COMP2511

23T1 Week 10

WEDNESDAY 1PM - 4PM (W13B)

FRIDAY 11AM - 2PM (F11A)

Slides by Alvin Cherk (z5311001)

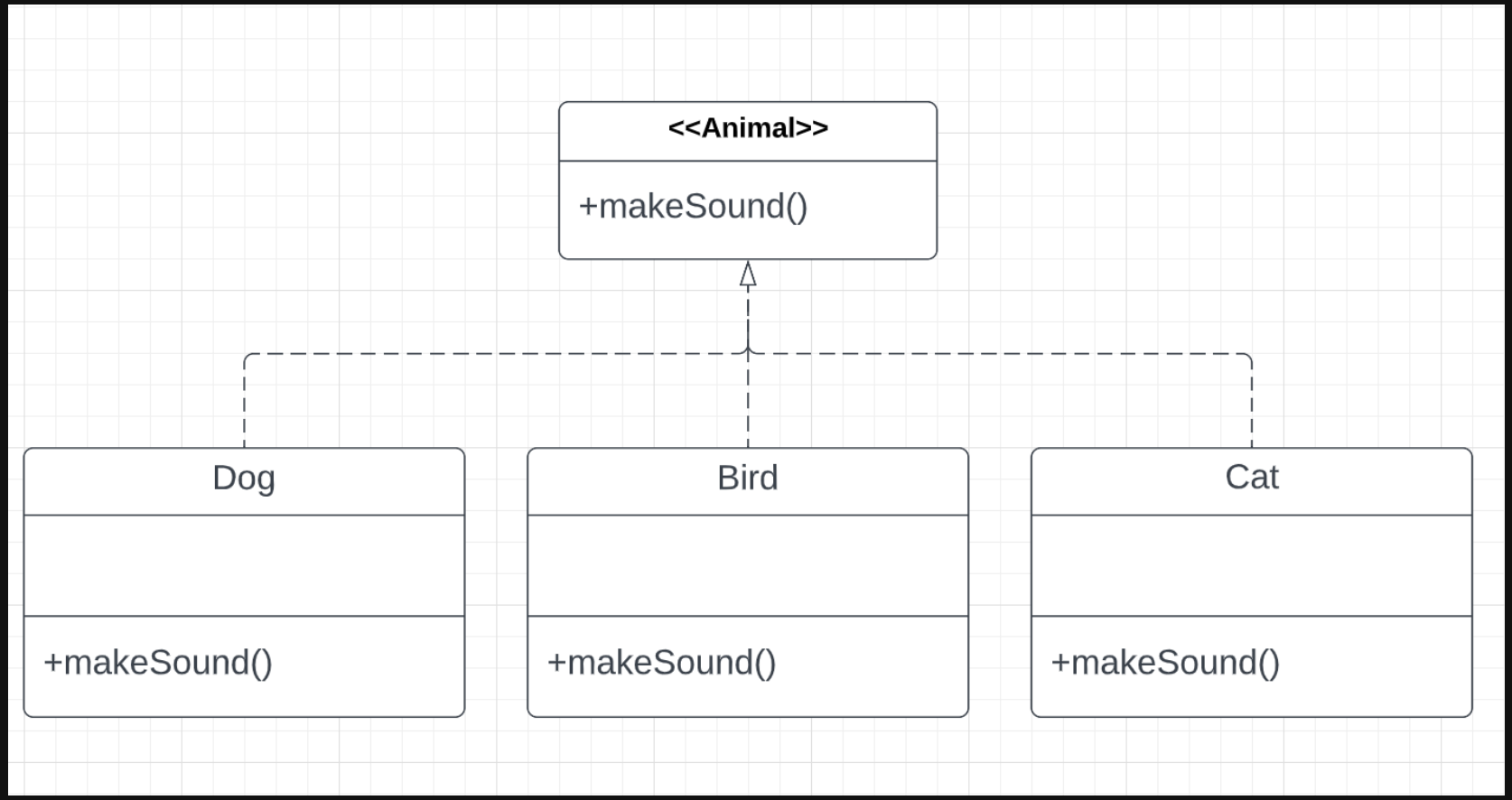
MyExperience

Please fill it in (10 mins)

