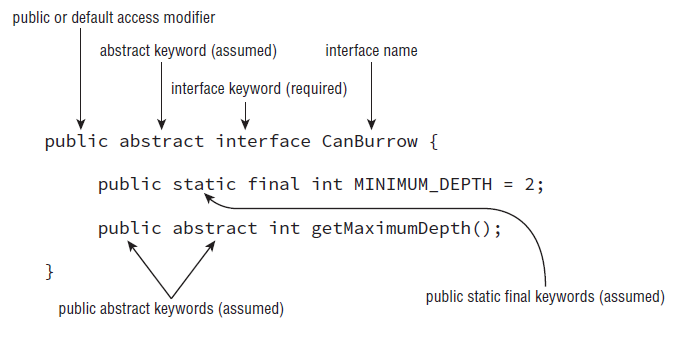
## Implementing Interfaces:



* The abstract keyword with interface definition is assumed. Whether you provide it or not.
* The method modifiers in this example, abstract and public, are assumed. In other words, whether or not you provide them, the compiler will automatically insert them as part of the method definition.
* The variables declared in an interface are assumed to me public, static, and final. Compiler will automatically insert **public static final** as part of the variable declaration.

## Defining an Interface:

The following is a list of rules for creating an interface, many of which you should recognize as adaptions of the rules for defining abstract classes.

1. Interfaces cannot be instantiated directly.
2. An interface is not required to have any methods.
3. An interface may not be marked as final.
4. All top-level interfaces are assumed to have public or default access, and they must include the abstract modifier in their definition. Therefore, marking an interface as private, protected, or final will trigger a compiler error, since this is incompatible with these assumptions.
5. All nondefault methods in an interface are assumed to have the modifiers abstract and public in their definition. Therefore, marking a method as private, protected, or final will trigger compiler errors as these are incompatible with the abstract and public keywords.

## Inheriting an Interface:

1. An interface that extends another interface, as well as an abstract class that implements an interface, inherits all of the abstract methods as its own abstract methods.
2. The first concrete class that implements an interface, or extends an abstract class that implements an interface, must provide an implementation for all of the inherited abstract methods.

* When a concrete class wants to inherit an interface it is do so by implementing it.
  + public class FieldMouse implements CanBurrow
* When an interface class wants to inherit an interface it is do so by extending it.
  + public interface Seal extends HasTail, HasWhiskers {}
* When an abstract class wants to inherit an interface it is do so by implementing it.
  + public class FieldMouse implements CanBurrow

## Multiple Inheritance: