

CMSC 132: Object-Oriented Programming II



UML III
Department of Computer Science
University of Maryland, College Park

UML Examples

- Read UML class diagram
- Try to understand relationships
- Examples
 - Pets & owners
 - Computer disk organization
 - Banking system
 - Home heating system

UML Example – Veterinary System

- Try to read & understand UML diagram



UML Example – Veterinary System

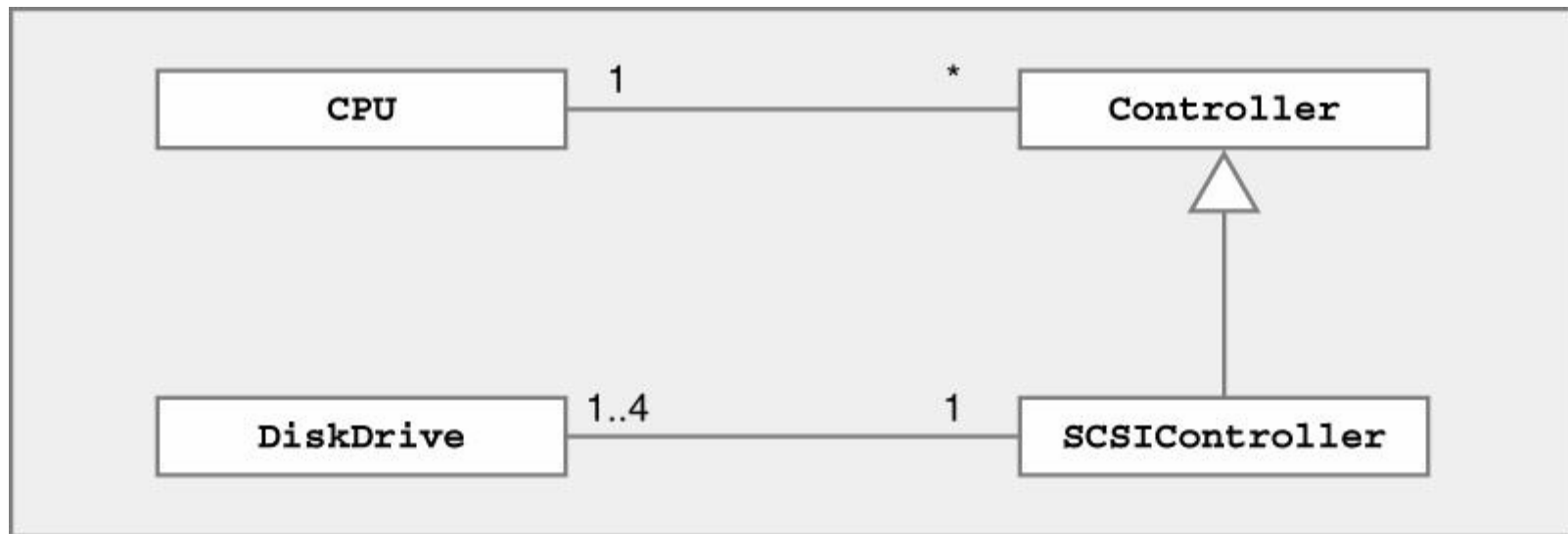
- Try to read & understand UML diagram



- 1 or more Pets associated with 1 PetOwner

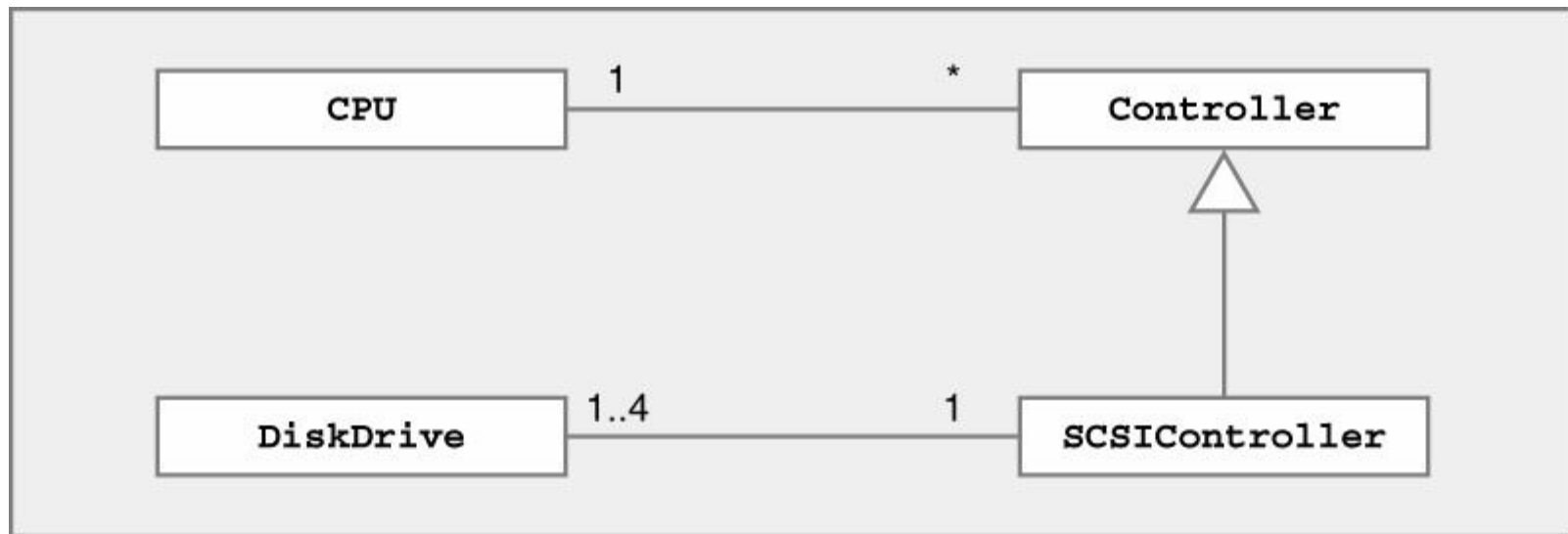
UML Example – Computer System

- Try to read & understand UML diagram



