

# Software Design Patterns

## Practice 1: Design principles, analysis and architectural patterns (evaluated out of 50 points)

Given the rise of artificial intelligence in 2025, a group of entrepreneurs have come up with the idea of providing a pay-per-use service for artificial intelligence agents in order for different companies to provide their employees with access to new artificial intelligence systems.

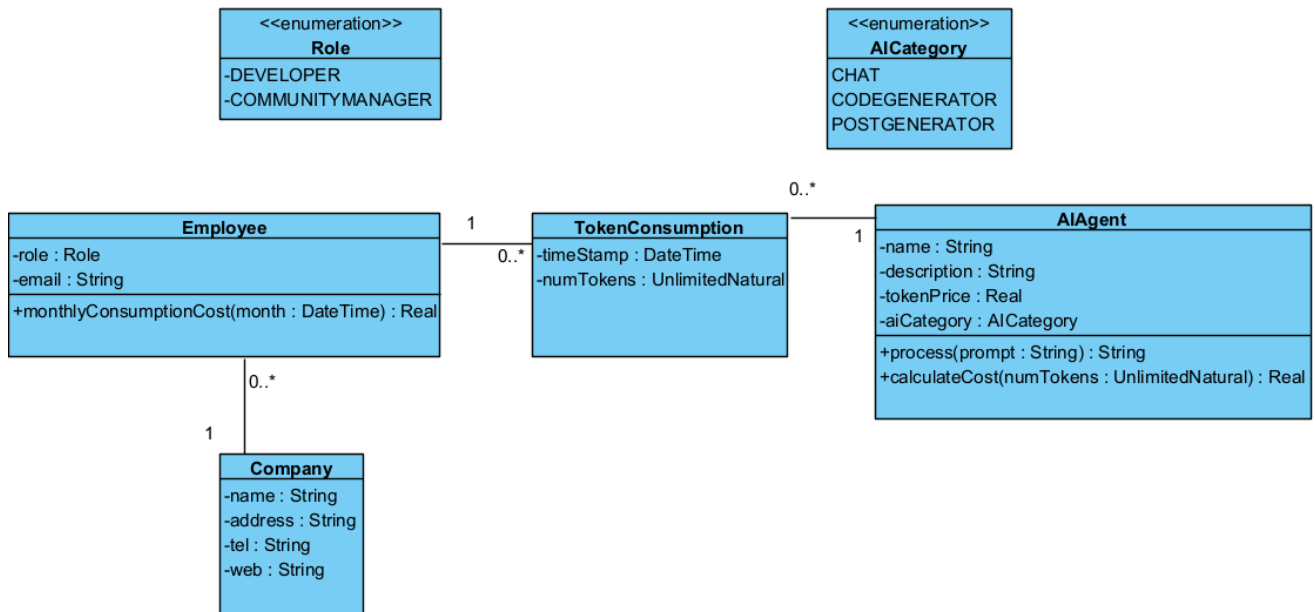
Each of the companies (*Company*) that subscribe to the service can register as many employees (*Employee*) as they see fit. In order to register employees (users of the system) all that is required is an email address and the employee's role. The role allows the system to determine whether an employee is a 'developer' or a 'community manager', which are the types of employees for which it will initially offer services.

The system offers a service of artificial intelligence agents (AIAgent) specialised in conversational chats, generating posts for social networks and generating code for programmers. The system has a list of agents that are identified by name, a short description, a category and a cost per token. Currently, the agents are categorised into 3 types (Chat, Code generator and Post generator), although more types are expected in the future.

Each time an employee uses/executes an agent, a number of tokens are consumed that carry a price (TokenConsumption).

In order to be able to invoice the company monthly for services rendered, the system registers all agent consumptions made by the different users of your company on a monthly basis.

A possible model representing the system described above is presented below:



### Integrity Constraints

- A Company is identified by *name*.
- An Employee is identified by *email*.
- An AI Agent is identified by *name*.
- TokenConsumption is identified by *timeStamp* and the *email* of the employee.

## Question 1 (10 points)

### Statement

- A) **(2 Points)** Which of the analysis patterns is being applied in the model? Identify which, if any, of the patterns are applied, and describe whether it is applied as it is in the pattern catalogue or is a variation of it, in which case explain the differences.
- B) **(2 Points)** Describe an example where one of the analysis patterns could be applied to the model in the statement. (The exercise consists of: naming which pattern could be applied, where in the model/problem it would be applied and what advantages it would bring over the current model. The class diagram of how the pattern is applied does not have to be provided).
- C) **(2 Points)** Does the Employee class comply with the Single Responsibility Principle (SRP) (Yes/No and Why?)?
- D) **(2 Points)** Does the use of the enumerated AICategory violate any principles? Note that the system may evolve and incorporate new categories (Yes/No and Why?).
- E) **(2 Points)** Does the AIAgent model correctly follow the Liskov Substitution Principle (LSP)? Note that the system can evolve and incorporate new agents of different kinds (Yes/No and Why?).

## Question 2 (10 points)

### Statement

In order to accommodate new types of complex AI Agents to assist its employees in increasingly elaborate tasks, the company wants to modify the system to support current AI Agents that could be called simple agents that are specialised in performing a single function, such as natural language processing, image classification or data analysis. However, to solve more complex problems, composite agents will be included, which can use both simple agents and other composite agents.

This means that a composite agent can delegate tasks to its internal components, combining their responses in a coherent way. In this way, the company can build highly specialised agents without needing to modify the code of the individual agents.

