CS309
OBJECT-ORIENTED ANALYSIS AND DESIGN

Yuqun Zhang （张煜群）
Department of Computer Science and Engineering
Southern University of Science and Technology

Who and Where Am I?

- Dr. Yuqun Zhang （张煜群）
- Research Interests: LLM-based Software Engineering, Software Analysis, Testing, and Security (Fuzz Testing, Taint Analysis, Software Component Analysis, etc.)
- Email: zhangyq@sustech.edu.cn
- Office: Room 610, Engineering Building South
- Office Hours: 2-4pm, Tuesday, or appointment by email

A LITTLE SOMETHING ABOUT ME…

My Styles and Rules

- Casual
- Interaction
- Mutual Respect
- NO CHEATING!!!!!
  - You may work together in this class, as specified on each specific assignment. Do **NOT** use any resource without citation.

Teaching Assistant

- Jiahong Xiang (香佳宏)
- Hanrui Qi (戚涵睿)
- Jiaming Liu (刘家铭)
- Zirui Zhou (周子睿)
- Xingyu Zhou (周兴雨)

Textbooks

- Freeman et al., *Head First Design Patterns*
- *Martin Fowler, Refactoring*
- Block, *Effective Java*
- Zeller and Krinke, *Essential Open Source Toolset: Programming with Eclipse, JUnit, CVS, Bugzilla, Ant, Tcl/TX and More*
- McConnell, *Code Complete: A Practical Handbook of Software Construction*
- Pilone, *UML 2.0 Pocket Reference*

Evaluation and Grading

- Weekly Lab Tutorials– 20% (for now)
- Project – 35%
  - Agentic applications
  - Group of 5 (before the end of next week)
    - Please be subject to the group size!!
  - 3 presentations (proposal, progress, final)
  - 1 written report
- Exams – 35%
  - Final:
    - What's on an exam? Anything from any aspect of class, including lab sections.
    - No hints (重点)
- In-Class Exercises/Attendance – 10%
  - Spontaneous (That means in general I do not call the roll. But I have my own moves  )

You could say this is a “breathing” class (not a “氵” class).

I just want you to be happy in this semester.

ALRIGHT, LET'S GET REAL

Expectations

- You're going to have to "own" your education in this class
  - I have a feeling this is going to be an awesome semester…
- But…
  - Expect that I may not be able to give you an immediate answer (I'm alright if my response to your question is "I don't know," so you're going to have to be alright with that, too)
  - I (or the TAs) WILL always try to help find you the answers you need in a timely fashion. Be patient.

Tips of Handling Problems

- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself

Tips of Handling Problems

- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.

