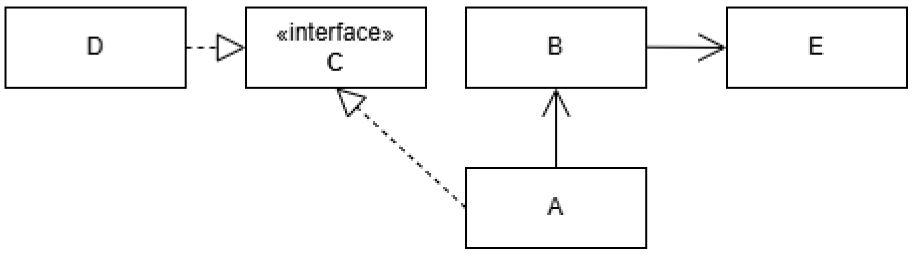
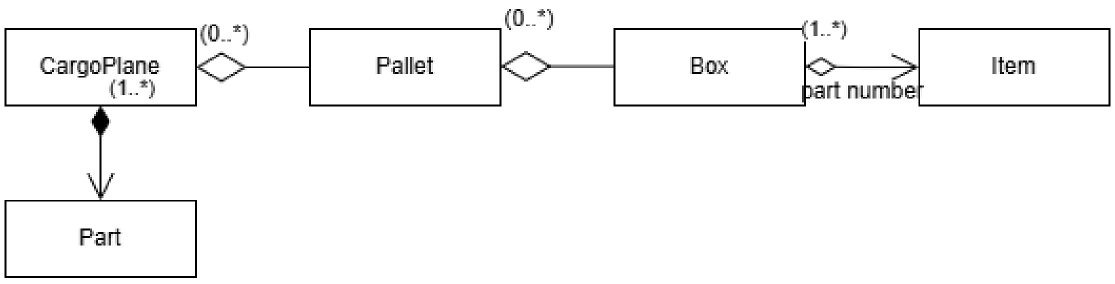
學號: 41147046S　　　　姓名: 楊子萱

1.

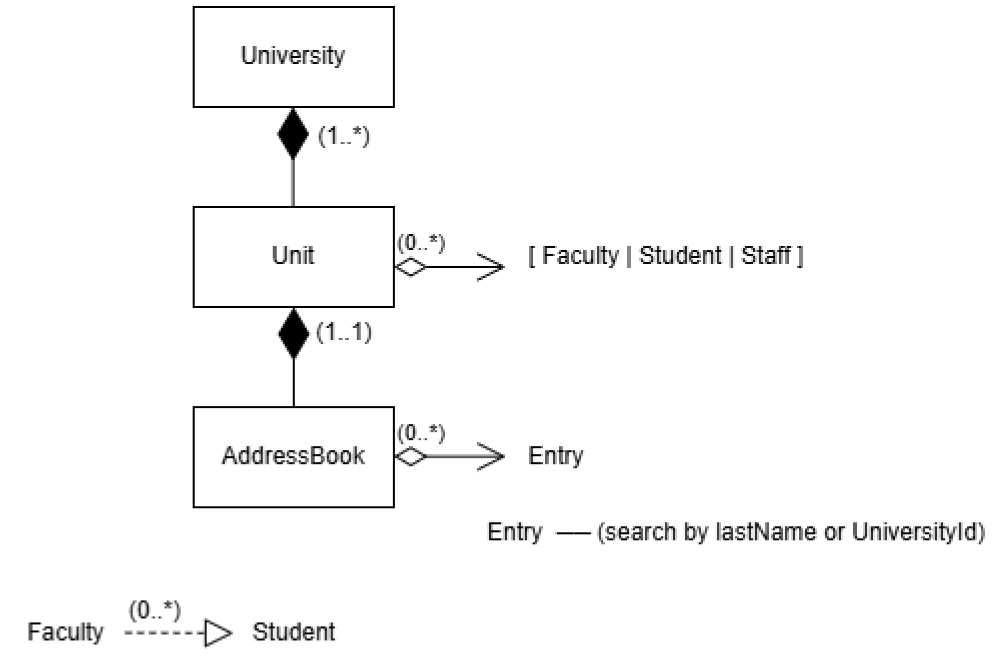
(a)



(b)



(c)

