

Rule-based Soft Compressor

Source Code File \mathcal{D}

Code-block Splitter

C++

Java

JS

Rule-based Summary Token Number Calculation

IF I_x is High, THEN $y^1 = a^1 I_x + b^1$

IF I_x is Medium, THEN $y^2 = a^2 I_x + b^2$

IF I_x is Low, THEN $y^3 = a^3 I_x + b^3$

$$n = \text{clamp}(\sum_{k=1}^K \mu^k y^k)$$

summary tokens $n = 2$

<PAD> <PAD>

LM

summary embeddings 2 1 <PAD>

LM

$n = 2$

3 2 1 <PAD> <PAD>

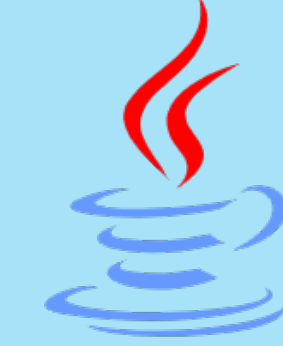
LM

5

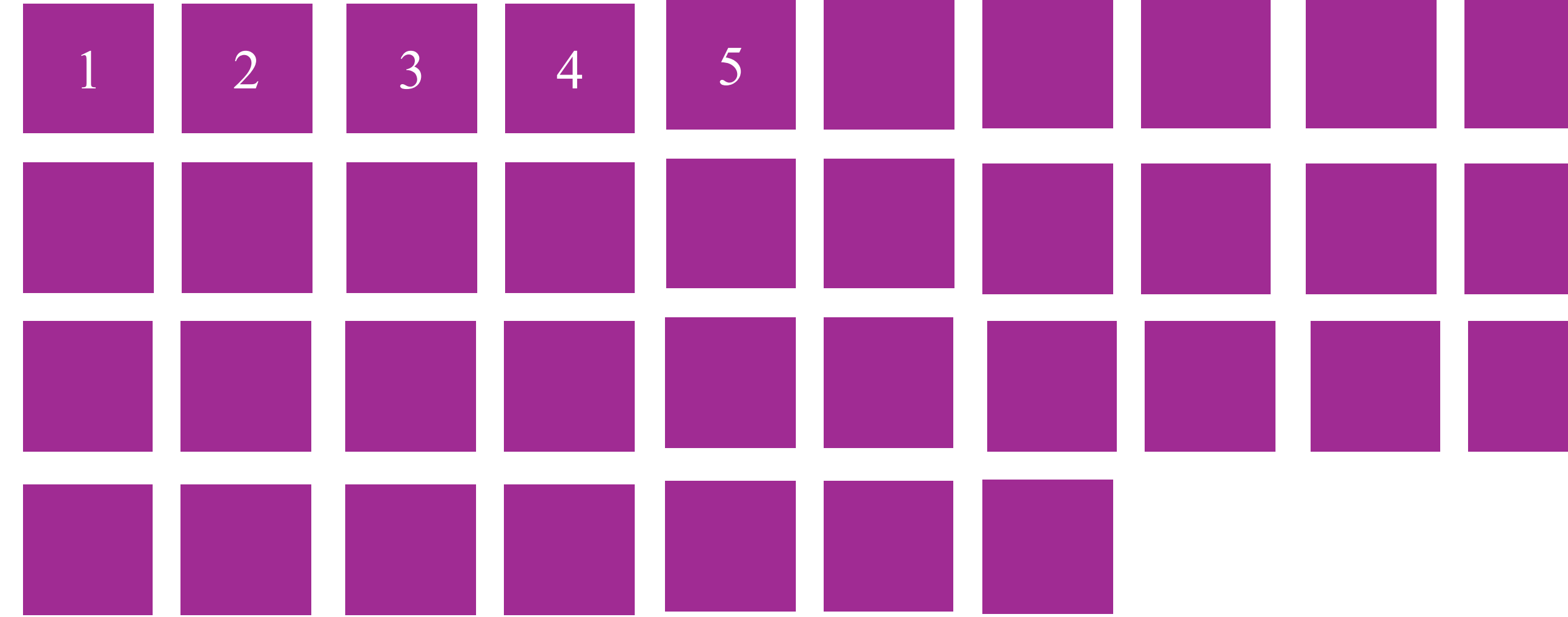
4

Multi-Agent Code Generation

The Original Large-scale Project



Memory \mathcal{M}



$\Delta \mathcal{C}$

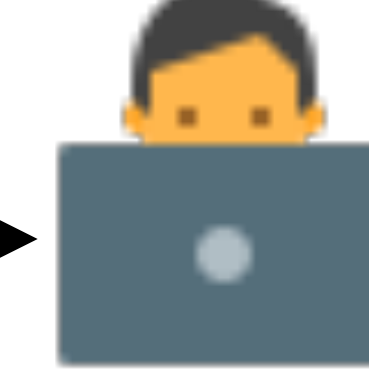


Reviewer

$\Delta \mathcal{M}$



New Features



Coder

```
1 // main.cpp
2 #include <iostream>
3 #include "SystemManager.h"
4 #include "AgentManager.h"
5 #include "TaskScheduler.h"
6 #include "Logger.h"
7 #include "ConfigLoader.h"
8
9 int main(int argc, char* argv[]) {
10     Logger::init("log/config.json");
11     Logger::info("System starting...");
12     ConfigLoader config("config/project_config.json");
13     if (!config.isValid()) {
14         Logger::error("Invalid configuration. Exiting.");
15         return -1;
16     }
17     SystemManager systemManager;
18     systemManager.loadConfig(config);
19     AgentManager agentManager;
20     agentManager.initializeAgents(config.getAgentConfig());
21     TaskScheduler scheduler(&agentManager);
22     scheduler.loadTasks(config.getTaskList());
23     Logger::info("Starting main execution loop...");
24     scheduler.run();
25     Logger::info("Shutting down...");
26     agentManager.shutdown();
27     systemManager.cleanup();
28     Logger::info("System exited successfully.");
29     return 0;
30 }
```



Tester