Tips of Handling Problems

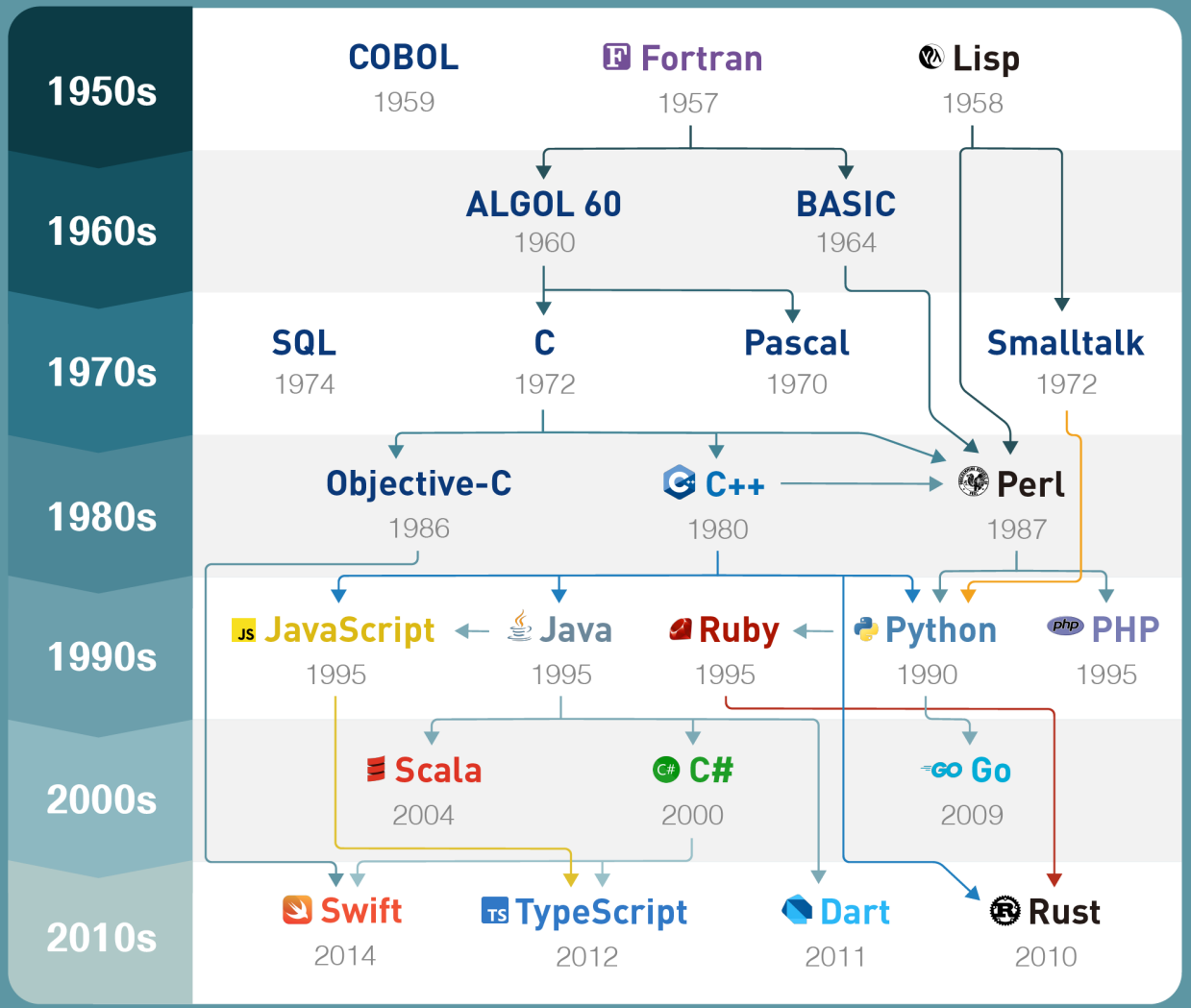
- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.
  - if not working, then ask us

Tips of Handling Problems

- Once you encounter problems (theory or practice), you are expected to
  - first, try your real best to solve them by yourself
  - if not working, try to talk with your cohorts.
  - if not working, then ask us
- If we find that you are not paying effort by yourself, we would be reluctant to help you at later time.

TECKY ACADEMY

Timeline of Programming Languages



What You Would Learn

- Of course the object-oriented design and analysis
- Typically, you are going to learn something about
  - requirement engineering (UML)

UML Examples

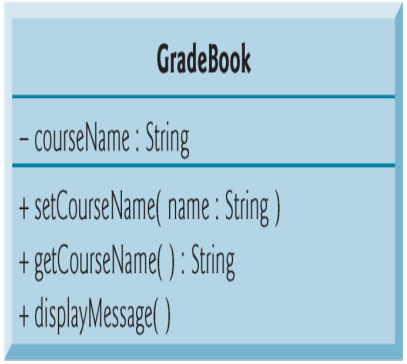


Fig. 7.3 | UML class diagram for class `GradeBook`.

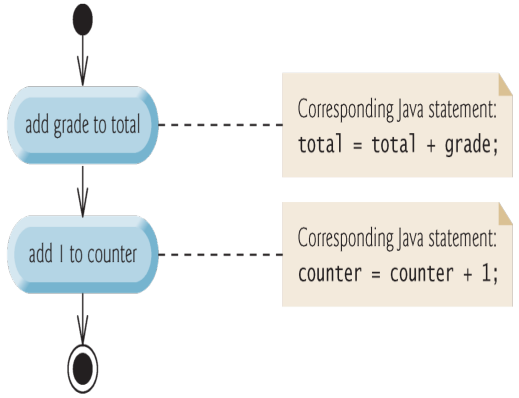


Fig. 3.1 | Sequence structure activity diagram.