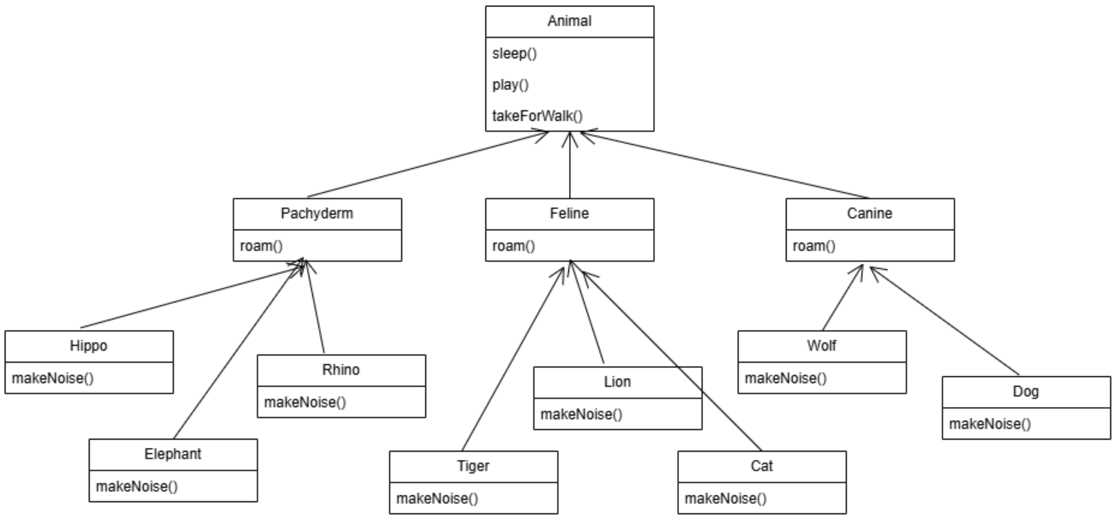


2.

By violating LSP, the engineer has **broken the expected behavior of the system**, leading to incorrect calculations, difficulties in debugging, and potential failure of constraints. This reinforces the importance of ensuring that subclass implementations **preserve the intended behavior of base class methods**.

3.

(a)



Advantages:

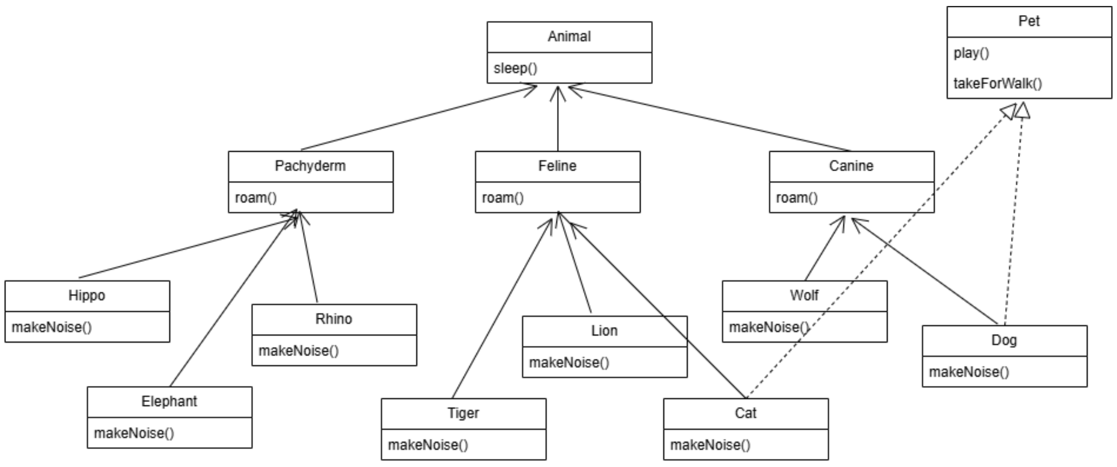
- **Simple structure:** Only need to add methods in the Animal class without requiring additional interfaces or modifications to the inheritance structure.
- **Consistency:** All animals share the same behavior, leading to greater code consistency.

Disadvantages:

- **Violates the Liskov Substitution Principle (LSP):**
  - Non-pet animals such as Elephant, Tiger, or Wolf should not have play() or takeForWalk() methods, but they still inherit them, leading to unreasonable behavior.
- **Potential code pollution:**
  - If Animal needs to provide default implementations for play() and takeForWalk(), these methods become unnecessary for non-pet classes.

- **May require overriding these methods in unrelated classes:**
    - For example, Tiger might need to explicitly override the play() method to throw an exception or return a no-op, which reduces code maintainability.

(b)



Advantages:

1. **Better Adherence to Liskov Substitution Principle (LSP)**
    - Only Dog and Cat implement the Pet interface, preventing unrelated animals (e.g., Tiger, Elephant) from inheriting pet-specific behaviors.
2. **Improved Code Maintainability and Scalability**
    - The interface-based approach allows for easier future extensions. If new pet-related methods are needed, they can be added to Pet without affecting unrelated classes.
3. **More Flexible Design**
    - Other potential pet classes (e.g., Rabbit, Hamster) can implement Pet without modifying the existing class hierarchy. This aligns with **the Open-Closed Principle (OCP)**, as new features can be added without modifying existing classes.

Disadvantages:

1. **Increased Complexity**

- An additional Pet interface introduces more design complexity. Developers must determine which classes should implement it, increasing maintenance efforts.

2. **Inconsistency in Method Access**
   - Since play() and takeForWalk() are not part of Animal, accessing these methods requires explicit type checks or casting, reducing ease of use.
   - This forces developers to use type checking like dynamic_cast (C++) or instanceof (Java), which can introduce runtime inefficiencies.

3. **Potential Over-Engineering**
   - Mixing interfaces with inheritance may lead to **over-design** if the system does not truly require the flexibility.
   - If Pet contains too many methods, it risks becoming a **God Interface**, violating the **Interface Segregation Principle (ISP)** and making implementations harder to manage.