Visitor Pattern

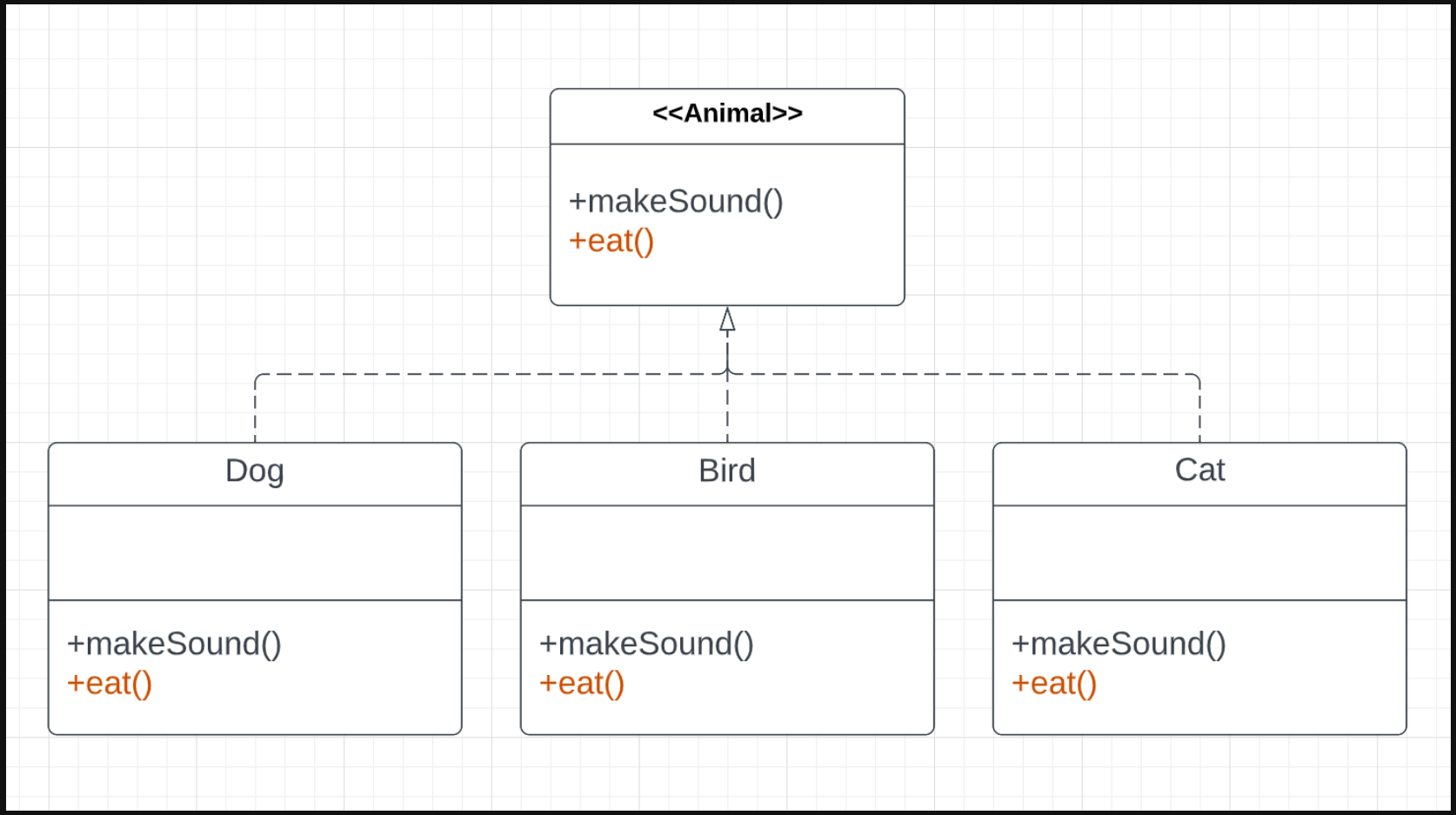
Visitor Pattern

Problem: How do I add extra functionalities to subclasses without violating open/closed principle.



