中国软件行业协会

关于“AI Agent与端侧AI深度实践：从理论架构到工程落地的闭环实战”高级研修班的通知

各有关单位：

当前，人工智能正步入以“大模型+智能体”为核心、端云协同发展的新阶段。AI Agent通过自主规划、工具调用与多步推理能力，正深刻重塑人机交互与业务流程自动化范式；而端侧AI凭借低延迟、高隐私、强实时的优势，成为移动、物联网、边缘计算等领域智能化落地的关键路径。为推动AI Agent与端侧AI技术的深度融合与产业应用，助力各单位系统构建从智能体架构到移动端部署的全链路工程能力，我协会定于2026年一季度举办“AI Agent与端侧AI系统实战：从架构设计到移动端落地”高级研修班。

现将有关事项通知如下：

本次研修围绕“AI Agent工业级架构”与“Android端侧AI工程实践”两大主线，系统覆盖ReAct框架、Agentic RAG、多Agent协作、Memory系统、SLA保障等核心模块，以及NNAPI、JNI集成、模型量化、实时推理等端侧关键技术。课程采用“架构解析+代码实战+行业案例”相结合的方式，帮助学员打通从智能体设计到移动端落地的完整技术闭环，构建高可用、低延迟、可扩展的AI系统能力。

研修班特邀华为昇腾生态资深专家、HanLP 开源技术部总经理崔老师主讲。崔老师长期致力于 AI Agent 工程化与端侧高性能推理技术的推广与落地，具备深厚的架构设计与产业实践背景，将结合金融、医疗、移动终端等行业实例，进行系统化讲授与实战指导。

通过本次研修，学员将系统掌握 AI Agent 与端侧 AI 的全景技术体系，获得从架构设计、工程实现、性能优化到行业解决方案的闭环实战能力，为各单位推进智能化升级、打造自主可控的 AI 产品提供关键技术支撑。

敬请各相关单位积极参加！

联系电话：010-85913702 郭老师

联系电话：010-62118502 张老师

联系邮箱：csia_org@yeah.net

“附件：“AI Agent 与端侧 AI 深度实践：从理论架构到工程落地的闭环实战”高级研修班简章

中国软件行业协会

2026 年 1 月 8 日

《AI Agent 与端侧 AI 深度实践：从理论架构到工程落地的闭环实战》高级研修班简章

一、时间和方式

时间：2026 年 2 月 7 日至 2 月 8 日（共 2 天）

方式：线上直播（腾讯会议）

二、研修对象

面向广泛致力于 AI 技术融合、产业应用与前沿研究的各领域专业人士，包括但不限于：负责技术战略与架构的 CTO、技术总监、系统架构师；专注算法研发与工程落地的 AI 工程师、机器学习专家与大模型开发人员；深耕移动端、物联网及边缘计算的开发与系统工程师；主导智能产品规划与行业解决方案设计的产品经理与解决方案专家；来自通信、能源、制造等领域的运营商与大型企业技术团队；关注技术趋势与数字化治理的政府相关部门、智库与研究机构人员；以及从事人工智能教学、科研与人才培养的高校教师、研究员与博士后。课程同时适合推动企业智能化转型的业务决策者、创新实验室负责人，以及希望系统构建 AI 工程能力的技术管理者与高级研发人员等。

