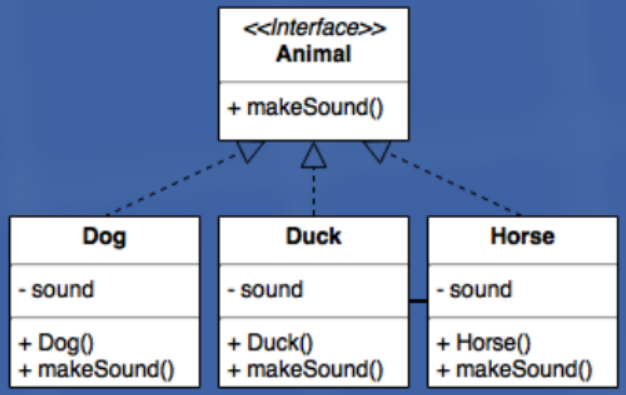
What does a dog say?



Client

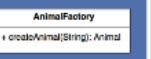
```
void clientFoo(String choice) {
  Animal animal;

  switch (choice) {
    case "Dog":
      animal = new Dog();
      break;
    case "Duck":
      animal = new Duck();
      break;
    case "Horse":
      animal = new Horse();
      break;
  }

  animal.makeSound();
}
```

Factory

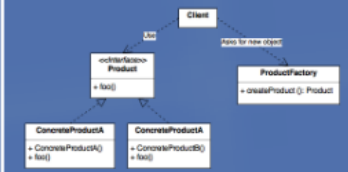
Responsible for *instantiating* concrete objects

Client refers to newly created object through common *interface*

Facilitates dependency inversion principle



Design skeleton

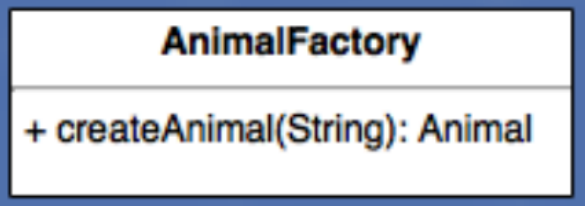


Factory

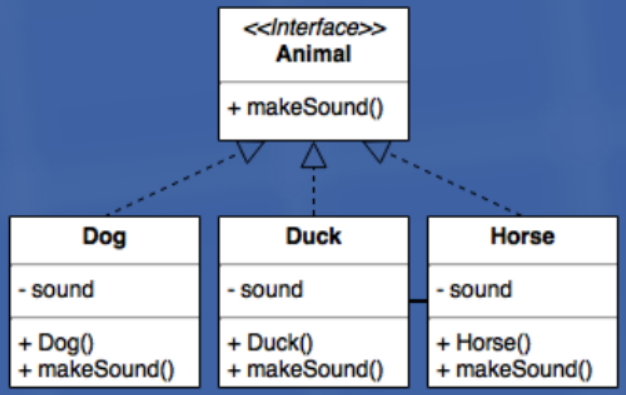
Responsible for *instantiating* concrete objects

Client refers to newly created object through common *interface*

Facilitates dependency inversion principle



What does a dog say?



Client

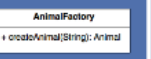
```
void clientFoo(String choice) {
  Animal animal;

  switch (choice) {
    case "Dog":
      animal = new Dog();
      break;
    case "Duck":
      animal = new Duck();
      break;
    case "Horse":
      animal = new Horse();
      break;
  }

  animal.makeSound();
}
```

Factory

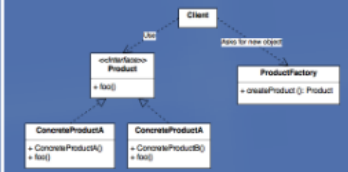
Responsible for *instantiating* concrete objects

Client refers to newly created object through common *interface*

Facilitates dependency inversion principle



Design skeleton



Design skeleton