

# 软件体系结构2024作业1

学号：2024140793

姓名：李国泽

班级：310

## 分析软件体系结构对软件设计的指导与实践——以关键词索引生成系统（KWIC）为例

软件体系结构是计算机科学与工程领域的重要内容之一，其旨在研究如何通过高层次的抽象设计，构建具有良好性能、可维护性和扩展性的复杂软件系统。在实际开发中，软件体系结构指导着软件系统设计时的模块划分、交互方式以及数据流动路径，是连接需求分析与代码实现的关键桥梁。以下，本文将以前文软件体系结构2024作业1中所实现的关键词索引生成系统（KWIC）为例分析软件体系结构对软件设计的指导与实践价值。

关键词索引生成系统的核心任务是对输入的文本数据生成关键词的上下文索引。若以一次性程序的标准来看，这种任务实现起来并不复杂，甚至可以通过简单的脚本程序完成。然而，当我们目标从临时使用转向长期维护和扩展，设计一个要具体应用到某项目中的搜索系统时，那么对该系统的要求便截然不同。一个脚本程序可以忽略代码的组织与结构，但一个具备可维护性和扩展性的系统必须经过体系结构层面的分析和设计。

我们以属性驱动的架构设计方法ADD去分析。该系统的功能需求是支持用户提供多行文本输入，并能够在输入完成后输出一个所有行、经过所有循环移位后的排序列表。在后续还有一些潜在的需求，比如输入可以以Socket形式输入，部分模块可以自定义等等。除了基本的功能需求外，系统的设计还受到一定的限制条件约束。系统需要使用JAVA语言实现，且输入输出均以指定文本文件的形式给出。整个系统与外界的交互相对独立，在基础要求中该系统仅以文本文件的形式与外部实现输入输出交互，而在扩展要求中系统允许进一步以Socket的形式实现输入交互。由于整个系统可以基于数据流实现功能分工，因此我们选择采用流水线架构来组织系统的整体设计，这样既能够符合KWIC系统的处理流程，也为后续扩展预留了足够的空间。

在这个架构上我们将系统进一步分解为一个主模块 KWICSystem 和多个协作模块 Input、Process、Shift、Sort 和 Output。主模块 KWICSystem 负责整体流程的协调和控制，依次调用各功能模块完成数据读取、预处理、关键词位移生成、排序处理以及结果输出。为实现模块可独立拓展以及可修改排序顺序、增加处理步骤的要求，每个功能模块均采用面向对象的封装方式，通过接口定义操作细节，由具体实现类完成功能执行，以为后续功能的扩展和修改提供灵活的支持。

之后我们进一步为每个模块划分责任，明确限制，确认质量属性目标并给出实现方案。

KWICSystem模块作为系统的主类，负责协调各个模块的工作。它通过命令行接收用户输入的参数，根据这些参数初始化不同的功能模块，如输入模块、数据处理模块、排序模块和输出模块，并调度各个子模块的执行完成文本循环排序后的结果输出。该模块的设计核心属性要求是高效简洁地管理各个模块之间的工作。因此在实现过程中，除了输入参数的读取，我们在实现时尽可能的以调用形式完成该模块逻辑，减少其他计算逻辑。

InputHandler模块定义了系统的输入接口，负责从不同的数据源读取输入数据，并将其逐行返回为一个列表。FileInputHandler是InputHandler接口的一个实现，专门用于从本地文件中读取数据。它通过指定文件路径逐行读取文件内容，并将结果以列表的形式返回。另一种实现是SocketInputHandler，它负责从网络Socket连接中读取数据。通过监听指定端口，SocketInputHandler接收来自客户端的文本数据，并逐行存储到列表中，直到接收到结束信号。为了确保系统的可靠性，我们在实现中引入了异常处理机制，以便在输入过程中发生错误时，能够及时给出异常提示，并提供相应的解决方案。这样，系统能够在遇到问题时及时反馈，并保证输入数据的稳定和正确。

