **亮点展示**（30 秒）：
   - 展示特殊功能
   - 对比展示效果
   - 强调创新点
4. **结果展示**（30 秒）：

Demo 执行：实时问题处理

当 Demo 出问题时保持专业

技术故障应对原则：

保持冷静

- 不要慌张
- 不要反复尝试
- 保持微笑

快速切换

- 立即启动 Plan B
- 平滑过渡
- 不要浪费时间

应对话术示例：

- ”看来我们的 AI 今天有点害羞，让我用录制好的视频来展示...”
- ”由于现场网络限制，我切换到本地模式继续...”
- ”这个小问题正好展示了我们的错误处理机制...”

Demo 执行：讲解与同步

边演示边讲解，让观众跟上思路

讲解同步技巧：

- **预告下一步**：
  - ” 接下来系统将进行...”
  - ” 大家注意看这里的变化...”
- **实时解释**：
  - ” 现在系统正在检测边缘...”
  - ” 这里可以看到定位结果...”
- **突出重点**：
  - ” 这个效果就是我们的创新点...”
  - ” 注意这里处理速度的提升...”
- **控制节奏**：
  - 关键步骤放慢
  - 简单操作快速带过
  - 等待观众理解

Demo 执行：总结收尾

有力的结尾让 Demo 更完整

Demo 总结要素：

1. **回顾演示内容**：
   - ” 刚才我们演示了...”
   - ” 展示了 X、Y、Z 三个功能...”
2. **强调价值**：
   - ” 整个过程只需 3 秒”
   - ” 准确率达到 95% 以上”
   - ” 相比人工节省 80% 时间”
3. **开放讨论**：
   - ” 大家对哪个功能最感兴趣？ ”
   - ” 有什么问题可以讨论”

完美 Demo 的标志

演讲与展示工具概览

选择合适的工具提升展示效果

工具类型	推荐工具	适用场景
幻灯片制作	PowerPoint/Keynote	通用场景，易上手
在线协作	Google Slides	团队协作，云端访问
创意展示	Prezi	非线性叙事，创意展示
设计美化	Canva	快速美化，模板丰富
学术演示	LaTeX Beamer	学术场合，公式多
代码演示	Reveal.js	技术分享，代码多

工具选择建议：

- 熟悉的工具 > 新潮的工具
- 简单的工具 > 复杂的工具
- 内容 > 形式：工具是手段，内容是核心

快捷键：

- Ctrl+D：快速复制
- F5：从头放映
- Shift+F5：从当前页放映

模板库：善用专业模板

设计原则：

- 一页一个观点
- 大字体（24pt+）
- 高质量图片
- 留白很重要

LaTeX Beamer 简介

学术和技术演讲的专业选择

Beamer 的优势：

- **专业排版**：学术级别的排版质量
- **公式支持**：完美的数学公式
- **代码高亮**：专业的代码展示
- **版本控制**：纯文本，便于版本管理
- **结构化**：强制结构化思维

Beamer 的使用场景：

- 学术报告
- 技术演讲

Demo 演示工具概览

选择合适的工具让演示更顺畅

工具类型	推荐工具	特点
屏幕录制	OBS Studio	免费、开源、功能强大
简单录制	ScreenFlow/Camtasia	易用、编辑功能强
快速分享	Loom	录制 + 分享一体化
在线演示	Zoom/Teams	远程演示、录制
GIF 制作	GifCam/Kap	轻量级动画展示
截图工具	Snipaste	截图 + 标注

工具选择建议：

- 准备用：录制完整演示视频
- 实时用：现场演示（Plan A）

OBS Studio 快速入门

最流行的开源录屏工具

OBS 的特点：

- 免费开源
- 功能强大（录制 + 直播）
- 跨平台（Windows/Mac/Linux）
- 支持多源混流

基本设置：

1. 添加源：屏幕捕获、窗口捕获
2. 设置输出：视频格式、质量
3. 开始录制：点击开始录制

文档与协作工具

高效的工具提升团队协作效率

知识管理工具：

- **Notion**：一体化工作空间，支持文档、数据库、看板
- **Confluence**：企业级知识管理，与 Jira 集成
- **语雀/飞书文档**：国内协作工具，适合中文环境

文档编写工具：

- **Markdown + Typora**：轻量级，专注写作
- **VS Code**：代码 + 文档一体化
- **Sphinx**：自动生成 API 文档

代码托管工具：