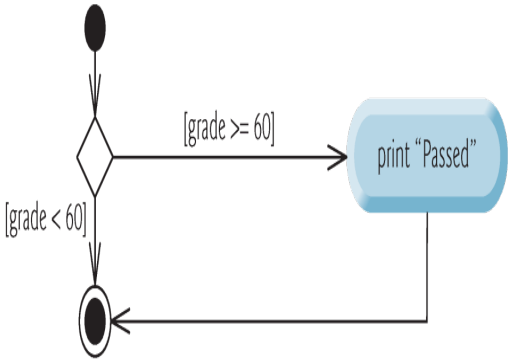


Fig. 3.2 | `if` single-selection statement UML activity diagram.

What You Would Learn

- Of course the object-oriented design and analysis
- Typically, you are going to learn something about
  - requirement engineering (UML)
  - design pattern (including information hiding, design principles,etc)
  - refactoring
  - LLM-based software engineering?

I know what you are thinking about

- With the tools below, what are the points of learning the course?

I know what you are thinking about

- With the tools below, what are the points of learning the course?

  - They bring higher requirements for software engineers in “piloting” projects.
  - They bring more challenges for software maintenance.
  - They bring potentially more tasks for software engineers

Your Projects

- Each group picks one problem from a pool
- We created 5 projects for you. Yet you can work on your own if you want to.
- Come talk to me if you want to come up with your own ideas. DO EXPECT THAT YOUR IDEAS MIGHT BE ASSIGNED WITH A LOWER STARTING SCORE.
- One contact person is needed for each group (This year, all the project requirements are proposed by student assistants. It is pretty necessary to contact them during your progress. They are the bosses!).
- I will cover your cost, e.g., tokens!

The Project List

- Intelligent Learning Assistant Agent (RAG + Homework Tutoring):
- When utilizing large language models (LLMs) to acquire new knowledge or address homework-related questions, students often encounter a key limitation: most general-purpose LLMs lack sufficient expertise in specialized domains or perform inadequately in them, thereby failing to meet learners' practical needs. This project aims to design and implement a learning-oriented intelligent assistant that integrates textbook-based retrieval-augmented question answering and homework tutoring functionalities. The system should allow users to upload textbooks and pose queries, with the agent autonomously planning tasks, leveraging textbook content to improve response quality. Additionally, users can upload homework problems to obtain problem-solving approaches and reference materials (rather than direct answers, in order to prevent academic misconduct).

The Project List

- Multi-Agent Collaborative Task System (Multi-Agent):
- In real world, complex tasks often require collaboration among multiple roles with different areas of expertise. Traditional single-agent systems may sometimes struggle to handle such complexity, as they cannot adequately address multiple dimensions of a task simultaneously. Consequently, multi-agent collaboration architectures have gained attention. For example, Manus does not rely solely on a single-agent model but instead employs a distributed multi-agent collaboration framework, functioning like a team of specialized sub-agents—including planning agents, code execution agents, browser agents, and others—all coordinated under a central controller. In this project, you are required to design and implement a multi-agent collaborative system in which multiple agents with distinct responsibilities coordinate and cooperate to jointly accomplish a complex real-world task. The system should be capable of automated task decomposition, role assignment, execution supervision, and results integration.

The Project List

- CCF-A Paper Aggregation DeepResearch Agent:
- In the early stages of academic research, scholars often spend substantial amounts of time searching for and filtering top-tier conference papers within a specific field, as well as extracting key information to form a preliminary literature review. This process is not only time-consuming but also prone to overlooking important works or critical details. This project aims to develop an automated academic research agent capable of autonomously planning and executing a sequence of tasks for a user-specified research domain (e.g., *program repair* or *fuzz testing*). These tasks include automatically retrieving high-quality papers from CCF-A conferences and journals, traversing citation networks for in-depth exploration, extracting structured information, and ultimately generating a comprehensive research survey report.

The Project List

- SUSTech Intelligent Campus Life Assistant:
- In today's fast-paced campus life, students are required to manage fragmented information originating from multiple systems (such as Blackboard, academic administration systems, email, and campus service apps). To address this challenge, this project aims to develop a comprehensive intelligent assistant agent tailored to the Southern University of Science and Technology (SUSTech) campus. This agent will serve as a unified information hub, aggregating essential academic and life-related information, while also offering accurate Q&A services to significantly enhance the convenience and efficiency of campus life. More than a simple information query tool, it is envisioned as a considerate campus companion that can understand user needs and proactively provide services.

The Project List

- Financial Data Extraction and News Aggregation Agent:
- In today's rapidly evolving financial markets, investors and researchers are confronted with an overwhelming volume of news and financial data, including corporate announcements, industry reports, and stock market updates. Given the diversity of sources and the high frequency of updates, manual filtering and organization are both time-consuming and prone to omissions of critical information. To address this challenge, this project proposes the development of a Financial Data Extraction and News Aggregation Agent, which can automatically collect, filter, and generate structured summaries from multiple sources, thereby providing valuable support for investment decision-making and academic research.

Intelligent Learning Assistant Agent

- Basic Requirements (70%)
  - Knowledge Base Construction and Retrieval-Augmented Generation (RAG)
    - Implement automatic parsing and preprocessing of textbooks in PDF format.
    - Construct a vectorized knowledge base (vector database) using embedding techniques.
    - Invoke LLM APIs to generate logically coherent and contextually enriched answers by incorporating retrieved content.
  - Homework Tutoring Functionality
    - Accept homework problems in text or image format, parse the questions, and extract key concepts.
    - Generate solutions based on the textbook knowledge base, providing step-by-step reasoning along with references to source passages.
    - Classify problems according to subject or unit type.
    - Prevent direct answer disclosure and incorporate mechanisms to defend against prompt injection attacks.
  - User Interface
    - Deliver an intuitive and visually appealing interface that supports textbook upload and question input.
    - Enable management of previously uploaded textbook files.
    - Provide a logging or history system for tracking user queries and system responses.
    - Ensure a clear, well-structured, and manageable dialogue interface.

Intelligent Learning Assistant Agent

- Advanced Requirements (30%)
  - Support additional document types for parsing and content extraction (e.g., Word, Markdown, PowerPoint; at least three formats).
  - Implement Agentic RAG: design a more adaptive retrieval process where the LLM determines when retrieval is required.
  - Incorporate global memory mechanisms, inspired by ChatGPT or Claude, via prompt injection or tool calling techniques.
  - Apply reranking models to refine retrieval results.
  - Integrate with at least one third-party service and support exporting conversation content (e.g., Notion, Obsidian, Feishu, Yuque).

Multi-Agent Collaborative Task System

- Basic Requirements (70%)
  - Agent Role Definition and Communication
    - Create multiple agents with defined roles and ensure correct transmission of task information.
    - Support either a centralized coordination mode (all agents communicate exclusively with the central controller) or a decentralized mode (agents can directly communicate and negotiate with one another).
    - Implement mechanisms for task distribution and results aggregation.
    - Provide a logging system to facilitate tracking of the task execution process.
  - Task Execution Capabilities
    - Support at least one complete end-to-end execution scenario (e.g., software development, travel planning, research report writing).
    - Implement basic task decomposition logic, ensuring each agent can independently accomplish its assigned subtask.
    - Produce structured final outputs (e.g., a complete travel plan, an executable code project).
  - Interactive Interface
    - Deliver a modern UI, such as a Web-based interface or command-line interface.
    - Support task input and agent parameter configuration.
    - Display real-time agent status and collaborative artifacts.
    - Clearly present both intermediate outputs and final results.

