Code Challenge

First-in, First out Animal Shelter.

Specifications

- Read all of these instructions carefully. Name things exactly as described.
- Do all your work in a public repository called `data-structures-and-algorithms`, with a well-formatted, detailed top-level README.md.
- Create a new branch in your repo called `fifo_animal_shelter`.
- Your top-level readme should contain a "Table of Contents" navigation to all of your challenges and implementations so far. (Don't forget to update it!)
- This assignment should be completed within the `challenges` subdirectory of the repository.
- On your branch, create...
    - *C#*: a new .NET Core console project named `FIFOAnimalShelter`. Within your `Program.cs` create a new static method outside of `Main()` following the naming conventions below. Call your newly created method in `Main()` once complete.
    - *JavaScript*: a folder named `fifoAnimalShelter` which contains a file called `fifo-animal-shelter.js`
    - *Python*: a folder named `fifo_animal_shelter` which contains a file called `fifo_animal_shelter.py`
    - *Java*: a folder named `fifoAnimalShelter` which contains a file called `AnimalShelter.java`

- Include any language-specific configuration files required for this challenge to become an individual component, module, library, etc.
    - *NOTE: You can find an example of this configuration for your course in your class lecture repository.*

Feature Tasks

- Create a class called AnimalShelter which holds only dogs and cats. The shelter operates using a *first-in, first-out* approach.
- Implement the following methods:
    - `enqueue(animal)`: adds `animal` to the shelter. `animal` can be either a dog or a cat object.
    - `dequeue(pref)`: returns either a dog or a cat. If `pref` is not `"dog"` or `"cat"` then return null.

Stretch Goal

If a cat or dog isn't preferred, return whichever animal has been waiting in the shelter the longest.

Requirements

Ensure your complete solution follows the standard requirements.

1. Write unit tests
2. Follow the template for a well-formatted README
3. Submit the assignment following these instructions