三、研修大纲

模块一：AI Agent 工业级架构：ReAct 框架在关键行业的实现

模块二：Agentic RAG 技术体系：从检索增强到主动知识服务

模块三：Memory 系统工程：层次化架构与多模态管理实践

模块四：多 Agent 协作系统：任务分解机制与容错通信实现

模块五：Agent 性能工程：SLA 保障与可靠性评估

模块六：行业级 Agent 应用：智能服务系统工程实践与验证

模块七：Android 端侧 AI 基础：NNAPI 技术栈与硬件加速

模块八：JNI 集成工程：跨语言调用与 SO 库高效实现

模块九：端侧模型工程：INT8 量化与设备适配优化策略

模块十：端侧 AI 应用工程：实时推理系统构建与性能调优

四、研修详细内容

▶ 模块一：AI Agent 工业级架构：ReAct 框架在关键行业的实现

1. 行业痛点与核心价值

传统 AI 的局限性：静态响应、缺乏复杂推理与多步骤规划能力。

Agent 技术的颠覆性价值：实现自主决策、任务规划与外部工具调用，从“应答”走向“执行”。

企业应用场景分析：在金融、医疗、客服等关键行业的数字化转型中，Agent 如何驱动复杂业务流程自动化（如自动化贷款审批需联动征信、收入验证与风控模型）。

2. 定义与框架结构

AI Agent 的核心定义：感知-规划-行动（Perception-Planning-Action）的自主闭环智能实体。

核心推理框架：

- **ReAct（Reasoning + Acting）**：Thought-Action-Observation 双循环机制，是支持工具调用的首选范式。
- **Chain-of-Thought（CoT）**：思维链推理，提升复杂问题分步求解能力。
- **Tree-of-Thoughts（ToT）**：思维树框架，支持多路径探索与回溯，适用于创意生成类任务。

架构设计选型：

- 单 Agent 系统：适用于功能聚焦、工具链较少（如≤5 个）的垂直场景，如智能客服。
- 多 Agent 系统：用于需角色分工与复杂协同的综合性任务，如自动化研发团队。

工具集成与 API 调用机制：采用标准化接口（REST/gRPC）与 SDK，设计需内置权限控制、流量限制与完整调用审计日志。

3. 技术实现路径

基于 LLM 的 Agent 构建策略：

- **Prompt Engineering**：通过结构化提示词（如 ReAct 格式）快速构建与验证智能体原型。
- **Fine-tuning**：使用领域任务数据对底座模型进行微调，以提升指令遵循与工具调用的准确性与稳定性。
- **函数调用（Function Calling）与工具使用**：使用严格定义的 JSON Schema 描述工具，并在调用前后进行参数校验、类型转换与智能缺省值填充。

