LECTURE

Interface

Dr. Rubeena DOOMUN
Open University of Mauritius

INTERFACE

- A collection of abstract methods.
- A class implements an interface, thereby inheriting the abstract methods of the interface.
- Cannot instantiate an interface.
- An interface does not contain any constructors.
- An interface cannot contain instance fields. The only fields that can appear in an interface must be declared both static and final.
- An interface is not extended by a class; it is implemented by a class.

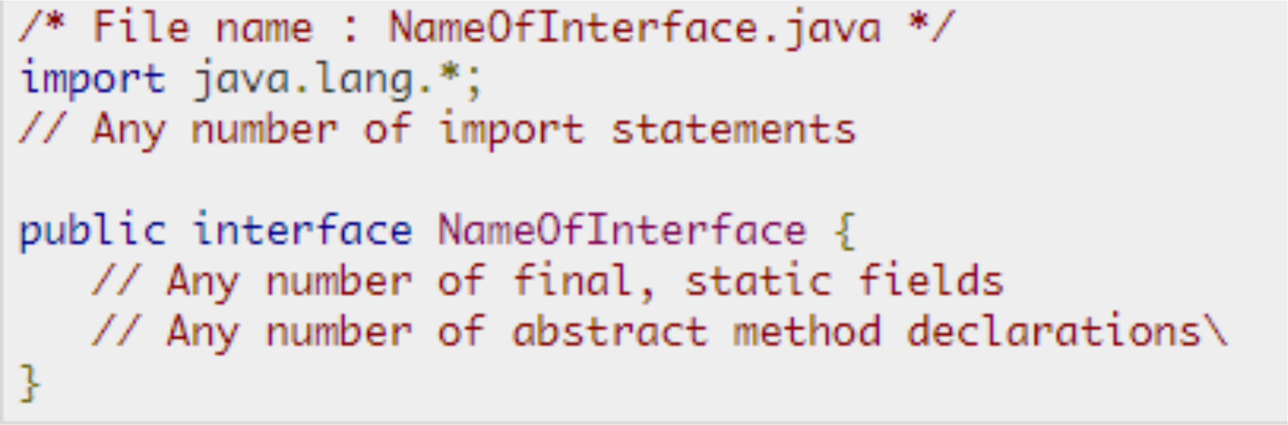
PROPERTIES

- Interfaces have the following properties :
- An interface is implicitly abstract. You do not need to use the **abstract** keyword while declaring an interface.
- Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.
- Methods in an interface are implicitly public.

Multiple inheritance

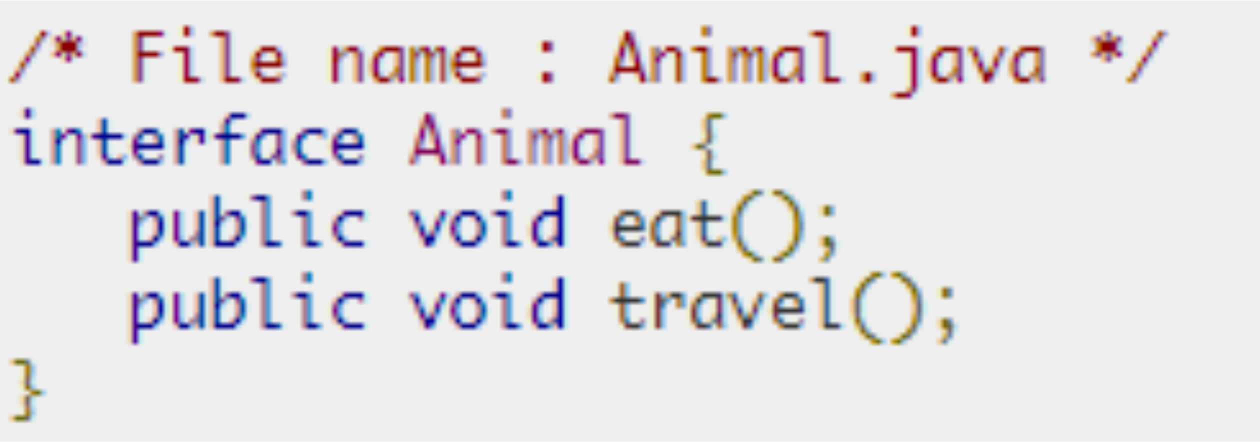
- As it was discussed in previous lecture, Multiple Inheritance is not allowed in Java.
- The use of interfaces can help achieve the concept of Multiple Inheritance.
- A class can extend only one class but can implement any number of interfaces.