UML Example – Computer System

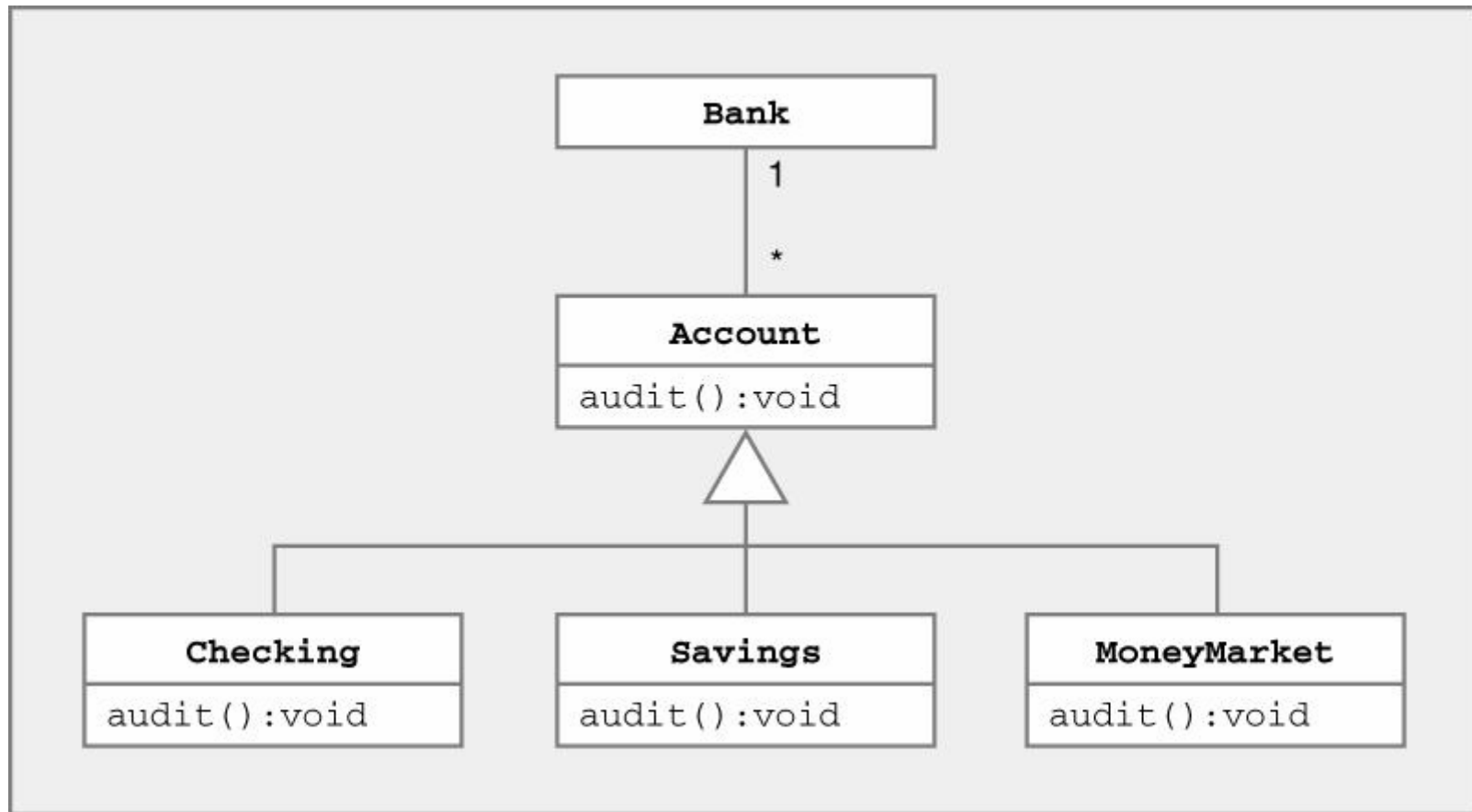
- Try to read & understand UML diagram



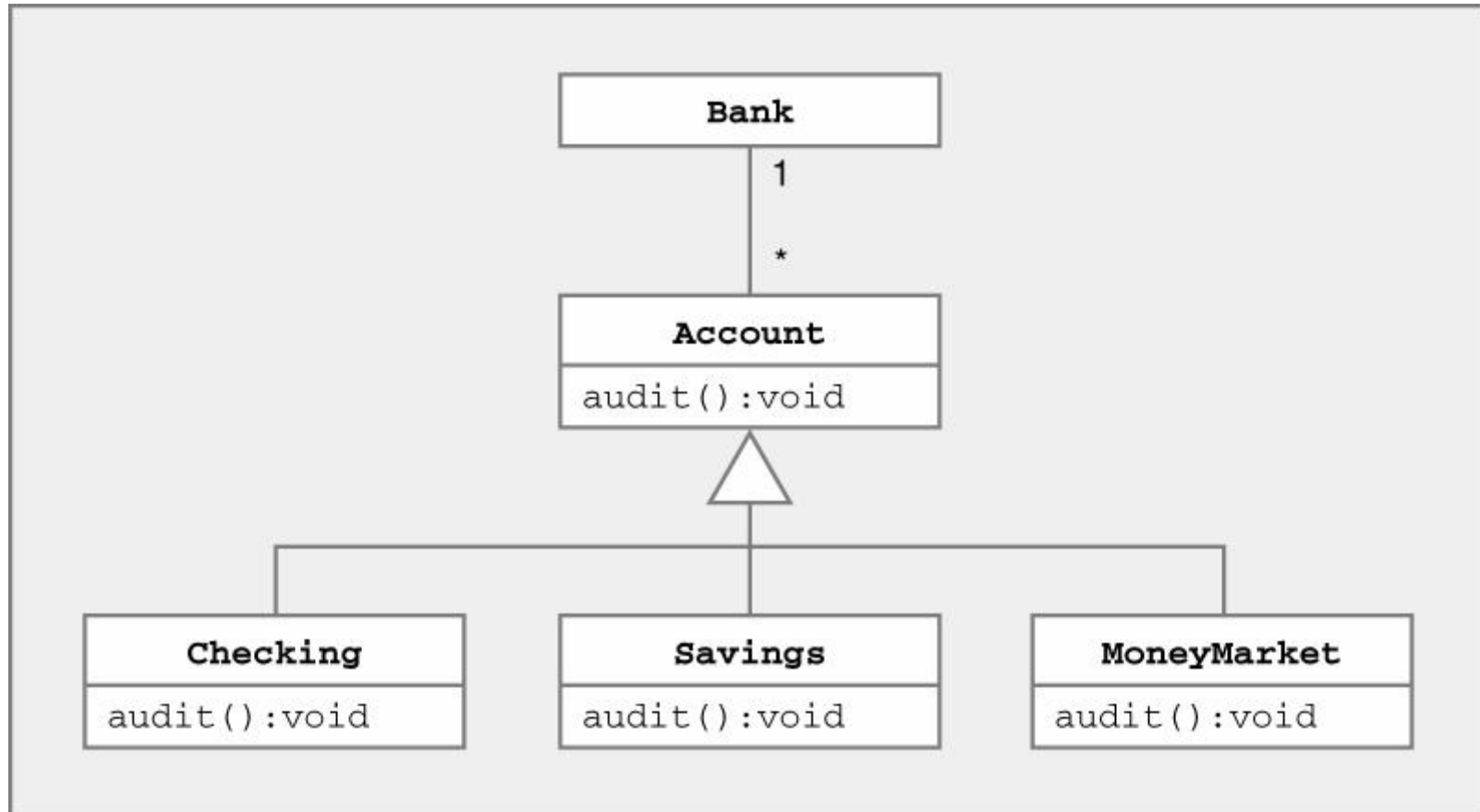
- 1 CPU associated with 0 or more Controllers
- 1-4 DiskDrives associated with 1 SCSIController
- SCSIController is a (specialized) Controller

