**Virtual Pet Simulation Program Documentation**

*1. Class Design and OOP Principles:*

**Class Name: `Pet`**

**- Attributes:**

- `name` (string): Represents the name of the pet.

- `hunger` (int): Indicates the hunger level of the pet.

- `happy` (int): Represents the happiness level of the pet.

- `energy` (int): Reflects the energy level of the pet.

- `age` (double): Represents the age of the pet.

**- Constructors:**

- `pet(string n, int h, int ha, int e, double a)`: Initializes the pet with specified attributes, ensuring that the age is not zero or negative.

- `pet(string n)`: Initializes the pet with a name, used when creating a new pet.

**- Member Functions:**

- `setname(string n)`: Sets the name of the pet.

- `getname()`: Returns the name of the pet.

- `hunger\_level(int h)`, `happiness\_level(int ha)`, `energy\_level(int e)`, `age\_level(double a)`: Sets the respective levels of the pet.

- `feed()`, `play()`, `rest()`: Simulate feeding, playing, and resting actions, adjusting attribute levels accordingly.

- `get\_info()`: Displays the information about the pet, ensuring attribute levels are within valid bounds.

- `saveToFile()`, `loadFromFile(const string &fileName)`: Save and load pet information to/from a file.

**- Basic Pet Actions:**

- The program supports basic pet actions such as feeding, playing, resting, and querying the pet's status.

- Be aware of the conditions and logic in the **feed()**, **play()**, **rest()**, and **get\_info()** functions.

**- Advanced Features:**

- Aging: The pet ages, and changes in behavior or needs are simulated as it gets older. Explore the ‘**ageOneYear()’** function.

- Multiple Pets: The ability to have multiple pets with different characteristics is present.

- Save/Load Mechanism:

- Pets can be saved to and loaded from files using the **saveToFile()** and **loadFromFile()** functions.

- The program manages pet data persistence between sessions.

**Design Choices:**

**1. Class Design:**

* The **Pet** class is the central component, representing a virtual pet with attributes like name, hunger, happiness, energy, and age.
* Each action on the pet (e.g., feeding, playing) is encapsulated within member functions to keep the code organized and modular.

**2. Member Functions:**

* Member functions are designed to perform specific actions on the pet, ensuring that the code is readable, maintainable, and adheres to the single responsibility principle.

**3. Error Handling:**

* Error handling has been introduced to address potential issues, such as creating a pet with a negative age or loading data from a non-existent file.

**4. Persistence:**

* The **saveToFile()** and **loadFromFile()** functions provide a mechanism for persisting pet data between sessions, enhancing the user experience.

*2. Sample Interaction Scenario:*

This script demonstrates the user interacting with the virtual pet simulation, creating a new pet, playing with it, saving the session, and loading a previously saved pet. The user has options to perform various actions on the pet and manage multiple pets throughout the session.

Welcome to petsvilla

1. Load a pet

2. Play with the pet

3. Let the pet rest

4. Feed the pet

5. Info of the pet

6. Enter a new pet

7. Save session

Enter your choice: 6

Enter the following info of the new pet:

Enter the name of the pet: Bella

Enter the happiness of the pet: 70

Enter the energy of the pet: 50

Enter the hungriness of the pet: 40

Enter the age of the pet: 2.5

Pet Bella created and saved.

1. Load a pet

2. Play with the pet

3. Let the pet rest

4. Feed the pet

5. Info of the pet

6. Enter a new pet

7. Save session

Enter your choice: 2

Levels after this session:

Name: Bella

Hunger level: 25

Happiness level: 90

Energy level: 30

Age of pet: 2.5

1. Load a pet

2. Play with the pet

3. Let the pet rest

4. Feed the pet

5. Info of the pet

6. Enter a new pet

7. Save session

Enter your choice: 7

Session saved for pet Bella.

1. Load a pet

2. Play with the pet

3. Let the pet rest

4. Feed the pet

5. Info of the pet

6. Enter a new pet

7. Save session

Enter your choice: 1

Which pet do you want to load: Bella

Pet Bella loaded.

1. Load a pet

2. Play with the pet

3. Let the pet rest

4. Feed the pet

5. Info of the pet

6. Enter a new pet

7. Save session