状态管理与上下文维护策略：

- 采用分级策略管理对话上下文，如前 5 轮保留完整上下文，5－10 轮使用摘要与关键节点，10 轮以上仅保留核心决策链。
- 结合向量数据库实现长期记忆。

错误处理与异常恢复机制：

- 实施具备指数退避（如 $t_n=\min(t_{max}, t_0 \cdot 2^n+\text{Jitter})$ ）的超时重试。

- 服务降级（如 fallback 到规则引擎）及人工接管通道，保障生产环境鲁棒性。

▶ 模块二：Agentic RAG 技术体系：从检索增强到主动知识服务

核心内容：

1. 传统 RAG 的演进

Naive RAG 的瓶颈：被动触发、检索结果信息碎片化、缺乏对检索过程的迭代优化能力。

Agentic RAG 的范式跃迁：将检索决策权赋予智能体，使其在推理过程中自主判断“是否需要检索”及“检索什么”。

核心优势：实现“思考驱动检索”，显著提升知识获取的精准度与答案的上下文相关性。

2. 技术架构与实现

检索智能体（Retrieval Agent）设计：

- 构建具备查询理解（Query Understanding）能力的智能体，集成同义词扩展、专业术语归一化与用户意图识别模块，将检索准确率提升至 >92%。

多步推理与迭代检索机制：

- 设计反馈循环，当本次生成结果与前次结果的语义相似度（如 $sim(Ok, Ok-1)>0.85$ ）过高时，自动触发二次或定向检索。

知识融合应用策略：

- 融合向量数据库（如 FAISS，用于相似性检索）与知识图谱（如 Neo4j，用于关联推理），实现“语义匹配+逻辑关联”的联合检索。

动态知识更新与版本管理：

- 结合事件驱动架构，监听知识源变更，实现知识片的实时或准实时索引更新，并支持多版本知识库的查询与回溯。

3. 高级 RAG 策略

Self-RAG：引入反思标记，训练模型对自身生成过程进行批判性评估，主动判断段落的相关性、输出的事实支持度，实现检索质量的自我优化。

Adaptive-RAG：根据任务复杂度或模型对当前知识的置信度阈值，动态切换检索模式（如密集型检索、稀疏型检索或跳过检索）。

Multi-hop RAG：支持跨越多个文档进行多跳推理，通过迭代式检索与信息整合，解决需要间接关联和复杂逻辑链的问题。

GraphRAG：利用图神经网络（GNN）对知识图谱进行表示学习，挖掘实体间的深层、隐含路径，实现基于图结构的复杂推理与增强。

▶ 模块三：Memory 系统工程：层次化架构与多模态管理实践

核心内容：

1. Memory 系统设计原理

记忆分层架构：

- **短期记忆**：维持当前会话的完整上下文，保障对话连贯性。
- **长期记忆**：支持跨会话、跨周期的用户知识与偏好复用。

记忆存储形态融合：

- **向量记忆**：基于嵌入（Embedding）的语义记忆，用于相似性匹配。
- **符号记忆**：存储结构化的事实、规则与用户画像。
- **参数记忆**：通过模型微调（如 LoRA）将关键知识或偏好固化至模型参数中。

2. 实现技术与优化

基于 Embedding 的语义记忆系统：

- 采用 Sentence-BERT 等模型生成高质量、固定维度的语义表示，构建高效的向量检索库。

层次化架构工程实现：

- **工作记忆**：采用 LRU 缓存，存储最近 N 轮（如 5 轮）高细粒度对话。
- **情景记忆**：使用文档数据库（如 MongoDB），按 session_id 索引存储完整的历史交互序列。
- **语义记忆**：利用图数据库（如 Neo4j）构建动态演化的用户兴趣图谱与领域知识网络。

记忆压缩与遗忘策略：

- 实施基于聚类的记忆条目合并，并应用时间衰减函数（如 $wt=e-\lambda\cdot\Delta t, \lambda=0.01$ ）自动降低旧记忆的检索权重。

多模态记忆统一管理：

- 通过图像描述生成（Image Captioning）、语音转文本（ASR）等技术，将非文本信息映射至统一的语义嵌入空间进行处理与检索。

3. 企业级 Memory 解决方案

对话历史智能管理：支持基于关键信息提取的对话摘要生成与快速上下文追溯。

用户画像与个性化记忆：实现用户偏好、行为模式的持续学习与动态更新机制。

知识库的持续学习与更新：设计增量学习流程，支持在线吸收新知识，并与已有知识进行冲突检测与融合。

安全与隐私保护：遵循 GDPR 等数据合规要求，实现记忆数据的加密存储、访问控制与用户数据删除权（Right to be Forgotten）的技术支持。

▶ 模块四：多 Agent 协作系统：任务分解机制与容错通信实现

1. 多 Agent 系统架构

组织模式对比：

- **中心化架构（Hub-Spoke）**：由协调者（Coordinator）统一进行任务分配、调度与结果汇总，结构简单，易于控制。
- **去中心化架构（P2P）**：智能体间通过点对点通信与共识机制（如 Gossip 协议）进行协作，无单点故障，扩展性强。

角色定义与分工：

- 明确划分智能体角色，如**规划者（Planner）**负责任务分解，**执行者（Executor）**负责调用工具，**验证者（Validator）**负责结果质检。

通信协议与消息机制：

- 根据场景选择同步请求-响应（低延迟）或异步消息队列（高吞吐、解耦）。
- 并定义统一的消息格式与序列化协议。

2. 协作策略与优化

复杂任务分解方法：

- 采用层次任务网络（HTN）等算法递归拆解高层目标，确保子任务成本总和 $\sum c(Ti)<c(T)$ 。

基于依赖图的并行执行优化：

- 构建任务依赖图（DAG），通过拓扑排序确定执行顺序。
- 并结合动态优先级（$P=w1\cdot$紧急性$+w2\cdot$业务价值 ）进行调度。

资源共享与负载均衡：

- 建立计算、数据等资源的虚拟化池。
- 并实施基于智能体负载状态的动态任务分配，防止局部过载。

协商与冲突解决机制：

- 设计基于规则的协商或简单的拍卖机制，解决智能体间的目标冲突与资源竞争问题。

3. 容错机制实现

故障检测：实现基于心跳（Heartbeat）与进程健康检查的监控系统，快速发现故障节点。

消息可靠性保障：

- 采用“至少送达一次（At-least-once Delivery）”语义，配合消息去重与幂等处理器，确保通信可靠。

状态恢复与系统自愈：

- 设计检查点（Checkpoint）机制保存关键状态。
- 支持故障转移（Failover）与任务重新派发，实现系统级自愈，服务窗口期（SWT）目标 < 500ms。

