GRASP JUSTIFICATION FOR A PARTICULAR CLASS

CLASS: PetClinic

OPERATIONS

- **Operations**
    1. Import()
        - The responsibility of the PetClinic class is to act as an Information Expert and Creator to the DAO objects as well as provide Indirection to them. Having the Import function allows to maintain high cohesion as the PetClinic can read in and update its DOAs as needed
    2. Export()
        - The responsibility of the PetClinic class is to act as an Information Expert and Creator to the DAO objects as well as provide Indirection to them. Having the Export function allows to maintain high cohesion as the PetClinic can write to a file to later update its DAOs as needed
    3. isEmployeeEmpty()
        - The responsibility of the PetClinic class is to act as an Information Expert and Creator to the DOA objects as well as provide Indirection to them. The PetClinic Class must know if it's EmployeeDAO is empty ensure the DAO never tries to refer to a null object.
    4. findPet()
        - The findPet method allows for indirection from outside controllers by searching through the PetDAO class by going through the PetClinic class first before doing a DML function inside the PetDAO class. This also supports High cohesion between the PetClinic and its petInventory member variable while also supporting low cohesion as the PetDAO will do the desired DML and not the PetClinic Class
    5. findCustomer()
        - The findCustomer method allows for indirection from outside controllers by searching through the CustomerDAO class by going through the PetClinic class first before doing a DML function inside the CustomerDAO class. This also supports High cohesion between the PetClinic and its custDatabase member variable while also supporting low cohesion as the CustomerDAO will do the desired DML and not the PetClinic Class
    6. findEmployee()
        - The findEmployee method allows for indirection from outside controllers by searching through the EmployeeDAO class by going through the PetClinic class first before doing a DML function inside the EmployeeDAO class. This also supports High cohesion between the

PetClinic and its employDatabase member variable while also supporting low cohesion as the EmployeeDAO will do the desired DML and not the PetClinic Class

7. Adopt()
    - The Adopt method allows for a pure fabrication approach when manipulating the PetClinic Class's member variables. It allows for Indirection for the Sale class by only being able to create a receipt. By using Indirection and Pure Fabrication in this way, we can maintain a lower coupling for the PetClinic class and also a high cohesion between the PetClinic Class and its DAO member variables