**Satellite Command System**

**Overview:**

The Satellite Command System is designed to simulate and control the operations of a satellite. It allows users to issue commands to modify the satellite's orientation, manage solar panels, and collect data. The system uses a command-line interface to interact with the satellite and reflects changes in real-time.

**Components:**

1. **Satellite Class:**
   * **Attributes:**
     + orientation: Current direction of the satellite (e.g., North, South, East, West).
     + solarPanelsStatus: Status of the solar panels (Active or Inactive).
     + dataCollected: Amount of data collected (in units).
   * **Methods:**
     + rotate(String direction): Updates the satellite's orientation based on the specified direction if it is valid.
     + activatePanels(): Sets the solar panels to "Active".
     + deactivatePanels(): Sets the solar panels to "Inactive".
     + collectData(): Increments the data collected by 10 units, but only if the solar panels are active.
     + displayState(): Displays the current state of the satellite, including orientation, solar panels status, and data collected.
2. **SatelliteCommandSystem Class (Main Program):**
   * **Functionality:**
     + Initializes a Satellite instance and provides a command-line interface for user interaction.
     + Accepts and processes commands in a loop:
       - rotate(direction): Changes the satellite's orientation.
       - activatePanels(): Activates the solar panels.
       - deactivatePanels(): Deactivates the solar panels.
       - collectData(): Collects data if the solar panels are active.
       - exit: Terminates the command loop and exits the system.
   * **Command Parsing:**
     + Commands are parsed and executed based on user input.
     + Invalid commands are handled with appropriate error messages.

**Usage:**

The system operates in a command-line environment. Users enter commands to control the satellite and view its current status. For example:

* **Command:** rotate(North)  
  **Action:** Sets the satellite’s orientation to North.
* **Command:** activatePanels()  
  **Action:** Activates the solar panels.
* **Command:** collectData()  
  **Action:** Collects data if the panels are active, otherwise displays an error.
* **Command:** exit  
  **Action:** Exits the command loop and terminates the program.

**Benefits:**

* **Real-Time Control:** Allows for interactive control of the satellite’s features and immediate feedback on actions.
* **User-Friendly:** Provides clear and simple commands for users to operate the satellite.
* **State Management:** Maintains and displays the satellite’s state consistently after each command, ensuring transparency in operations.