▶ 模块五：Agent 性能工程：SLA 保障与可靠性评估

1. 性能瓶颈识别

核心耗时环节分析：

- 剖析全链路，识别主要瓶颈：LLM 生成推理（占 40 - 60%）、外部工具调用的网络 I/O、记忆系统的检索与加载开销。

全链路追踪与剖析方法：

- 集成 OpenTelemetry 等标准，实现从用户请求到最终响应的全链路 Trace 追踪，精准定位性能热点。

2. 优化策略实施

上下文管理优化：

- 应用滑动窗口技术限制输入长度。
- 结合文本摘要（Summarization）对历史对话进行压缩，平衡信息完整性与推理效率。

缓存机制：

- 对高频且稳定的检索结果、工具调用结果实施缓存，设置合理的 TTL（如 5 分钟），显著降低重复计算与检索延迟。

并行处理：

- 对相互独立的子任务或可并行调用的外部工具，采用异步并发执行，最大化利用系统资源。

3. 评估体系构建

关键 SLA 指标定义：

- **端到端延迟**：P95 < 2 秒（行业常见水平为 3 - 5 秒）。
- **任务成功率**：> 90%（涵盖理解、规划、执行全流程）。
- **工具调用准确率**：> 95%（参数正确、结果有效）。

可观测性三支柱建设：

- **日志（Logging）**：结构化的操作与审计日志。
- **指标（Metrics）**：面向 SLA 的核心性能与业务指标仪表盘。
- **追踪（Tracing）**：请求粒度的分布式调用链追踪。

可靠性测试实践：

- 引入混沌工程（Chaos Engineering），在生产仿真环境中注入网络延迟、服务中断等故障，验证系统的容错与自愈能力。

▶ 模块六：行业级 Agent 应用：智能服务系统工程实践与验证

核心内容：

1. 行业解决方案剖析

智能客服 Agent：

- 集成对话状态跟踪（DST，如 BERT-DST 模型，准确率 > 92%）与业务逻辑，实现从多轮问询到业务系统（如 CRM）调用的端到端自动化办理。

数据分析 Agent：

- 将自然语言查询自动转换为可执行的 SQL 或 Python 脚本（结合 AST 语法树校验），并自动执行分析与可视化，标注关键异常点（如 Z-score > 3）。

代码开发 Agent：

- 根据产品需求文档（PRD）自动生成代码骨架、单元测试，并与代码质量平台（如 SonarQube）集成进行静态检查。

文档处理 Agent：

- 实现长文档的智能摘要、关键信息（如合同条款、金额）的精准提取与基于知识的问答。

2. 系统工程实践

MLOps 流程：

- 对智能体的核心资产——提示词（Prompt）、工具定义（Tool Schema）、记忆结构（Memory Schema）——进行版本化、协同化的配置管理。

安全审计：

- 部署策略检查器（Policy Checker），对所有外部动作（Action）调用进行前置授权与后置审计，防止越权操作。

部署规范：

- 在企业私有 VPC 内进行容器化部署，对敏感操作实施双因素认证（2FA）与操作录像。

3. 验证与上线策略

评估方法：

- 使用标注好的真实业务工单数据集，从意图识别准确率、任务完成率、用户满意度等多维度进行 F1-score 等量化评估。

部署策略：

- 采用灰度发布（Canary Release），逐步将流量切至新版本智能体，严格控制风险暴露面。

根因分析：

- 建立生产环境故障案例库，对失败案例进行技术根因（如工具异常、上下文误解）与流程根因的深度分析，驱动持续优化。

▶ 模块七：Android 端侧 AI 基础：NNAPI 技术栈与硬件加速

核心内容：

1. 行业痛点与核心价值

云端 AI 的痛点：网络往返延迟（RTT）高（常 ≥ 100ms）、流量成本昂贵、用户隐私数据存在泄露风险。

端侧 AI 的核心优势：

- 实现毫秒级实时响应（目标 < 50ms）。
- 支持完全离线可用。
- 保障用户数据在本地处理，隐私安全可控。

典型应用场景：图像识别（如实时滤镜）、语音处理（如离线唤醒词）、文本分类（如敏感内容过滤）等对延迟与隐私要求高的场景。

2. Android AI 技术栈详解

NNAPI（神经网络 API）：

- Android 系统级 AI 接口（API Level ≥ 27），作为硬件抽象层，统一调用不同芯片厂商（如高通、联发科、华为）的 NPU/GPU/DSP 加速单元。