**Note:** When providing the class diagram you can represent new or modified classes.

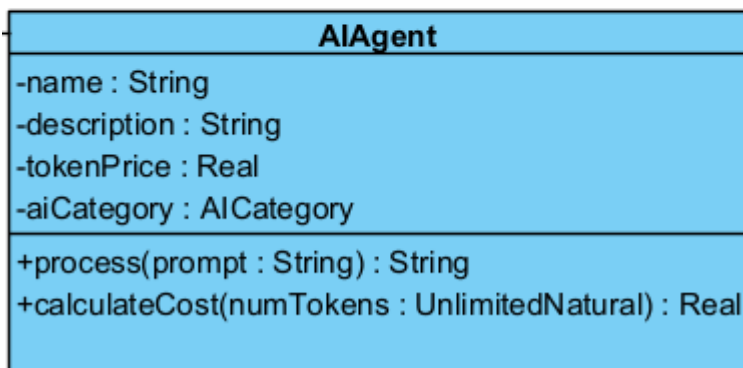
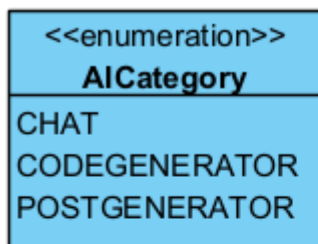
- A) **(2 Points)** Explain what pattern you would use to address this situation.
- B) **(4 Points)** Show the corresponding class diagram. Add any integrity constraints you consider relevant.
- C) **(4 Points)** Name and describe one method/operation that only makes sense for composite agents and one that makes sense for all agents. You can use existing methods in the class diagram of the statement or you can invent new methods.

## Question 3 (15 points)

### Statement

Starting from the initial model of the practice statement, the `AIAgent` has a method called `process(prompt:String): String` that allows to ask the agent to process a query from a user by using an AI model. `AIAgents` currently use internally an AI engine that is hardcoded in the function code and that forces us to modify the code if we want to update and use a more current engine.

Below is the part of the model relevant to the exercise along with the implementation of the `AIAgent` class.



```
public class AIAgent {

    private String name;
    private String description;
    private double tokenPrice;
    private AICategory aiCategory;

    public AIAgent(String name, String description, double tokenPrice, AICategory
    iaCategory ) {
        this.name = name;
        this.description = description;
        this.tokenPrice = tokenPrice;
        this.aiCategory = aiCategory;
    }

    //Method to invoke the different AI engines depending on
    //the category the agent belongs to
    public String process(String prompt){
        if(this.name == "ChatGPT"){
            ChatObject oChatGPT = new ChatGPTAgent();
            return oChatGPT.ProcesarModeloChatGPT.(prompt, "o4mini");
        }else if(this.name == "Llama"){
            LlamaObject oLlama = new LlamaAgent();
            return oLlama.InvokeModel(prompt, "llama7");
        }else if(this.name == "GROK"){
            GROKobject oGrok = new GrokModel();
            return oGrok.query(prompt);
        }else{
            throw new Exception("Engine not found");
        }
    }

    public double calculateCost(int numTokens){
        return this.tokenPrice*numTokens;
    }
}
```

As AI is advancing very fast and new and better AI engines are appearing every day, we want AI Agents to be configurable and to be able to set/configure which AI engine is running for which Agent. Currently we are considering using ChatGPT, Llama and GROK engines. The system must allow the AI Agent logic to be decoupled from its implementation through **dependency injection**, and allow us to invoke the AI engines through the method `processRequest(prompt: String): String`

When applying dependency injection, you can use constructor injection.

It is required:

- A) **(5 Points)** Using the dependency injection pattern, provide a UML diagram that incorporates this new requirement to support the 3 AI engines that you want to use for AgentIA. To do this you can create and/or modify the UML elements you require.
- B) **(5 Points)** Provide the code for the AI Agent class and the new classes you added in the previous section.
- C) **(5 Points)** Create an example where the same AI Agent is used with different engines without modifying your code. You can modify the AI Agent class to support engine switching.

Note: You do not need to add all the classes from the original diagram, but only those that are strictly related to the problem of the exercise.

## Question 4 (15 points)

### Statement

Based on the model of the statement (without taking into account the changes in the requirements of previous exercises).

It has been decided that the system we want to build will follow a layered architecture (with a presentation layer, a domain layer and a technical services layer). We want to define the presentation layer of our system and to do so we apply the Model, View, Controller (MVC) architecture pattern. Propose a class diagram where the MVC is applied for the ExecuteAgent use case. For each class, define the declaration of the operations corresponding to the processing in the presentation layer when the user starts the Execute Agent use case and until the system displays the message 'Agent successfully executed'. Describe briefly each of the operations.

The Execute Agent use case is described as follows:

Name: Execute Agent

Summary of functionality: The employee selects and executes an agent.

Main flow of events:

- 1) The system displays the different menu options available to the user
- 2) The user selects the Use Agent menu option.
- 3) The system displays in a window all the available agents.
- 4) The user selects the agent he/she wants to execute.
- 5) The user fills in the Prompt of the agent
- 6) The user executes the agent (presses OK).
- 7) The system returns the result of the agent execution
- 8) The system displays a success message.



It is required:

- A) **(5 points)** Class diagram of the MVC Model corresponding to this case, with the classes involved and the methods necessary to complete this functionality.
- B) **(5 points)** Explanation of each of the methods in the class diagram.
- C) **(5 points)** Draw the sequence diagram corresponding to the Execute Agent operation.