

SupremeHeadAddAssets

30/09/2024

```
def remove_protocol(self, protocol): self.protocols.remove(protocol)
```

#Other methods for controlling swarm asse

In this example,

Finished: Design and implement a new class or module for the supreme head of the swarm, which will have the necessary functions and protocols for the financial agents

Starting task: Test the functionality of the supreme head module by simulating different Financial agent scenarios, such as sending transactions, and verify that it performs as expected

- Generating response...

Executing: Test the functionality of the supreme head module by simulating different financial agent scenarios, such as sending transactions, and verify that it performs as expected

To achieve the overall objective and complete the sub-task, you can add the following additional code to the swarm:

1. Define the SupremeHead class:

Python

Copy Code

```
class SupremeHead:

def _init(self):

self.assets = []

self.agents = []

self.protocols = []

def add_asset(self, asset): self.assets.append(asset)

def add_agent(self, agent): self.agents.append(agent)

def add_protocol(self, protocol):

self.protocols.append(protocol)

def send_transaction(self, sender, receiver

# Implement the logic to send transact:

# Here, you can use the agents' protoco

pass

def simulate_agent_scenarios(self):

# Implement the logic to simulate diffe

# This can include sending transactions pass
```

1. Instantiate the SupremeHead and add assets, agents, and protocols:

Python

Copy Code

```
supreme_head SupremeHead()
```

```
#Add assets asset1 Asset(...)
```

```
asset2 Asset(...)
```

```
supreme_head.add_asset(asset1)
```