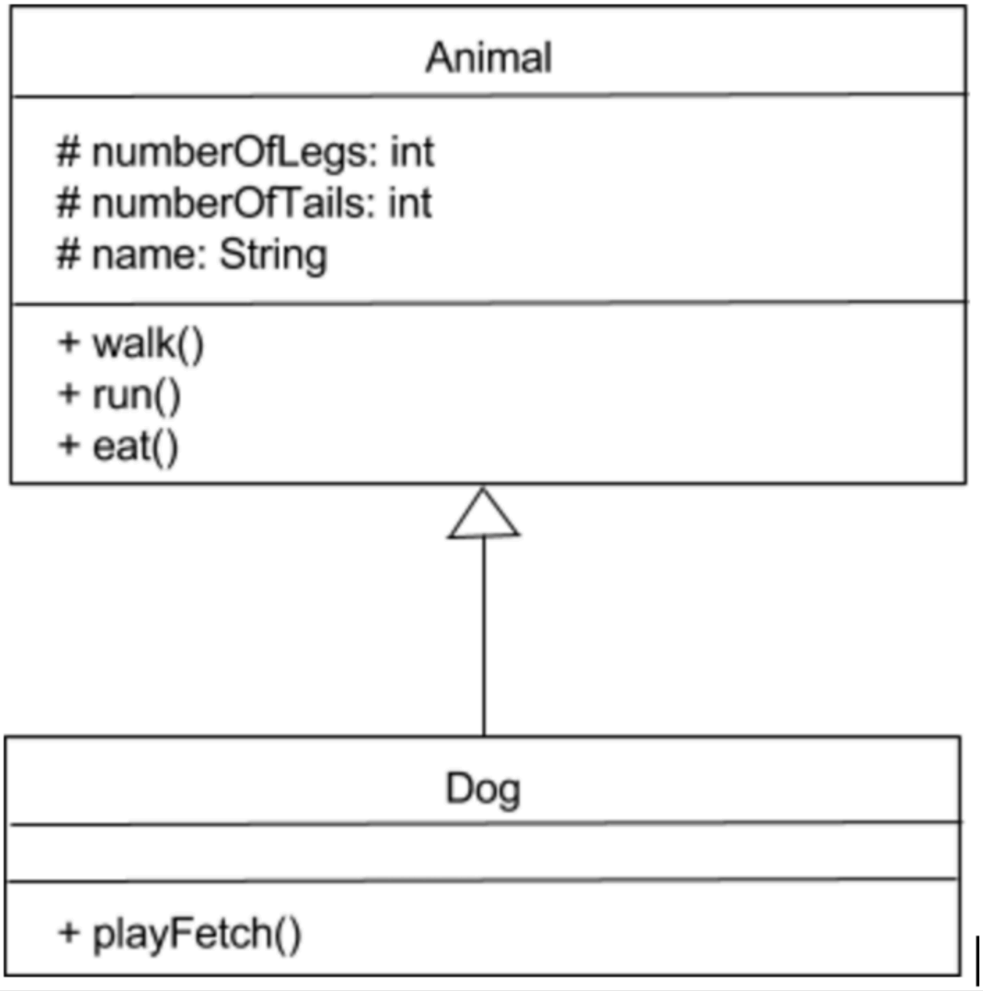


In our UML class diagram, which attributes and behaviors of the Animal class are inherited by the Dog class? Note: You can assume that the two classes are in the same package.

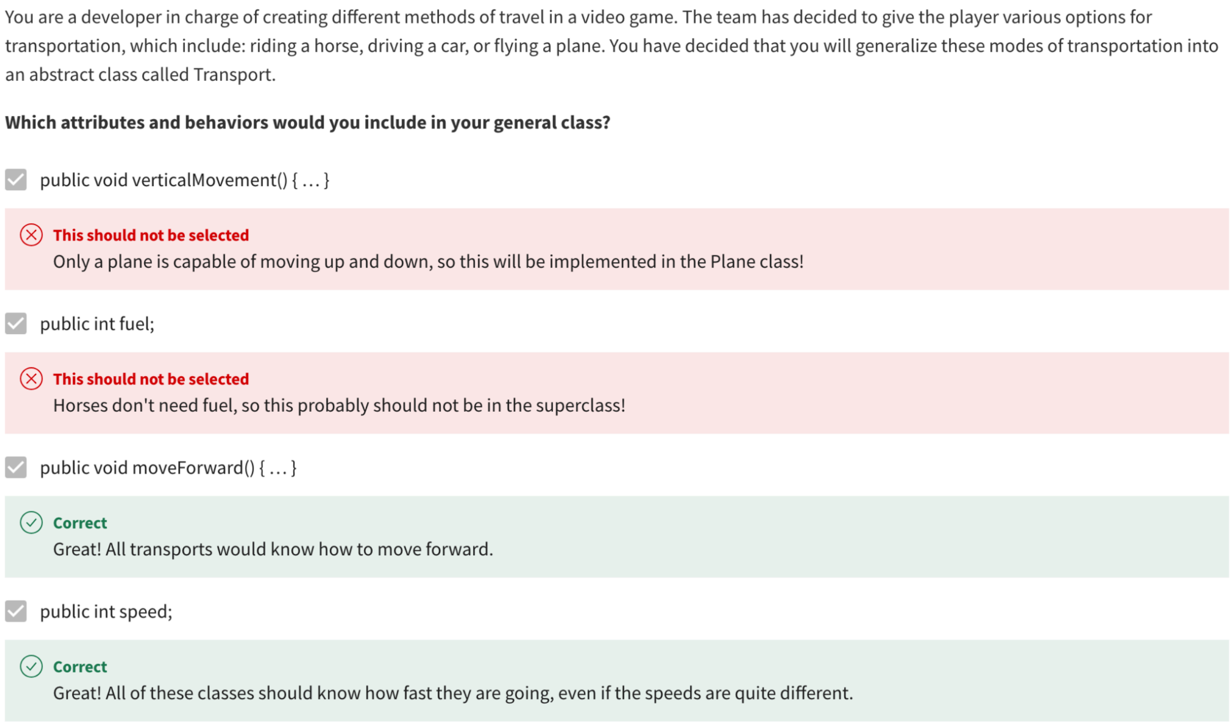
○ playFetch()

