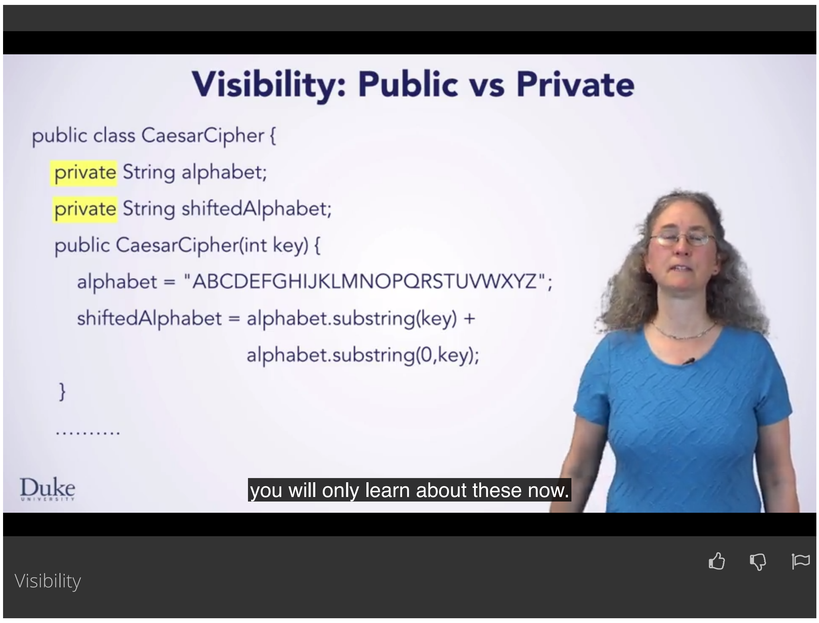

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript

English

0:03
>> Welcome back. In this lesson, you are going to learn about the two visibility modifiers that you saw in the object-oriented version of the CaesarCipher.

0:13
Public, which was used here, and private, which was used here. There are two other visibility modifiers, however. Explaining them requires some more advanced concepts, so you will only learn about these now.

0:27
When you declare something, whether it is a class, field, method, or constructor as public it means that any code, anywhere in your program can access that thing. Code from any class can call a public method, read or update a public field, use a public constructor when making an object, and make use of a public class.

0:48
By contrast, when you declare something private, it tells Java that only code inside of this particular class can see the thing you declare private. These two fields are declared private so all of the code inside of the CaesarCipher class can read and update them but code outside of the CaesarCipher class is not allowed to access them at all. So what happens if you write code that tries to access private fields or methods from outside the class?

1:17
The Java compiler will give you an error like this, which says that you are not allowed to do that. In general, when you get this sort of error, it either means that you are improperly using a class. Especially if it's a class that is part of an existing library. Or that you have designed a class and made something private that should not be.

1:37
So why do you want to declare things as private? After all, it seems like it would just be easier to make everything public. So you can just access whatever you want, wherever you want. Remember the idea of distraction? The principle that you want to separate the interface from the implementation. Restricting the visibility of the implementation details helps you enforce abstractions you design.

2:01
You can make the interface of a class, all the methods that other classes are supposed to call public, and then you can make the implementation details private.

2:13
No other class should know about the implementation details directly. And declaring them private lets you enforce that rule in your code. In our CaesarCypher example, you want other classes to be able to call encrypt. But they should not know about the implementation details, such as the fact that you made a variable called shifted alphabet.

2:34
Keeping these details private means that you can change them and be sure that no other code relies on the private implementation details.

2:43
As you start to think about designing your own classes, there are a few general pieces of guidance to get you started on how to choose public or private. First, fields are generally part of the implementation of an object. So they typically should be private.

3:00
For methods it depends on what the purpose of the method is. If the method is part of the interface you want your class to have, that is part of the behaviors you want it to provide to other pieces of code, you should declare that method public.

3:14
On the other hand, some methods are helpers. You array them to abstract out specific complex tasks which are not meant for other classes to call.

3:24
They just help accomplish the public interface. These methods should be private so that only the code in your class can call them. For classes, you should always declare them public. As you become more skilled in Java, you will learn some more advanced topics that lead to situations where you might want non-public classes. But for now, always use public.

3:47
Likewise, for constructors, you should always make these public for now. Typically, constructors are part of the public interface of a class. They specify how to make an instance. There are situations where non-public constructors are appropriate. But as with non-public classes, these only come up when you have learned some more advanced topics. So for now, you should just make constructors public.

4:14
Great. So now you know what public and private mean. Why they are useful and some general guidelines for how to use them in your own classes. Thank you.

Downloads

- **Lecture Video** mp4
- **Subtitles (English)** WebVTT
- **Transcript (English)** txt

Would you like to help us translate the transcript and subtitles into additional languages?