TFLite（TensorFlow Lite）：

- Google 主推的轻量级推理框架，专为移动和嵌入式设备优化，提供模型转换、量化及委托（Delegate）至 NNAPI 的完整工具链。

硬件加速机制：

- **GPU 加速**：通过 OpenGL ES Compute Shaders 或 Vulkan 进行并行计算。
- **NPU 加速**：调用专用 AI 处理核心（如华为达芬奇架构），能效比极高。

3. 性能对比与选型

典型性能数据：以 MobileNetV2 图像分类为例，

- CPU：89ms
- GPU：32ms
- NPU：18ms

→ 展示硬件加速的显著收益（3-5 倍提升）。

模型选型建议：优先选择针对移动端优化的轻量级架构（如 MobileNet、EfficientNet-Lite 系列），并在精度与速度间取得平衡。

▶ 模块八：JNI 集成工程：跨语言调用与 SO 库高效实现

核心内容：

1. JNI 开发基础

JNI 原理：Java 虚拟机（JVM）与本地（Native）代码（C/C++）之间的通信桥梁，涉及复杂的 JNI 环境、内存模型与引用管理。

标准开发流程：

- Java 层声明 Native 方法 → C/C++层实现对应函数 → 编译生成动态链接库(SO 文件) → Android 应用集成加载。

数据类型转换：掌握 `jint`，`jstring`，`jbyteArray` 等 JNI 类型与原生 C/C++类型（`int32_t`，`char`，`uint8_t`）之间的映射与转换规则。

2. SO 库集成 AI 模型

C++推理引擎选择：使用 ONNX Runtime C++ API 或 TFLite C++ API 作为核心推理引擎，实现模型加载与执行。

模型加载与内存管理：

- 正确地从 Assets 或文件系统读取模型文件。
- 管理模型生命周期的内存分配与释放，避免内存泄漏。

完整推理流程实现：

- 封装输入预处理（图像归一化、尺寸调整）。
- 模型推理（Session->Run()）。
- 输出后处理（Softmax、NMS 非极大值抑制）的完整 Native 函数。

错误处理：在 Native 层捕获异常，并通过 JNI 机制抛回 Java 层，同时记录详细的本地日志便于调试。

3. 性能优化关键技术

零拷贝传递：使用 DirectByteBuffer 或直接操作 Java 数组的底层内存地址，避免 Java 与 Native 层之间大规模数据的冗余拷贝。

异步调用：在独立 Native 线程池中执行耗时推理任务，防止阻塞 Android 主线程(UI 线程)，保障应用流畅性。

JNI 引用生命周期管理：正确管理局部引用与全局引用，及时释放不再使用的引用，防止局部引用表溢出。

▶ 模块九：端侧模型工程：INT8 量化与设备适配优化策略

核心内容：

1. 模型准备与转换

格式转换标准路径：

- PyTorch/TensorFlow → ONNX（中间格式） → TFLite（端侧格式）。
- 确保算子兼容性。

INT8 量化技术：

- 详解后训练量化（PTQ）原理：
  $W_{int8}=round(scale_W fp32+zero\_point)$
- 使用代表性校准数据集确定动态范围。
- 将模型权值与激活从 FP32 降至 INT8，模型体积减少 75%，推理速度提升 2-4 倍，精度损失控制在<1%。