UML Example – Banking System

- Try to read & understand UML diagram



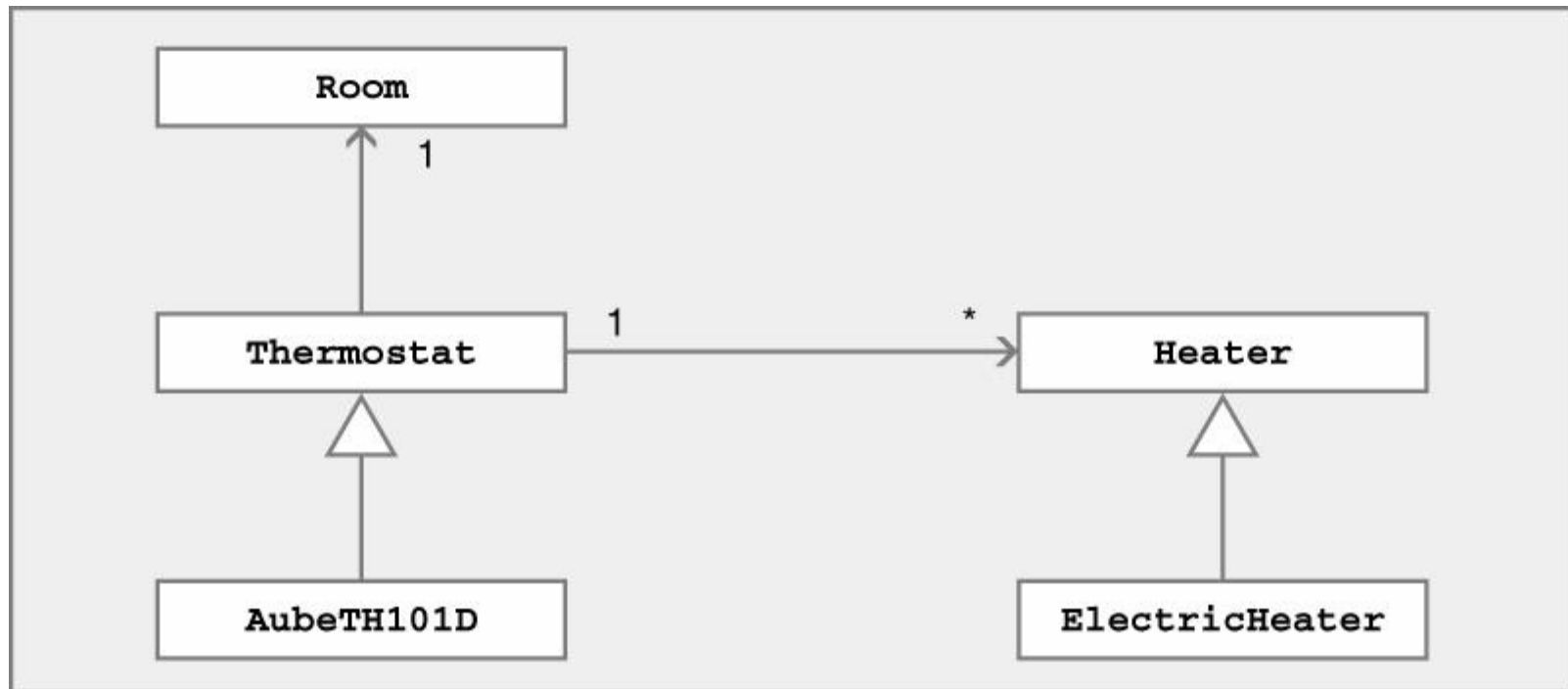
UML Example – Banking System