Visitor Pattern

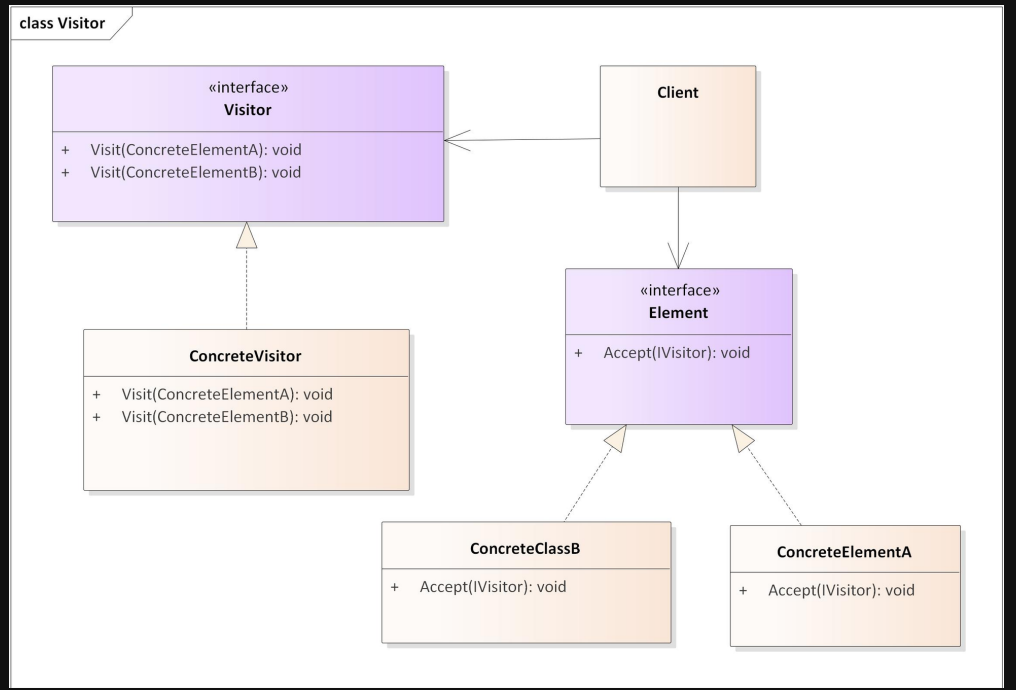
Problem: How do I add extra functionalities to subclasses without violating open/closed principle.