Multi-Agent Collaborative Task System

- Advanced Requirements (30%)
  - MCP Integration
    - Support integration with [Model Context Protocol (MCP)](https://modelcontextprotocol.io/docs/getting-started/intro) servers using JSON or more user-friendly formats.
    - Extend functionality by adopting tools such as [mcp-auto-install](https://www.npmjs.com/package/@mcpmarket/mcp-auto-install), enabling the model to autonomously configure MCP servers.
  - Browser/GUI Operation
    - Utilize frameworks such as [Browser Use](https://github.com/browser-use/browser-use) or [Computer Use](https://docs.anthropic.com/en/docs/agents-and-tools/tool-use/computer-use-tool), enabling at least one agent to manipulate browsers or other complex GUIs.
    - Allow users to observe agent behavior in real time and pause execution if needed.
    - Enable users to take over the interface manually and intervene in the process.
  - Asynchronous Backend Execution
    - Ensure the system can continue execution asynchronously in the backend even if the user closes the application, until task completion.

CCF-A Paper Aggregation DeepResearch Agent

- Basic Requirements (70%)
  - Multi-Source Retrieval and Filtering
    - Support automatic retrieval of papers from at least two public academic platforms (e.g., arXiv, DBLP, Semantic Scholar).
    - Implement a configurable white list of CCF-A conferences/journals, enabling the agent to automatically filter qualifying papers.
    - Support keyword-based topic searches, along with data cleaning and normalization across different sources to produce a unified format.
  - Deep Search and Data Extraction
    - Implement citation-based deep search. For each retrieved *seed paper*, the agent should automatically collect both references (backward tracing) and citations (forward expansion), with configurable depth (e.g., recursive 2 levels).
    - Implement a memory module to record visited papers, preventing duplicate retrievals or infinite loops in the citation network.
    - Leverage LLMs to automatically extract key structured information from abstracts or full texts, including at minimum: core methods, performance metrics, and datasets used.

CCF-A Paper Aggregation DeepResearch Agent

- Basic Requirements (70%)
  - Structured Aggregation and Report Generation
    - Aggregate all collected paper information (including title, authors, year, venue, abstract, key methods, performance metrics, etc.).
    - Support export of structured data in both CSV and JSON formats for further analysis.
    - Automatically generate a basic Markdown-formatted research report, including an overview, a summary table of papers, and the agent's complete chain of thought and action logs to ensure transparency and interpretability.

CCF-A Paper Aggregation DeepResearch Agent

- Advanced Requirements (30%)
  - Automated Survey Generation
    - Beyond aggregation, enable the agent to perform higher-level analyses and automatically generate a mini-survey.
    - The survey should include: trend analysis of research activity over time; clustering of core methods using LLM-based analysis; and a comparative performance table for specific datasets or tasks to visually present experimental results across papers.
  - Visualization and Interactive Interface
    - Build a Web-based user interface to serve as the agent's control panel and visualization hub.
    - The interface should display the agent's real-time status and its "thought-action-observation" decision-making process.
    - Visualize the citation network, with nodes representing papers and edges representing citation links. Users should be able to click nodes to view paper details, providing an experience similar to *Connected Papers*.
  - Interactive Q&A with Precise Source Tracing
    - Enable the agent to automatically download PDFs of the collected papers and construct a vectorized knowledge base (vector database).
    - Support interactive natural-language Q&A over the collected paper set (e.g., *"Which papers propose methods to improve the accuracy of RAG retrieval?"*).
    - Ensure that the agent's answers include precise source attribution—not only citing the paper but also, where possible, referencing the specific location in the original text (e.g., *"According to Figure 1 and Table 2 of Paper X"*)—to enhance credibility.

SUSTech Intelligent Campus Life Assistant

- Basic Requirements (70%)
  - Automated Integration and Synchronization with at least Two Campus Information Sources
    - Simulate CAS login to securely proxy user access to different campus services.
    - Ensure stable integration with at least two of the following systems:
      - **Blackboard**: retrieve personal schedules and course material updates.
      - **Email system**: receive and perform basic parsing of emails.
      - **Academic affairs system**: retrieve course schedules (excluding automated course registration).
  - Backend Processing and Structured Storage of Core Information
    - Design and implement a database schema to store processed information in a structured manner for subsequent queries and use.
    - Establish scheduled tasks or webhook mechanisms for periodic updates to ensure data timeliness.