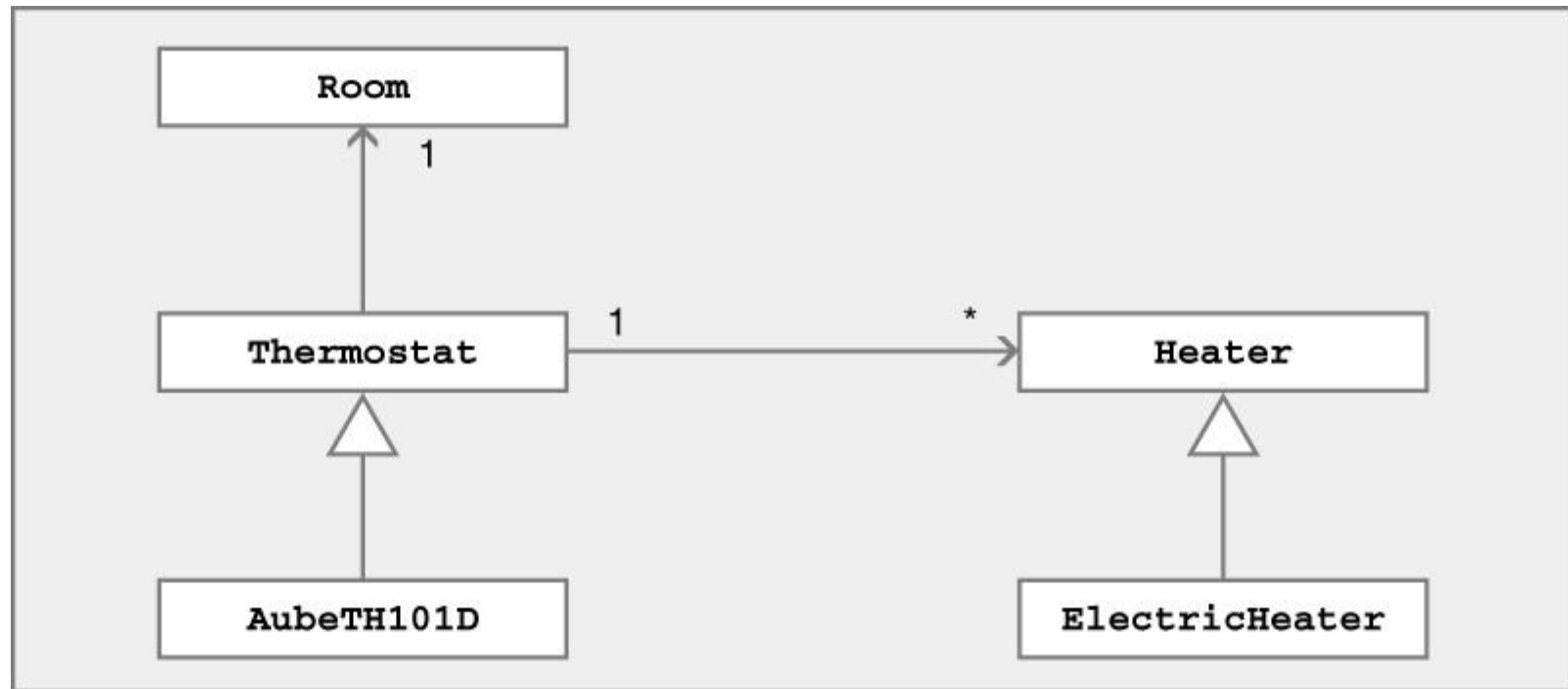
- **1 Bank associated with 0 or more Accounts**
- **Checking, Savings, MoneyMarket are Accounts**