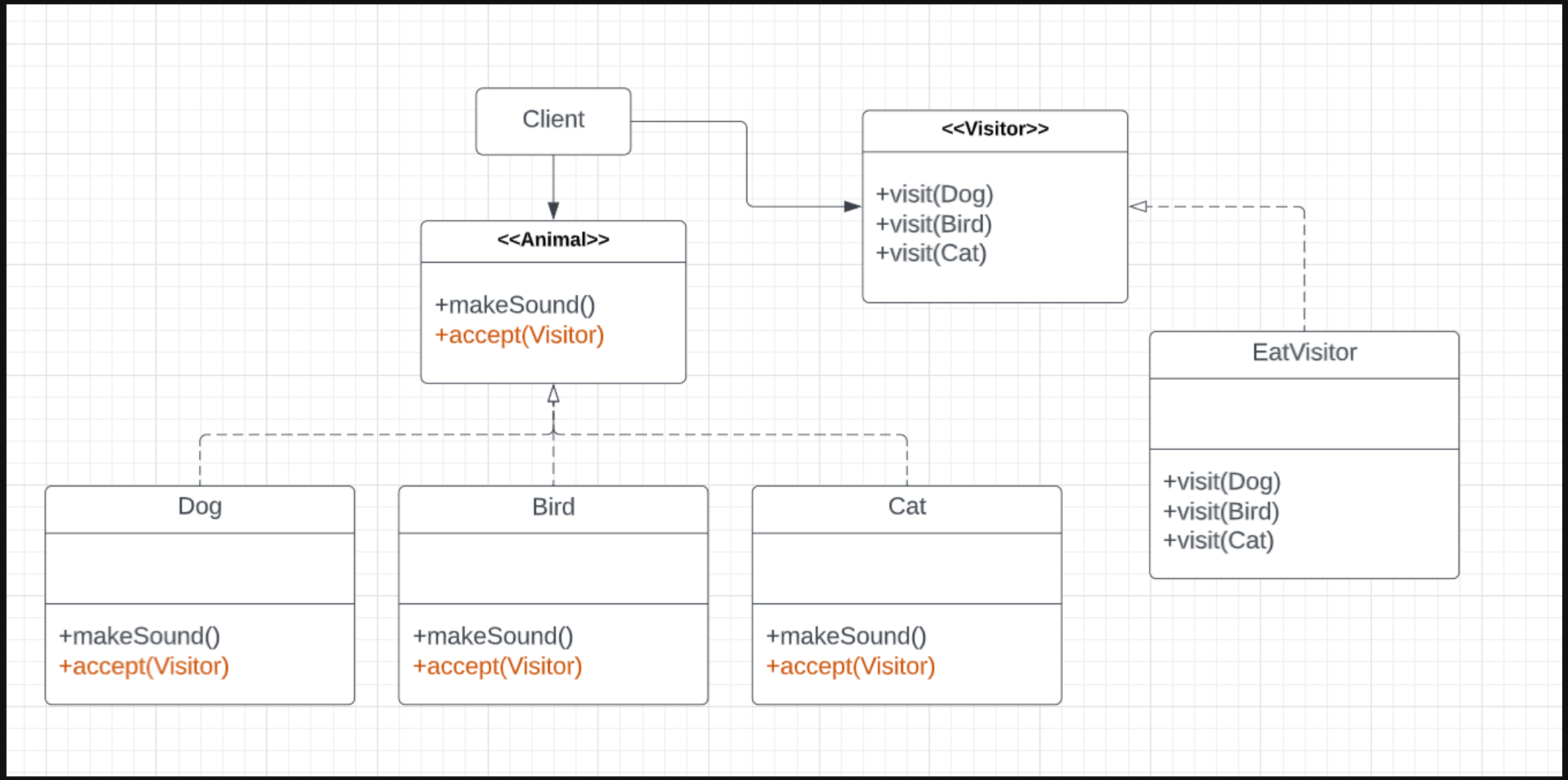


Visitor Pattern

Behavioural Pattern

- Adds extra functionality to class without modifying the original (abides by open closed principle)
- One class/interface (visitor) defines a computation/operation and another (visitable) is responsible for providing data access




Client
<<Visitor>>
+visit(Dog)
+visit(Bird)
+visit(Cat)
<<Animal>>
+makeSound()
+accept(Visitor)
EatVisitor
+visit(Dog)
+visit(Bird)
+visit(Cat)
Dog
+makeSound()
+accept(Visitor)
Bird
+makeSound()
+accept(Visitor)
Cat
+makeSound()
+accept(Visitor)

Kahoot

Exam Tips

Exam Tips

- Practice good exam techniques.
- Get familiar with writing Java & Generics
- Learn the patterns and the differences (YouTube has some really good resources if you don't understand certain ones).
- Know code smells and methods of refactoring

Code Demo

Computer.java

Code Demo

In this scenario we have Computers, Keyboards and Mouses which all are of type **`ComputerComponent`**. We want to be able to 'visit' different types of Computer components by logging the following messages:

```
1 Looking at computer Corelli with memory 500 GB.
2 Looking at keyboard Mechanical keyboard which has 36 keys.
3 Looking at mouse Bluetooth mouse.
```

In particular though, anyone which is visiting a **`Computer`** must be **validated** prior to being able to visit.

Extend/modify the starter code to use the Visitor Pattern to allow different computer components to be visited.

Attendance

Feedback

https://forms.gle/R4sMTTQzPC4vqXSN8