Model of Execution

Lecture Question

Question: in a package named "execution", implement the following classes:

- class Battery with
    - A constructor that takes a variable named "charge" of type Int
- class BoomBox with
    - A constructor that takes a variable named "battery" of type Battery
    - A method named "use" that takes no parameters and returns Unit
        - This method will decrease the charge of the BoomBox's battery by 3
        - If the charge of the battery is less than three, this method does nothing
    - A method named "replaceBattery" that takes a Battery as a parameter and returns a Battery
        - This method swaps the input Battery with the Battery currently stored in this BoomBox's state variable
        - The returned Battery is the one that was in the state variable when the method is called
- class FlashLight with
    - A constructor that takes no parameters
        - When a new FlashLight is created, declare a state variable named "battery" of type Battery and set it to a new Battery with 5 charge (ie. Batteries included)
    - A method named "use" that takes no parameters and returns Unit
        - This method will decrease the charge of the FlashLight's battery by 1
        - If the charge is 0, this method does nothing
    - A method named "replaceBattery" that takes a Battery as a parameter and returns a Battery
        - This method swaps the input Battery with the Battery currently stored in this FlashLight's state variable
        - The returned Battery is the one that was in the state variable when the method is called

Testing: In a package named "tests" create a Scala class named "TestBatteries" as a test suite that tests all the functionality listed above

More Memory Examples

- Live Examples

Lecture Question

Question: in a package named "execution", implement the following classes:

- class Battery with
    - A constructor that takes a variable named "charge" of type Int
- class BoomBox with
    - A constructor that takes a variable named "battery" of type Battery
    - A method named "use" that takes no parameters and returns Unit
        - This method will decrease the charge of the BoomBox's battery by 3
        - If the charge of the battery is less than three, this method does nothing
    - A method named "replaceBattery" that takes a Battery as a parameter and returns a Battery
        - This method swaps the input Battery with the Battery currently stored in this BoomBox's state variable
        - The returned Battery is the one that was in the state variable when the method is called
- class FlashLight with
    - A constructor that takes no parameters
        - When a new FlashLight is created, declare a state variable named "battery" of type Battery and set it to a new Battery with 5 charge (ie. Batteries included)
    - A method named "use" that takes no parameters and returns Unit
        - This method will decrease the charge of the FlashLight's battery by 1
        - If the charge is 0, this method does nothing
    - A method named "replaceBattery" that takes a Battery as a parameter and returns a Battery
        - This method swaps the input Battery with the Battery currently stored in this FlashLight's state variable
        - The returned Battery is the one that was in the state variable when the method is called

Testing: In a package named "tests" create a Scala class named "TestBatteries" as a test suite that tests all the functionality listed above