Exercise 01:

Declare an interface called “MyFirstInterface”. Decalre integer type variable called “x”. Declare an abstract method called “display()”.

1. Try to declare the variable with/without public static final keywords. Is there any difference between these two approaches? Why?

No,

When declaring a variable within an interface it’s always public, static, and final.

There for it’s no difference between declaring a variable with or without public static final keywords.

1. Declare the abstract method with/without abstract keyword. Is there any difference between these two approaches? Why?

No,

Declaring abstract methods with or without abstract methods is not any difference, because methods in interface are by default in abstract.

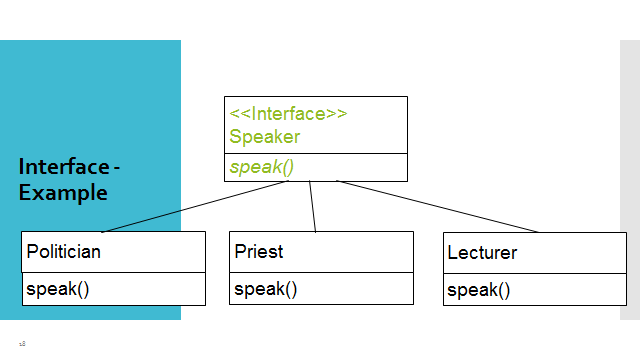
1. Implement this into a class called “IntefaceImplemented” . Override all the abstract methods. Try to change the value of x inside this method and print the value of x. Is it possible for you to change x? why?

No,

In interfaces, variables are always final. So, we can’t assign a value to the final variable.

Exercise 02:

Develop a code base for the following scenario. Recall what we have done at the lecture…



public class Lecturer implements Speaker {

@Override  
 public void speak(){  
  
 }

}

public class Priest implements Speaker {

@Override  
 public void speak(){  
  
 }

}

public class Politician implements Speaker {

@Override  
 public void speak(){  
  
 }

}

public interface Speaker {

void speak();

}

public class Main {

public static void main(String[] args) {

Speaker politician = new Politician();  
 politician.speak();  
  
 Speaker priest = new Priest();  
 priest.speak();  
  
 Speaker lecturer = new Lecturer();  
 lecturer.speak();  
  
 }

}