Declaration



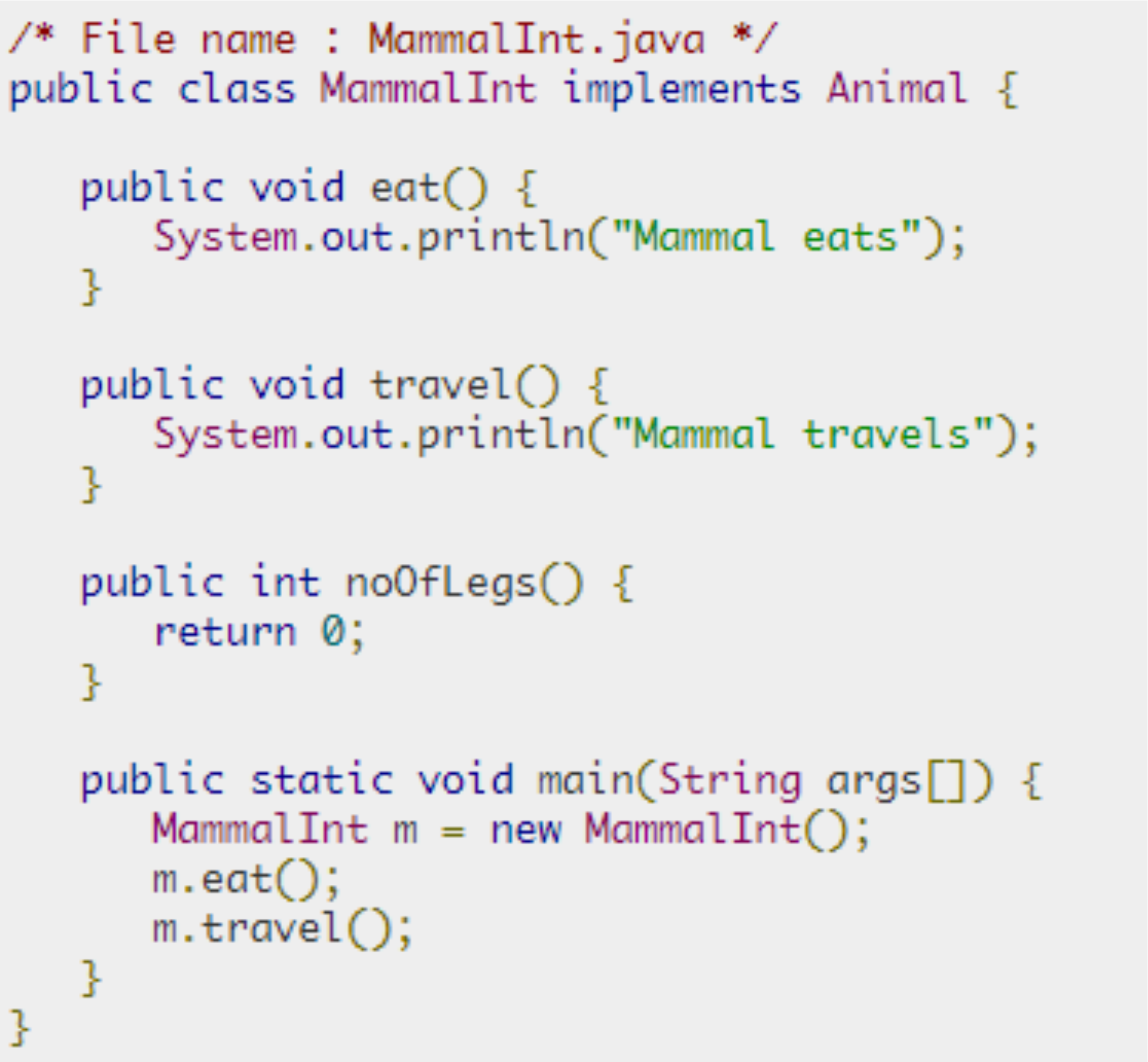
```
/* File name : NameOfInterface.java */
import java.lang.*;
// Any number of import statements

public interface NameOfInterface {
   // Any number of final, static fields
   // Any number of abstract method declarations\
}
```