○ numberOfLegs, numberOfTails, walk(), run(), eat()

○ None of the attributes and behaviors are inherited.

● All attributes and behaviors of Animal are inherited.

✓ **Correct**
Great job! According to the UML diagram, Animal has many behaviours and attributes inherited by Dog.



You are a developer in charge of creating different methods of travel in a video game. The team has decided to give the player various options for transportation, which include: riding a horse, driving a car, or flying a plane. You have decided that you will generalize these modes of transportation into an abstract class called Transport.

Which attributes and behaviors would you include in your general class?

☑ public void verticalMovement() { … }

☒ **This should not be selected**
Only a plane is capable of moving up and down, so this will be implemented in the Plane class!

☑ public int fuel;

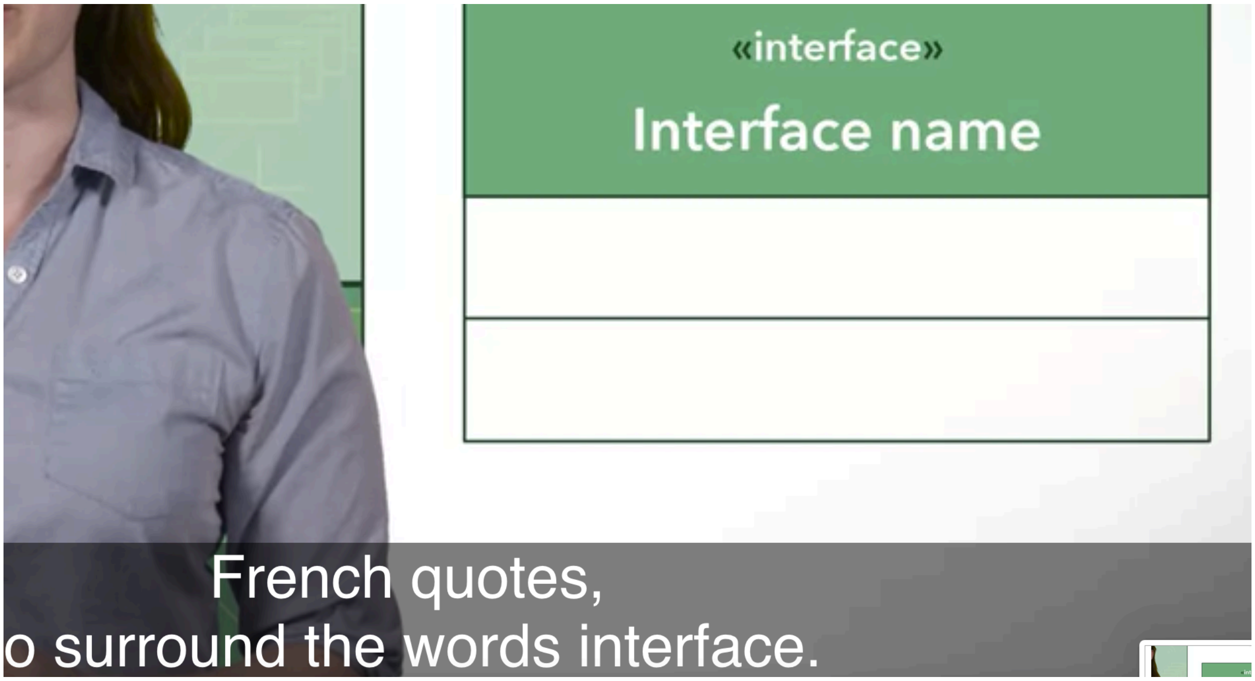
☒ **This should not be selected**
Horses don't need fuel, so this probably should not be in the superclass!

☑ public void moveForward() { … }

✓ **Correct**
Great! All transports would know how to move forward.

☑ public int speed;

✓ **Correct**
Great! All of these classes should know how fast they are going, even if the speeds are quite different.



Interfaces are drawn in a similar way that classes are drawn in UMLs. Interfaces are explicitly noted in UML class diagrams using **guillemets, or French quotes**, to surround the words interface.

The interaction between an interface and a class that is implementing the interface is indicated using a dotted arrow. The class touches the tail end of the arrow and the interface touches the head of the arrow.