UML Example – Home Heating System

- Try to read & understand UML diagram



UML Example – Home Heating System



- Each Thermostat is associated with 1 Room
- Each Thermostat associated with 0 or more Heaters
- ElectricHeater is a specialized Heater
- AubeTH101D is a specialized Thermostat

UML Class Diagrams ↔ Java

- Different representation of **same** information
 - Name, state, behavior of class
 - Relationship(s) between classes
- Practice deriving one from the other
 - Accurately depicting relationship between classes

UML → Java : Veterinary System

■ UML



■ Java



UML → Java : Veterinary System

■ UML



■ Java

class Pet {
 PetOwner myOwner; **// 1 owner for each pet**
}
class PetOwner {
 Pet [] myPets; **// multiple pets for each owner**
}

Java → UML : Veterinary System

■ Java

```
class Pet {  
    PetOwner myOwner;    // 1 owner for each pet  
}  
class PetOwner {  
    Pet [ ] myPets;    // multiple pets for each owner  
}
```



■ UML

Java → UML : Veterinary System

■ Java

```
class Pet {  
    PetOwner myOwner;    // 1 owner for each pet  
}  
class PetOwner {  
    Pet [ ] myPets;    // multiple pets for each owner  
}
```

■ UML



UML Class Diagrams ↔ Java

■ UML

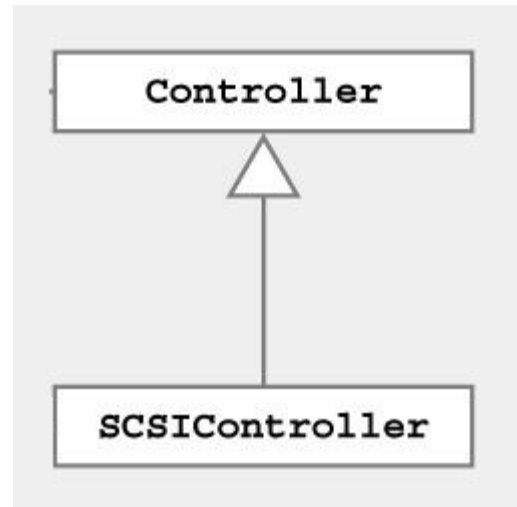


■ Java

class Pet {
 PetOwner myOwner; **// 1 owner for each pet**
}
class PetOwner {
 Pet [] myPets; **// multiple pets for each owner**
}