ProcessStrategy模块定义了一个处理输入数据的策略接口，旨在对每行数据进行必要的转换或清洗，以便后续的处理模块能够正确使用这些数据。该模块的核心任务是确保数据的预处理过程能够适应不同的需求和规则。DefaultProcessStrategy是该接口的默认实现，它对输入数据进行最简单的空处理，即不对数据做任何修改，直接保留原始格式。由于在需求分析中未对数据处理有更具体的要求，我们仅为系统预留了接口，以支持未来可能的处理策略，例如数据清洗或格式转换等。

ShiftGenerator模块定义了生成循环位移版本的核心逻辑。其主要任务是基于每行文本生成所有可能的循环位移版本，并将这些版本作为排序模块的输入。通过循环位移算法，系统能够生成每行文本的不同排列，以便后续的排序操作能够对这些排列进行处理。CyclicShiftGenerator是ShiftGenerator接口的具体实现，它使用循环位移算法生成输入文本的所有可能位移版本，并将结果以列表形式返回。该模块的设计重点是高效性，尤其在处理大规模文本数据时，必须确保位移生成过程不会成为系统性能的瓶颈。因此，CyclicShiftGenerator在实现过程中对性能进行了优化，我们使用了一些优秀的算法包以应对可能出现的大数据量和复杂计算需求，同时保持较高的执行效率。

SortStrategy模块定义了数据排序的策略接口，旨在根据特定的排序规则对循环位移生成的文本数据进行排序。其核心职责是确保文本数据按照用户指定的规则进行排列，以满足不同的应用需求。DefaultSortStrategy是该接口的默认实现，它按照字典顺序对数据进行排序，将文本行按升序排列。为了适应更复杂的需求，CustomSortStrategy模块支持自定义排序规则，如根据文本行中的单词数量进行排序，或者根据特定关键词的优先级对文本数据进行排序。该模块的设计强调高效性，为了确保排序操作能够在处理大规模文本数据时保持良好的性能表现，我们同样引入了优秀性能的排序算法以供使用。此外，模块具有灵活性和扩展性，能够根据用户需求进行定制，以满足不同的排序要求。

OutputHandler模块定义了输出数据的接口，主要任务是将排序后的文本数据以用户指定的格式保存或展示。FileOutputHandler是OutputHandler接口的具体实现，负责将排序结果逐行写入指定的文件。它确保输出的文件格式正确，内容完整，并符合用户的需求。在该模块的设计中，可靠性是关键属性，因此我们采用了异常处理机制，以确保在输出过程中遇到任何问题时，能够

及时捕获错误并提供解决方案。无论是文件路径不存在、权限不足，还是其他可能的输出错误，异常处理都能确保系统的稳定运行，避免因输出错误导致整个系统崩溃或数据丢失。

SocketClient模块是一个独立的测试模块，主要用于验证和测试Socket输入模式的功能。作为客户端，SocketClient连接到指定的Socket服务器，通过监听指定端口接收来自服务器的数据，并在控制台显示接收到的文本内容。该模块的设计重点是验证Socket连接的稳定性和数据接收的准确性，因此它不参与主程序的核心流程，而是作为一个辅助工具，确保Socket输入功能能够在真实环境中正常工作。通过使用SocketClient模块，可以有效地模拟客户端与服务器之间的数据传输，确保系统在实际应用中能够顺利接收并处理来自网络的文本数据。

本文通过分析关键词索引生成系统（KWIC）的软件体系结构，探讨了体系结构在软件设计中的指导作用。系统采用了属性驱动架构设计方法（ADD）和流水线架构，明确划分了各个模块的责任，如输入模块、数据处理模块、排序模块和输出模块等。每个模块通过接口与实现类的分离方式实现了灵活性和可扩展性，确保系统能够根据需求变化进行调整。此外，系统通过引入异常处理机制，保障了输入输出的可靠性。测试模块SocketClient验证了系统在实际环境中的稳定性，确保了数据接收的准确性。整体设计不仅满足当前需求，还为未来的扩展和维护提供了坚实基础，体现了良好的可维护性、可扩展性和高效性。