Lecture 8: More generics + subtyping

- List<? super T>
- Subtyping for generic types
- Copying collections, Comparable, Comparator

Announcements:

- No quiz this week
- Solutions
- Guest lectures
- Office hours this week

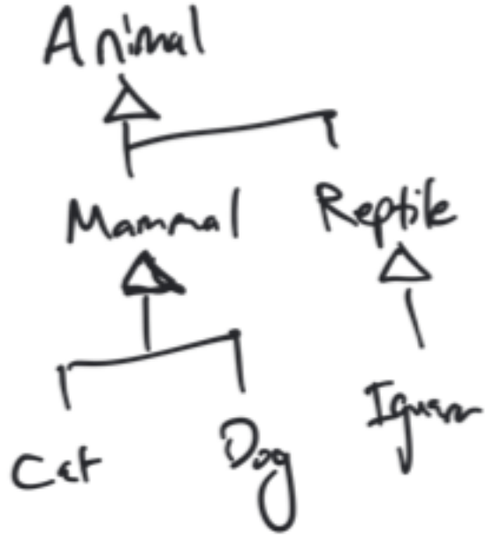
(supertype / interface)

Specification : interface MammalLover { void f(Mammal) }

⇒ implementation : void f(Animal) (almost) OK

inside class AnimalLover (not Java)

AnimalLover is DogLover



Specification : Mammal create Mammal()

⇒ implementation : Dog create Mammal()

DogBreeder is a MammalBreeder

```
Collection<E> {
    void add (E e);
    E get ();
}
```

Animal
↑
Mammal
↑
Dog..

Coll<Animal> Coll<Dog>

```
Collection<Animal> {
   add (Animal e);
   Animal get();
}
Collection<Dog> {
   add (Dog e);
   Dog get();
}
```

Java generates this from generic version

is a Collection<Animals> also a Collection<Dog>?

No:

```
Collection<Dog> cd;
cd.get().bark();
```

doesn't make sense if cd is a coll. of Animals.

Q: is Collection<Dog> a subtype of Collection<Animal>?

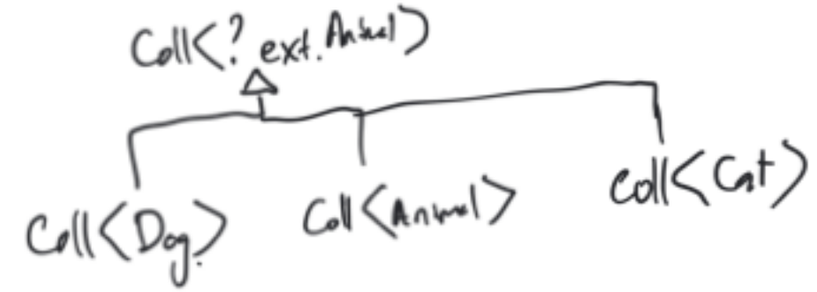
```
Coll<Animal> ca;        // Animal: Dog
ca.get().beCute();      // OK. (Animal)
ca.add(new Bird());     // not OK.
```

No!

```
void makeAllCute(Coll<? extends Animal> c){
    c.get().beCute();
    c.add(new Bird());
    c.add(new Animal());
}
```

```
Coll<Animal> ca;
Coll<Dog> cd;
makeCute(ca);
makeCute(cd);
```

(Animal is a subtype of Animal)




```
Coll<E>{
    void add(E e);
    E get();
}
```

could be Bird.

```
Coll<? extends Animal>{
    Animal get();
    void add();
}
```

Can't be Animal

c.add(new Dog())

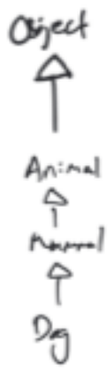
```
void makeDog (Coll<? super Dog> c) {
    c.add (new Dog());
    c.get() back!
}
```

```
Coll<Anim> ca; Coll<Dog> cd;
makeDog(ca);   makeDog(cd);
```

```
Coll<E> {
    void add (E e);
    E get();
}
```

```
Coll<Dog> {
    void add (Dog);
    Dog get();
}
```

```
Coll<Anim> {
    void add (Anim)
    Anim get()
}
```




```
Coll<? super Dog> {
    void add(Dog d);
    get()
}
```

Object get();