UML → Java : Computer System

■ UML

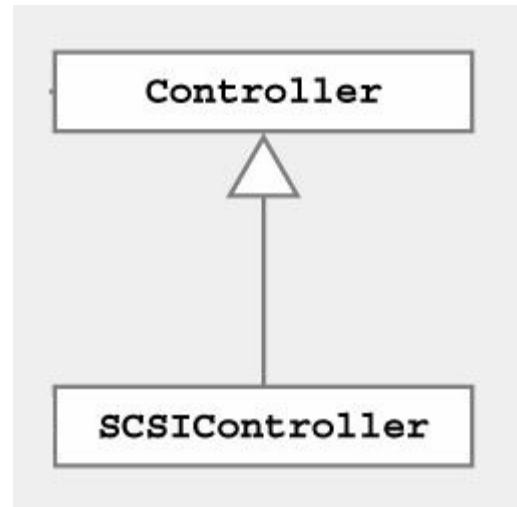


■ Java



UML → Java : Computer System

■ UML

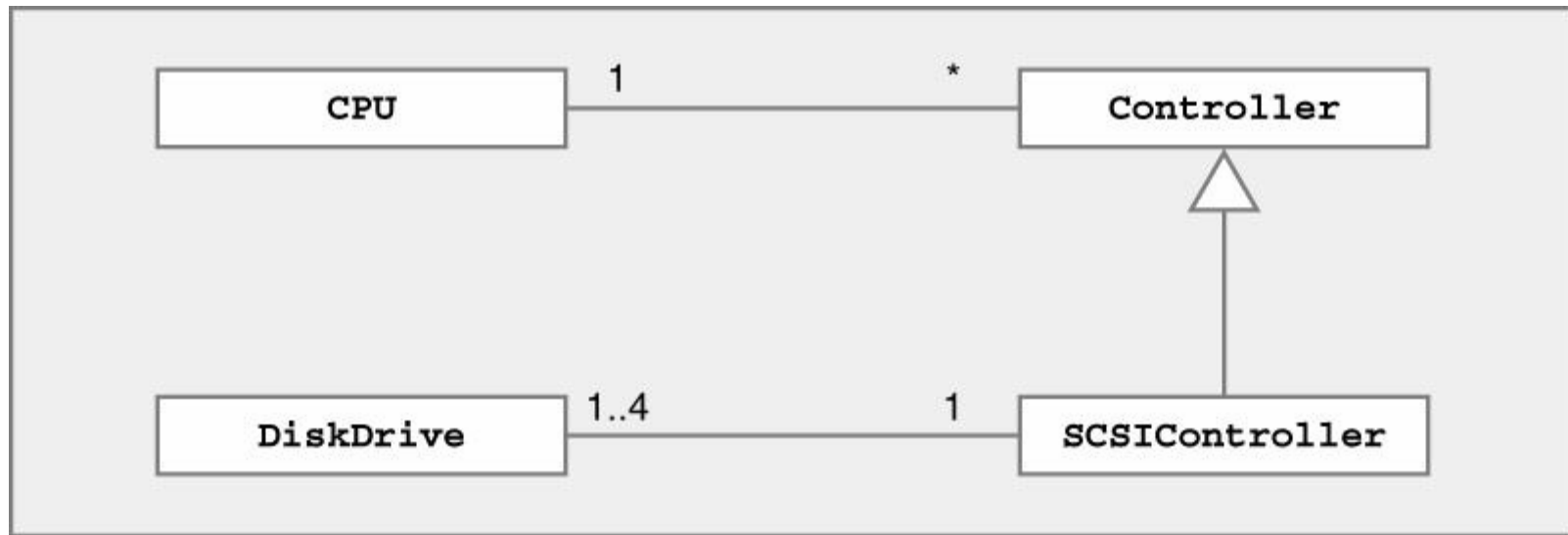


■ Java

**class Controller {
}
class SCSIController extends Controller {
}**

UML → Java : Computer System

■ UML



■ Java

- Design code using all available information in UML...

UML → Java : Computer System

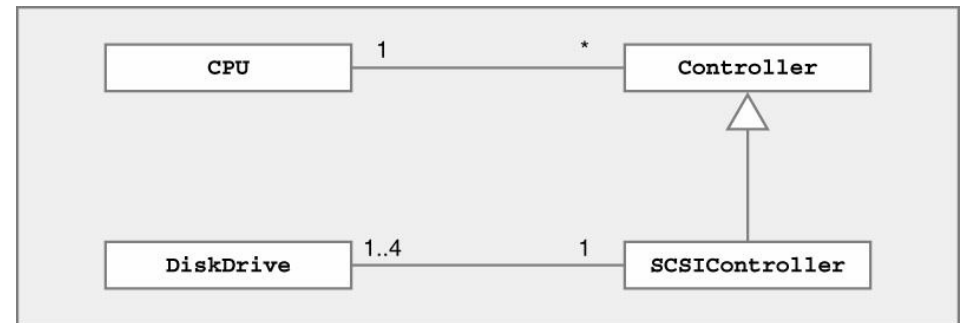
■ Java

```
class CPU {  
    Controller [ ] myCtrls;  
}
```

```
class Controller {  
    CPU myCPU;  
}
```

```
class SCSIController extends Controller {  
    DiskDrive [ ] myDrives = new DiskDrive[4];  
}
```

```
Class DiskDrive {  
    SCSIController mySCSI;  
}
```



UML Summary

- Graphics modeling language
- Visually represents design of software system
- We focused on **class diagrams**
 - Contents of a class
 - Relationship between classes
- You should be able to
 - Draw UML class diagram given Java code
 - Write Java code given UML class diagram