SUSTech Intelligent Campus Life Assistant

- Basic Requirements (70%)
  - Intelligent Q&A Agent
    - Enable the agent to query and modify database records through SQL statements or other methods.
    - Provide not only basic data retrieval but also support higher-level planning tasks, such as:
      - Plan my weekly schedule to balance study, leisure, and exercise, and add it to my calendar.
  - Interactive Interface
    - Provide clear commands or UI elements to guide users in utilizing core functionalities.
    - Supply logs or real-time feedback for key operations and status changes.
    - Automatically generate daily or weekly views based on course schedule information.

SUSTech Intelligent Campus Life Assistant

- Advanced Requirements (30%)
  - Real-time voice interaction with the agent (inspired by ChatGPT or Doubao).
  - Agent permission control: mechanisms to prevent unauthorized modifications or leakage of schedules, and defenses against prompt injection leading to SQL injection.
  - Long-context processing: strategies to handle large volumes of information (e.g., efficiently managing 100 emails in a single day) while maximizing recall.
  - Email sending via SMTP through the agent, with appropriate permission safeguards.
  - Model Context Protocol (MCP): support for an extensible MCP interface that allows plug-and-play integration of new information sources.
  - Mobile adaptation: if implemented as a desktop application, ensure mobile support; if implemented as a web application, ensure layouts remain clear and accessible on mobile devices.

Financial Data Extraction and News Aggregation Agent

- Basic Requirements (70%)

The Agent should autonomously plan tasks, invoke external tools, and ultimately deliver the requested financial data or news based on user queries.

  - Financial Data Collection and Analysis
    - Integrate with APIs for stocks, funds, or cryptocurrencies to support queries of both real-time and historical data (e.g., prices, trading volume).
    - Allow users to upload personal financial reports (PDF format) and intelligently extract key data such as assets and positions.
  - News Aggregation and Analysis
    - Retrieve financial news from at least two sources (e.g., Sina Finance, Xueqiu, Yahoo Finance).
    - Provide analytical summaries of financial news (e.g., “Provide ten key financial news items from today along with an analysis”).
    - Support customized queries by target company or industry (e.g., “Provide this week’s news about Tencent”).
    - Enable exporting structured summaries of financial information in Markdown or PDF format.
  - User Interface
    - Provide a user-friendly interface to display dialogues and statistical results.
    - Visualize the Agent’s task-planning process, allowing users to track real-time workflows and better understand the automation mechanism.
    - Allow users to review intermediate outputs at each stage of the Agent’s processing.

Financial Data Extraction and News Aggregation Agent

- Advanced Requirements (30%)
  - Multi-format Data Processing
    - Support unified parsing and aggregation of multiple financial reports in different PDF formats, accurately extracting core financial indicators.
    - Enable reading and extracting data from Excel spreadsheets.
  - Intelligent Dialogue and Contextual Memory
    - Support contextual memory and multi-turn dialogue, enabling users to submit complex queries in natural language (e.g., “Compare the stock price trends of the new energy industry over the past three months”), with the Agent automatically completing the analysis and returning results.
  - Cross-lingual Information Processing
    - Support news sources in both Chinese and English, with automated translation and unified analysis.
    - Allow users to choose between original text, translated text, or bilingual display.
  - Privacy and Compliance Support
    - Implement privacy-preserving mechanisms for user-uploaded financial data (e.g., anonymized storage or local processing).
    - Incorporate compliance checks to avoid the analysis or dissemination of illegal insider information.

Tips for your projects

- Frequently contact your stakeholders. They manage the requirements and have written detailed descriptions.
- Launch your projects ASAP. You don't want to start off just two weeks before the final ddl.
  - Whoever accomplish the projects and present them on mid-term presentation can be awarded with a bonus of 10% of your final project score.
- If you want to be better graded, you should go for as many bonus points as possible.
- You need to run your deliverables with test cases.
- Don't simply rely on the technical leader (大腿) in your team. We would grade you based on your individual contributions to the team in a rigorous manner.

QUESTIONS?