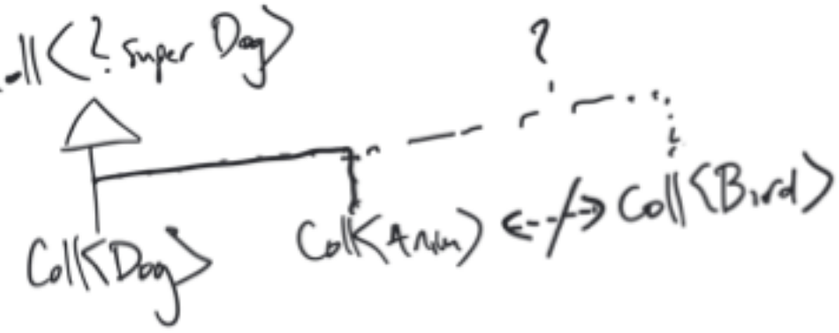
Dog ~~Mammal~~ Animal

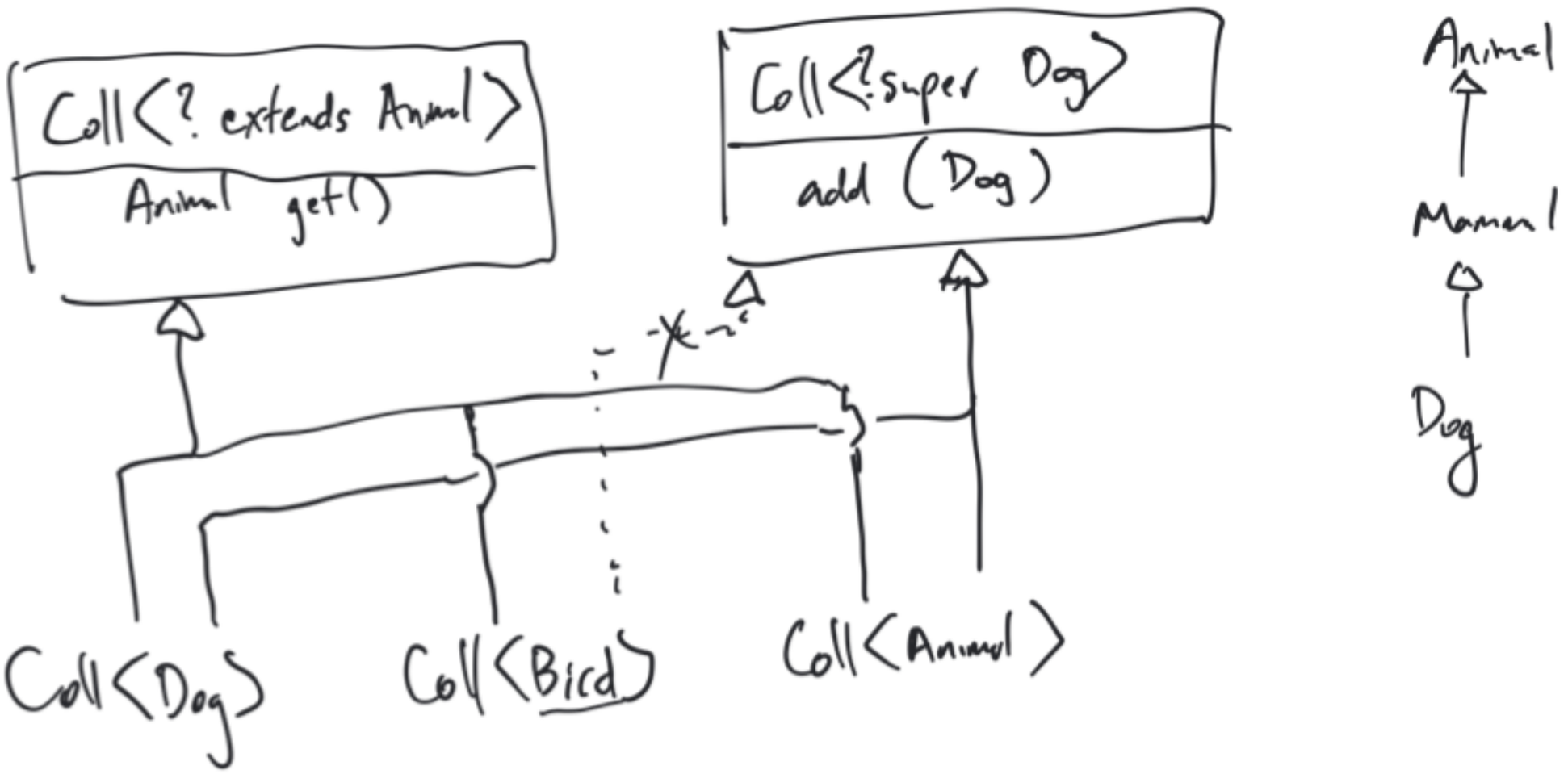
Should Coll<Dog> be a subtype of Coll<? super Dog> ?

Yes: everything Coll<? super Dog> can do Coll<Dog> can do better!

Should Coll<Animal> be a subtype of Coll<? super Dog> ?

Yes: everything Coll<? super Dog> can do, Coll<Anim> can't do better!




Coll<? extends Animal>
Animal get()
Coll<? super Dog>
add (Dog)
Coll<Dog>
Coll<Bird>
Coll<Animal>
Animal
Mamal
Dog