模型剪枝：

- 实施幅度剪枝（Magnitude Pruning）移除冗余权重连接，或结构化剪枝以保持硬件友好性，进一步压缩模型。

2. 性能优化策略

多线程推理：在支持多核的 CPU 上，利用线程池并行处理输入批次（Batch）或网络层，提升吞吐量。

内存池技术：预先分配并复用输入输出 Tensor 的内存空间，避免推理循环中频繁的内存分配与垃圾回收开销。

模型缓存：将初始化后的 TFLite 解释器（Interpreter）常驻内存，避免每次推理都重新加载模型，减少首次推理延迟。

3. 设备适配

ABI 差异化适配：在 Gradle 中配置 abiFilters（如‘armeabi-v7a’，‘arm64-v8a’），为不同 CPU 架构生成对应的 SO 库，控制 APK 体积。

模型热更新：设计从云端下载新版模型文件并安全替换的机制，支持模型迭代与 A/B 测试，无需重新发布应用。

▶ 模块十：端侧 AI 应用工程：实时推理系统构建与性能调优

核心内容：

1. AI 相机应用实战

功能目标：实现实时（≥15 FPS）图像分类或目标检测，完全离线运行，端到端延迟 <100ms。

系统架构：采用分层设计：

[CameraX 图像采集] → [GPU 预处理] → [JNI 桥接] → [SO 库推理引擎] → [UI 结果渲

染]。

完整执行流程：

- CameraX 获取预览帧。
- 在 RenderScript/Vulkan 上进行 YUV 转 RGB、Resize。
- JNI 调用 Native 推理函数。
- 获取分类结果/检测框。
- 回传至 UI 线程绘制。

2. 开发与部署要点

SO 库打包配置：在 Gradle 中精确定义 ndk 配置，控制输出的 ABI 类型，平衡兼容性与包大小。

模型热更新策略：实现安全的模型文件校验、下载与本地存储管理逻辑。

性能监控体系：

- 在关键路径插入耗时统计代码（埋点）。
- 实时监控并上报每帧推理耗时与应用帧率（FPS）。
- 设定性能基线（如 FPS < 15 触发告警）。

3. 未来展望

-端侧 AI 技术发展方向预测

-新兴应用场景探索

-开发者生态建设机会

-技术挑战与解决方案展望

五、研修收益

1. **构建全景知识体系，贯通 AI Agent 与端侧 AI 技术栈**
   系统掌握从 Agent 架构设计、RAG 增强、多 Agent 协作，到端侧模型优化、硬件加速及实时推理的全链路知识框架，理解技术融合发展趋势与落地逻辑。

2. **掌握工业级实现能力，从理论走向工程落地**
   通过模块化实战案例与代码级解析，深入理解 ReAct、Agentic RAG、Memory 系统、多 Agent 通信等核心技术的企业级实现路径，具备从 0 到 1 构建高可用智能系统的能力。

3. **获得行业解决方案设计与验证经验**
   深度剖析金融、医疗、客服、移动端等典型场景的 AI 应用案例，学会设计符合业务需求、兼顾性能、安全与可扩展性的可落地解决方案，并掌握其评估与迭代方法。

4. **掌握性能与可靠性工程方法，保障系统高可用**
   学习 Agent 与端侧 AI 系统的 SLA 设计、性能剖析、容错机制、可观测性建设等关键工程技能，具备构建低延迟、高稳定、易维护的 AI 系统的能力。

5. **获取权威认证，助力职业发展与项目竞争力**
   通过考核可获得由中国软件行业协会及工信系统颁发的高级职业技术证书，提升个人专业资质，为晋升、评职、项目投标提供有力支持。