Example



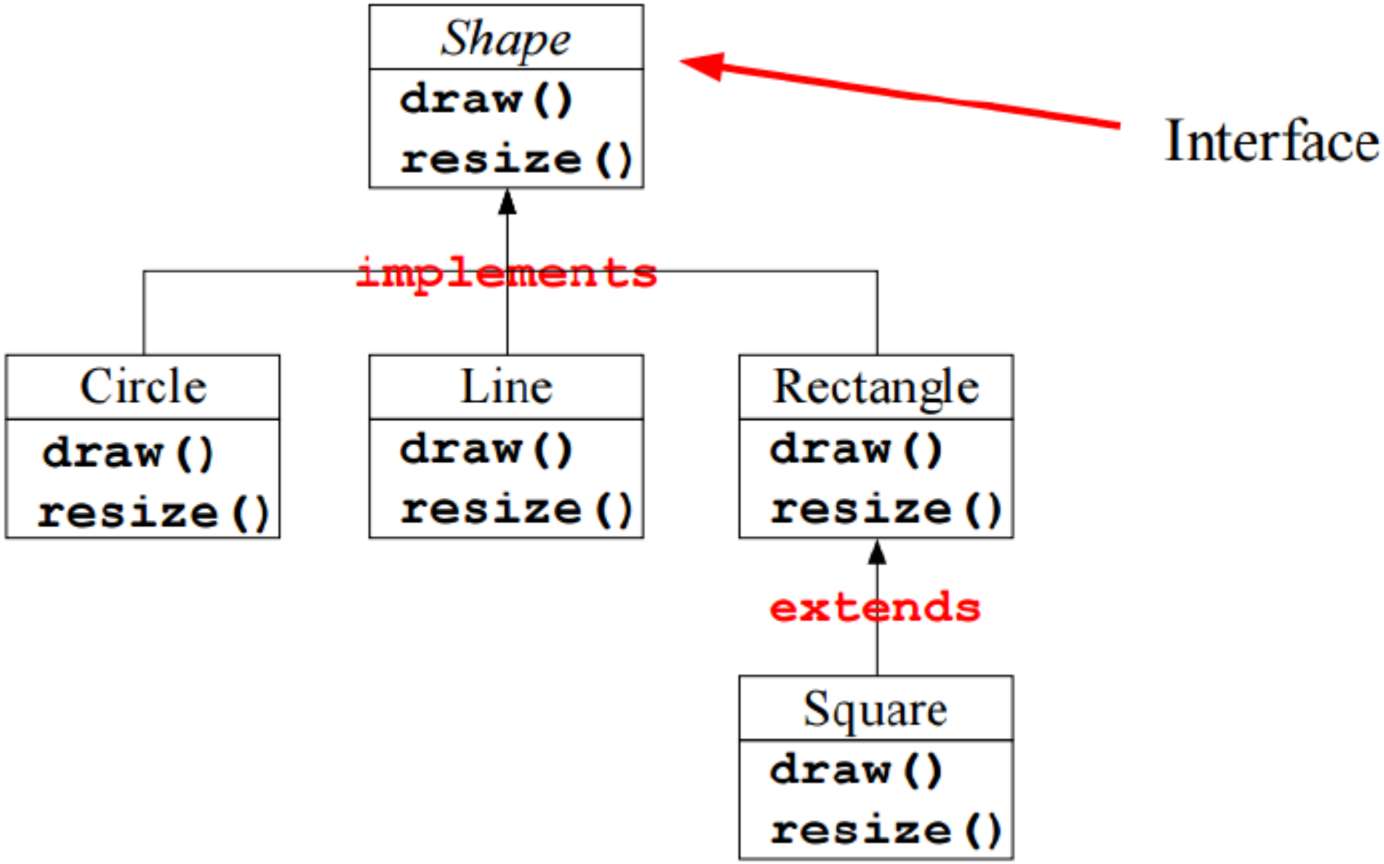
```
/* File name : Animal.java */
interface Animal {
   public void eat();
   public void travel();
}
```

Example



```
/* File name : MammalInt.java */
public class MammalInt implements Animal {

   public void eat() {
      System.out.println("Mammal eats");
   }

   public void travel() {
      System.out.println("Mammal travels");
   }

   public int noOfLegs() {
      return 0;
   }

   public static void main(String args[]) {
      MammalInt m = new MammalInt();
      m.eat();
      m.travel();
   }
}
```

