Coding Challenges: PetPals, The Pet Adoption Platform

Instructions

- Project submissions should be done through the partcipants' Github repository, and the link should be shared with trainers and Hexavarsity.
- Each section builds upon the previous one, and by the end, you will have a comprehensive application implemented with a strong focus on SQL, control flow statements, loops, arrays, collections, exception handling, database interaction.
- Follow object-oriented principles throughout the project. Use classes and objects to model real world entities, encapsulate data and behavior, and ensure code reusability.
- Throw user defined exceptions from corresponding methods and handled.
- The following Directory structure is to be followed in the application.
    - **entity/model**
        - Create entity classes in this package. All entity class should not have any business logic.
    - **dao**
        - Create Service Provider Interface/Abstract Class to showcase functionalities.
        - Create the implementation class for the above Interface/Abstract Class with db interaction.
    - **exception**
        - Create user defined exceptions in this package and handle exceptions whenever needed.
    - **util**
        - Create a DBPropertyUtil class with a static function which takes property file name as parameter and returns connection string.
        - Create a DBConnUtil class which holds static method which takes connection string as parameter file and returns connection object (Use method defined in DBPropertyUtil class to get the connection String).
    - **Main**
        - Create a class MainModule and demonstrate the functionalities in a menu driven application.

Problem Statement:

PetPals, The Pet Adoption Platform scenario is a software system designed to facilitate the adoption of pets, such as dogs and cats, from shelters or rescue organizations. This platform serves as a digital marketplace where potential adopters can browse and select pets, shelters can list available pets, and donors can contribute to support animal welfare.

Implement OOPs

Create SQL Schema from the pet and user class, use the class attributes for table column names.
1.Create and implement the mentioned class and the structure in your application.

Pet Class:
Attributes:

- Name (string): The name of the pet.
- Age (int): The age of the pet.
- Breed (string): The breed of the pet.

Methods:

- Constructor to initialize Name, Age, and Breed.
- Getters and setters for attributes.
- ToString() method to provide a string representation of the pet.

Dog Class (Inherits from Pet):
Additional Attributes:

- DogBreed (string): The specific breed of the dog.

Additional Methods:

- Constructor to initialize DogBreed.
- Getters and setters for DogBreed.

Cat Class (Inherits from Pet):
Additional Attributes:

- CatColor (string): The color of the cat.

Additional Methods:

- Constructor to initialize CatColor.
- Getters and setters for CatColor.

3.PetShelter Class:
Attributes:

- availablePets (List of Pet): A list to store available pets for adoption.

Methods:

- AddPet(Pet pet): Adds a pet to the list of available pets.
- RemovePet(Pet pet): Removes a pet from the list of available pets.
- ListAvailablePets(): Lists all available pets in the shelter.

4.Donation Class (Abstract):
Attributes:

- DonorName (string): The name of the donor.
- Amount (decimal): The donation amount.

Methods:

- Constructor to initialize DonorName and Amount.
- Abstract method RecordDonation() to record the donation (to be implemented in derived classes).

CashDonation Class (Derived from Donation):
Additional Attributes:

- DonationDate (DateTime): The date of the cash donation.

Additional Methods:

- Constructor to initialize DonationDate.
- Implementation of RecordDonation() to record a cash donation.

ItemDonation Class (Derived from Donation):
Additional Attributes:

- ItemType (string): The type of item donated (e.g., food, toys).

Additional Methods:

- Constructor to initialize ItemType.
- Implementation of RecordDonation() to record an item donation.

5.IAdoptable Interface/Abstract Class:
Methods:

- Adopt(): An abstract method to handle the adoption process.

AdoptionEvent Class:
Attributes:

- Participants (List of IAdoptable): A list of participants (shelters and adopters) in the adoption event.

Methods:

- HostEvent(): Hosts the adoption event.
- RegisterParticipant(IAdoptable participant): Registers a participant for the event.

6.Exceptions handling

Create and implement the following exceptions in your application.

- Invalid Pet Age Handling:
  - In the Pet Adoption Platform, when adding a new pet to a shelter, the age of the pet should be a positive integer. Write a program that prompts the user to input the age of a pet. Implement exception handling to ensure that the input is a positive integer. If the input is not valid, catch the exception and display an error message. If the input is valid, add the pet to the shelter.
- Null Reference Exception Handling:
  - In the Pet Adoption Platform, when displaying the list of available pets in a shelter, it's important to handle situations where a pet's properties (e.g., Name, Age) might be null. Implement exception handling to catch null reference exceptions when accessing properties of pets in the shelter and display a message indicating that the information is missing.
- Insufficient Funds Exception:
  - Suppose the Pet Adoption Platform allows users to make cash donations to shelters. Write a program that prompts the user to enter the donation amount. Implement exception handling to catch situations where the donation amount is less than a minimum allowed amount (e.g., $10). If the donation amount is insufficient, catch the exception and display an error message. Otherwise, process the donation.
- File Handling Exception:
  - In the Pet Adoption Platform, there might be scenarios where the program needs to read data from a file (e.g., a list of pets in a shelter). Write a program that attempts to read data from a file. Implement exception handling to catch any file-related exceptions (e.g., FileNotFoundException) and display an error message if the file is not found or cannot be read.
- Custom Exception for Adoption Errors:

- Design a custom exception class called AdoptionException that inherits from Exception. In the Pet Adoption Platform, use this custom exception to handle adoption-related errors, such as attempting to adopt a pet that is not available or adopting a pet with missing information. Create instances of AdoptionException with different error messages and catch them appropriately in your program.

7.Database Connectivity

Create and implement the following tasks in your application.

- Displaying Pet Listings:
  - Develop a program that connects to the database and retrieves a list of available pets from the "pets" table. Display this list to the user. Ensure that the program handles database connectivity exceptions gracefully, including cases where the database is unreachable.
- Donation Recording:
  - Create a program that records cash donations made by donors. Allow the user to input donor information and the donation amount and insert this data into the "donations" table in the database. Handle exceptions related to database operations, such as database errors or invalid inputs.
- Adoption Event Management:
  - Build a program that connects to the database and retrieves information about upcoming adoption events from the "adoption_events" table. Allow the user to register for an event by adding their details to the "participants" table. Ensure that the program handles database connectivity and insertion exceptions properly.