Exercise 01:

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

package com.mycompany.test02;

public interface myFirstInterface {

int x=5;

void display();

}

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

without public ,static, final:

package com.mycompany.test02;

public interface myFirstInterface {

int x=5;

void display();

}

with public,static,final:

package com.mycompany.test02;

public interface myFirstInterface {

public static final int x=5;

void display();

}

* both approaches will result in the same
* because all variables declared inside interface are implicitly public,static and final.
* then both approaches are no any different.

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

without abstract keyword:

package com.mycompany.test02;

public interface myFirstInterface {

int x=5;

void display();

}

with abstract keyword:

package com.mycompany.test02;

public interface myFirstInterface {

int x=5;

abstract void display();

}

* both approaches will result in the same.
* the method “display()” will be implicitly considered ‘abstract’ , meaning it has no method body and is meant to be implemented by classes that implement the interface.

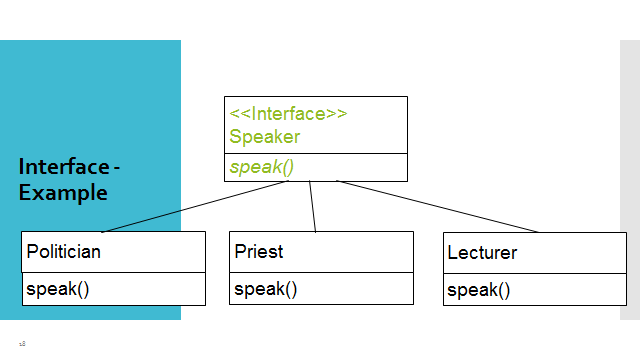
1. Implement this into a class called “IntefaceImplemented” . Override all the abstract methods. Try to change the value of x inside this method and print the value of x. Is it possible for you to change x? why?

cann’t change,

this will result in a compilation error because “x” is final variable in the interface, which means its value cann’t be modified after it’s assigned.

Exercise 02:

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



package com.mycompany.test02;

public interface Speaker {

void speak();

}

package com.mycompany.test02;

public class Lecturer implements Speaker {

public void speak()

{

System.out.println("i'm a lecturer");

}

}

package com.mycompany.test02;

public class Priest implements Speaker{

public void speak()

{

System.out.println("i'm a priest");

}

}

package com.mycompany.test02;

public class Politician implements Speaker{

public void speak()

{

System.out.println("i'm politician");

}

}