6. **加入技术社群，获得持续学习与资源连接**

进入专属学习交流群，与行业专家、技术同行深度互动，获取最新工具、源码、案例与学习资料，构建可持续成长的技术生态网络。

六、专家介绍

崔老师(华为)
华为昇腾资深专家 ｜ HanLP 开源生态技术部总经理 ｜ AI Agent 技术总负责人，华为技术布道师。 崔老师人工智能工程化与产业落地领域的权威专家，长期致力于推动大模型与 Agent 系统的工业化部署，并在移动端与边缘侧的高性能 AI 推理方向取得了行业公认的突破性成果。作为华为昇腾生态中关键 AI 技术栈的规划与赋能者，其主导的端侧 AI 优化方案已成为多家头部厂商的落地标准；在执掌 HanLP 技术部期间，他构建了国内最具影响力的自然语言处理开源技术生态之一，相关工具与框架被广泛应用于金融、科技及互联网企业的核心生产系统。他亦多次在国际顶级技术峰会及行业论坛发表主旨演讲，其关于 Agent 架构与端侧 AI 融合的前瞻观点，深刻影响了该领域的技术演进。在专业层面，崔老师精通基于 ReAct 与 CoT 的企业级 AI Agent 架构设计、端侧模型量化与硬件加速优化，并在 RAG 检索增强、多模态记忆系统及 AI 工程化落地方面拥有系统的方法论与丰富实战经验。

七、研修费用及证书

项　目	说明
测　评	本次培训结束后，将进行专业测评考试，经考核合格，可申请以下证书： 证书可在官方网站查询，同时可作为任职、定级、评职、招投标重要参考依据。
证书类型	A类：由中国软件行业协会颁发**《端侧AI与智能体应用工程师》**、**《人工智能系统架构工程师》**高级职业技术水平证书（二选一）
	B类：可在获得A类证书的基础上再申报一本由工信系统颁发**《人工智能应用工程师》**、**《大模型开发工程师》**高级职业技术证书（二选一）
费用标准	A类：3980元/人（含报名费、培训费、专家费、考核建档及申报证书费） B类：5380元/人（含A+B两本证书费用）
缴费方式	附件二：报名回执表

八、报名咨询方式

联 系 人：郭老师	电　　话：010-85913702
手　　机：17610465556（同微信）	报名邮箱：csia_org@yeah.net
报名材料及流程：	1. 填写完整报名回执表（附件（二）） 2. 2寸电子版证件照（蓝底白底均可） 3. 汇款凭证（如选择电汇） 4. 报名成功后加入专属学习交流群
内训定制、监督电话： 010-85913702	

附件:(二)

《AI Agent 与端侧 AI 深度实践：从理论架构到工程落地的闭环实战》高级研修班报名回执表

<table>
<tr><td>单位名称</td><td colspan="7"></td></tr>
<tr><td>收件地址</td><td colspan="2"></td><td>姓名</td><td></td><td>电话</td><td colspan="2"></td></tr>
<tr><td rowspan="2">参会费用</td><td colspan="7">☐A 类：3980 元/人。费用包含参会费用（考核建档及一本证书费等）</td></tr>
<tr><td colspan="7">☐B 类：5380 元/人。费用包含参会费用（考核建档及两本证书费等）</td></tr>
<tr><td>参会情况</td><td colspan="7">人数：（ ）人，费用：（ ） 元人民币</td></tr>
<tr><td>学员姓名</td><td>类别</td><td>性别</td><td>学历</td><td>部门/职位</td><td>手机号码</td><td>电子邮箱</td><td>身份证号</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="5">开票信息</td><td colspan="7">发票类型： ☐培训费 ☐其他 （如需其他类别请注明）</td></tr>
<tr><td colspan="3" rowspan="4">☐增值税普通发票

☐增值税专用发票</td><td colspan="4">公司名称：</td></tr>
<tr><td colspan="4">纳税人识别号：</td></tr>
<tr><td colspan="4">单位地址及电话：</td></tr>
<tr><td colspan="4">开户行及账号：</td></tr>
<tr><td>汇款须知</td><td colspan="7">开户名：中国软件行业协会
开户行：中国工商银行北京海淀西区支行营业室
账号：0200004509014490109</td></tr>
<tr><td>参加本次学习想要解决的实际问题?</td><td colspan="7">1.
2.
3.</td></tr>
<tr><td>注意事项</td><td colspan="7">1. 参训的学员需将报名回执表送至会务组或协会指定邮箱
2. 培训前 1 天建立学习群并告知详细课程安排等事宜
3. 汇款后需提交汇款凭证（传真或电子邮件均可）</td></tr>
<tr><td colspan="4">联 系 人：郭老师</td><td colspan="4">电 话：010-85913702</td></tr>
<tr><td colspan="4">手 机：17610465556（同微信）</td><td colspan="4">邮 箱：csia_org@yeah.net</td></tr>
</table>