Example

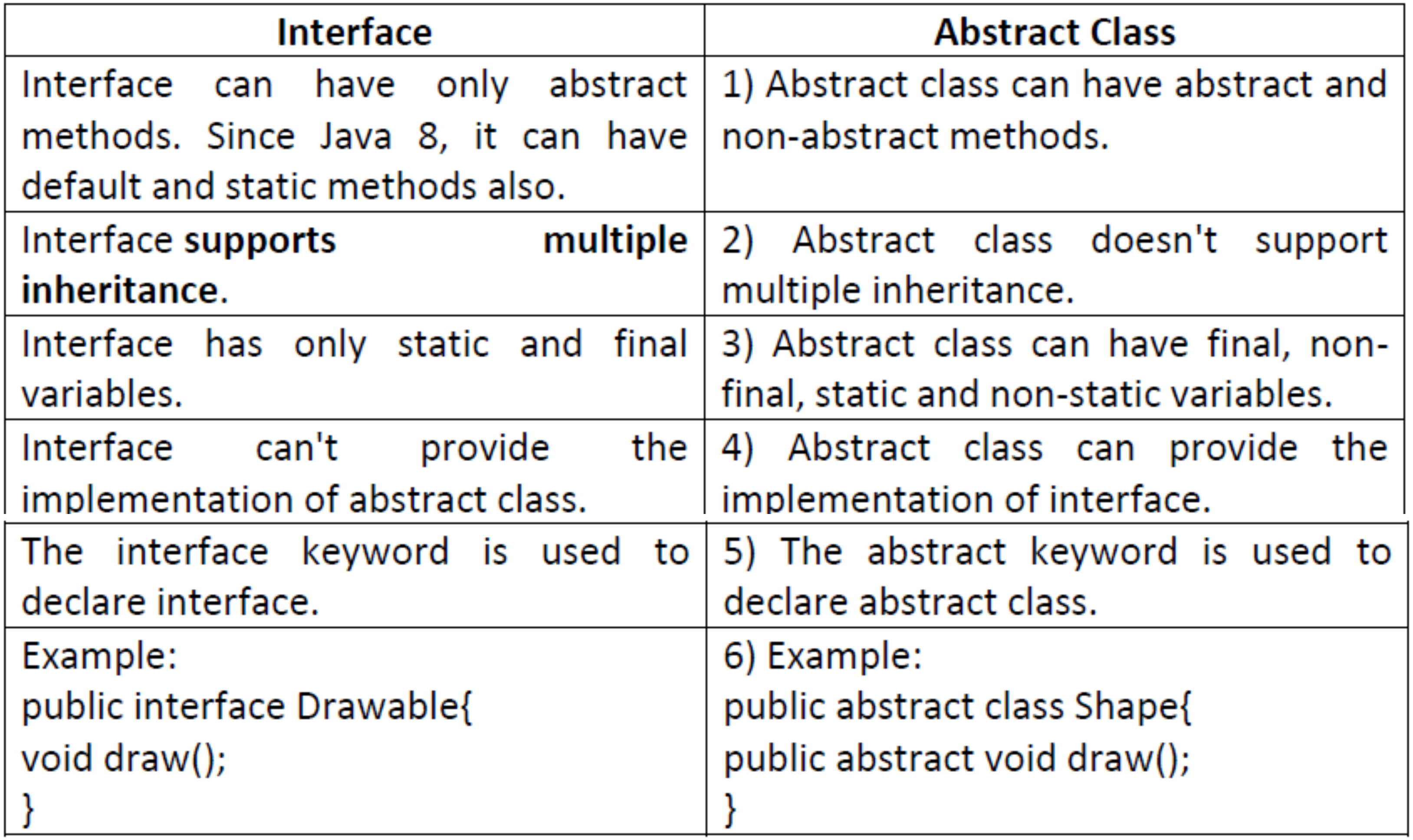


Shape
draw()
resize()
Interface
implements
Circle
draw()
resize()
Line
draw()
resize()
Rectangle
draw()
resize()
extends
Square
draw()
resize()

COMMONLY USED INTERFACES IN JAVA

- **Iterator**
  - loop
- **Cloneable**
  - copy
- **Serializable**
  - Store on file or persistent media
- **Comparable**
  - Comparison - returns a negative integer, zero, or a positive integer as this object is less than, equal to, or greater than the specified object.

INTERFACE V/S ABSTRACT CLASS



Interface	Abstract Class
Interface can have only abstract methods. Since Java 8, it can have default and static methods also.	1) Abstract class can have abstract and non-abstract methods.
Interface **supports multiple inheritance.**	2) Abstract class doesn't support multiple inheritance.
Interface has only static and final variables.	3) Abstract class can have final, non-final, static and non-static variables.
Interface can't provide the implementation of abstract class.	4) Abstract class can provide the implementation of interface.
The interface keyword is used to declare interface.	5) The abstract keyword is used to declare abstract class.

Example:
public interface Drawable{
void draw();
}
	6) Example:
	public abstract class Shape{
	public abstract void draw();
	}

REFERENCES

- https://www.tutorialspoint.com/java/java_interfaces.htm