**AntFarm Simulator**

**Overview:**

The AntFarm Simulator is a command-line program that allows users to create and manage ant colonies in a simulated environment. Users can spawn ant farms, allocate resources to them, run simulation cycles, and view the summary of the colonies' statuses.

**Program Components:**

**Classes:**

**Ant**  
Represents an individual ant with specific attributes.

* **Attributes:**
* **type:** Specifies the type of ant (e.g., worker or warrior).
* **strength:** Represents the strength level of the ant.
* **foodConsumption:** Amount of food consumed by the ant.
* **Methods:**
* **getStrength():** Returns the strength of the ant.
* **getFoodConsumption():** Returns the food consumption rate of the ant.
* **getType():** Returns the type of the ant.

**AntFarm**  
Represents a specific ant farm with workers, warriors, and food resources.

* **Attributes:**
* **species:** The species of the ants in the farm.
* **workers:** The number of worker ants in the farm.
* **warriors:** The number of warrior ants in the farm.
* **food:** The food available in the farm.
* **Methods**:
* **addResources(resource, amount):** Adds resources (food, workers, or warriors) to the farm.
* **tick():** Simulates the passage of time, where ants consume food. Reports starvation if food is insufficient.
* **printSummary(id):** Displays the current status of the ant farm.

**Meadow**  
Manages and contains two ant farms. It is implemented as a singleton to ensure only one instance exists.

* **Attributes**:
* **antFarm1:** A pointer to the first ant farm.
* **antFarm2:** A pointer to the second ant farm.
* **instance:** Static instance of the Meadow class.
* **Methods**:
* **getInstance():** Retrieves the singleton instance of the Meadow class.
* **spawnAntFarms():** Creates and initializes two ant farms with default species ("Killer" and "Pansy").
* **giveResources(id, resource, amount):** Allocates resources to a specific ant farm by its ID.
* **tick(ticks):** Simulates a given number of cycles for all ant farms.
* **summary(id):** Displays the summary of the specified ant farm.

**User Interface (Main Function):**

The program interacts with the user through the command line. Users can perform the following actions:

1. **Spawn AntFarms**:

* **Description:** Initializes two ant farms ("Killer" and "Pansy").
* **Output:** "Spawned AntFarms: Killer and Pansy."

1. **Allocate Resources**:

* **Description:** Allocates resources (food, workers, or warriors) to a specific ant farm.
* **Example Usage:** giveResources(1, "Food", 100)
* **Output:** "Gave 100 Food to AntFarm 1."

1. **Simulate Time Cycles**:

* **Description:** Simulates the passage of time for all ant farms.
* **Example Usage:** tick(3)
* **Output:** Prints details of each tick, including food consumption and starvation warnings.

1. **View AntFarm Summary**:

* **Description:** Displays the status of a specified ant farm.
* **Example Usage:** summary(1)
* **Output:** Prints details such as species, food, workers, and warriors.

**Execution Flow:**

1. The program begins by creating a singleton instance of the Meadow class.
2. Two ant farms are spawned and initialized.
3. Resources can be allocated to each farm.
4. Time can be simulated to observe food consumption and starvation effects.
5. The status of each farm can be displayed using the summary method.

**Example Execution:**

**Program Start**:

* Spawn AntFarms:  
  Output: "Spawned AntFarms: Killer and Pansy."

**Allocate Resources**:

* giveResources(1, "Food", 100)  
  Output: "Gave 100 Food to AntFarm 1."
* giveResources(1, "Worker", 30)  
  Output: "Gave 30 Worker to AntFarm 1."

**Simulate Time Cycles**:

* tick(3)  
  **Output:**

Tick: 1

Tick: 2

Tick: 3

**View Summary**:

* summary(1)  
  **Output:**

AntFarm ID: 1

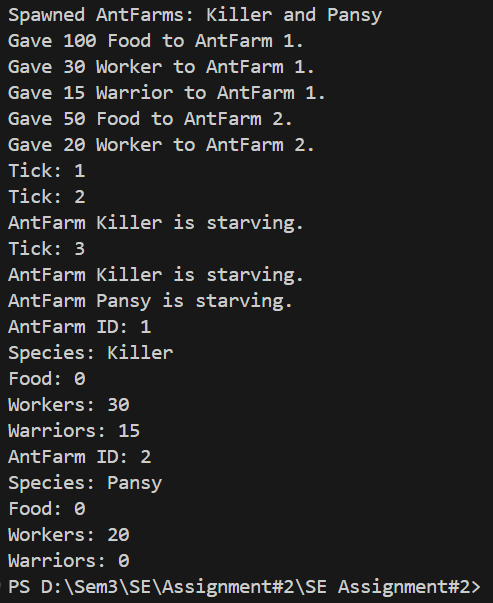
Species: Killer

Food: 70

Workers: 30

Warriors: 15

**Sample Output:**

****

**Future Improvements:**

* Implement detailed behaviors for ants (e.g., fighting or building).
* Introduce dynamic resource generation or decay.
* Expand to support more than two ant farms.
* Add persistence to save and load simulation states.
* Improve user experience by adding a graphical interface.

**Conclusion:**

The AntFarm Simulator provides a framework for managing and simulating ant colonies. It allows users to allocate resources, simulate cycles, and view colony statuses through a user-friendly command-line interface.