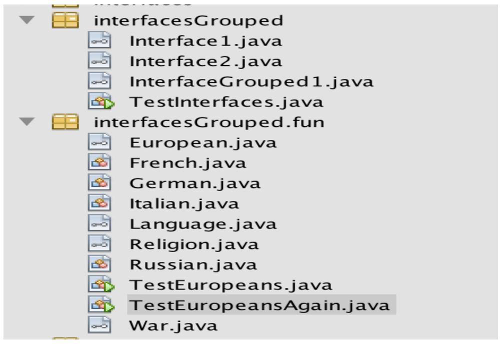
ADSV 2420, Advanced Programming I
Interfaces II

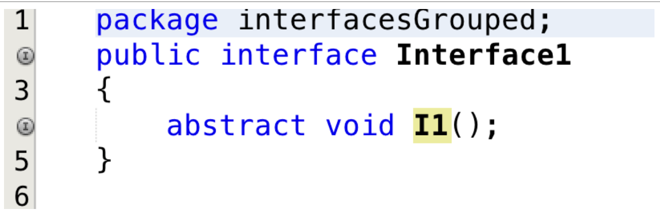
The final packages of all interfaces and classes of this lab:



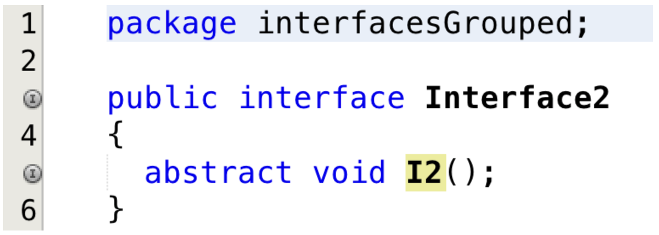
```
interfacesGrouped
    Interface1.java
    Interface2.java
    InterfaceGrouped1.java
    TestInterfaces.java
interfacesGrouped.fun
    European.java
    French.java
    German.java
    Italian.java
    Language.java
    Religion.java
    Russian.java
    TestEuropeans.java
    TestEuropeansAgain.java
    War.java
```

1. Add to your existing project (any) a new package called **interfacesGrouped.**

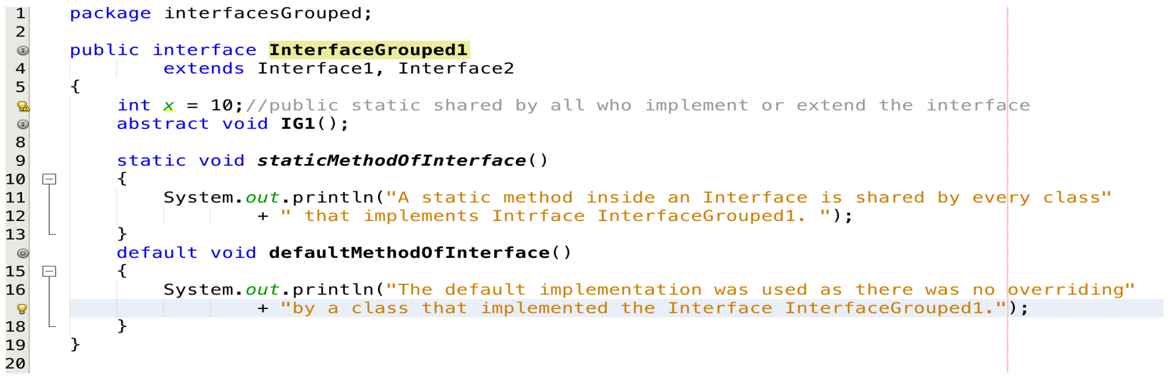
2. Under package ***interfacesGrouped*** right click and add the 3 interfaces shown below:
   Observe 1: We HAVE MULTIPLE INHERITANCE: The interface *InterfaceGrouped1 extends* interfaces *Interface1,* and *Interface2* , thus it inherits ALL from *Interface1,* and *Interface2* . The static method inside an interface *InterfaceGrouped1* will be shared by all classes that implement the intrerface *InterfaceGrouped1.*
   Observe 2: *InterfaceGrouped1* contains the method *defaultMethodOfInteface* with the keyword **default** in front of it. A default method is a method of an interface that **could** be overridden by the class that implements the interface.



```java
package interfacesGrouped;
public interface Interface1
{
    abstract void I1();
}
```

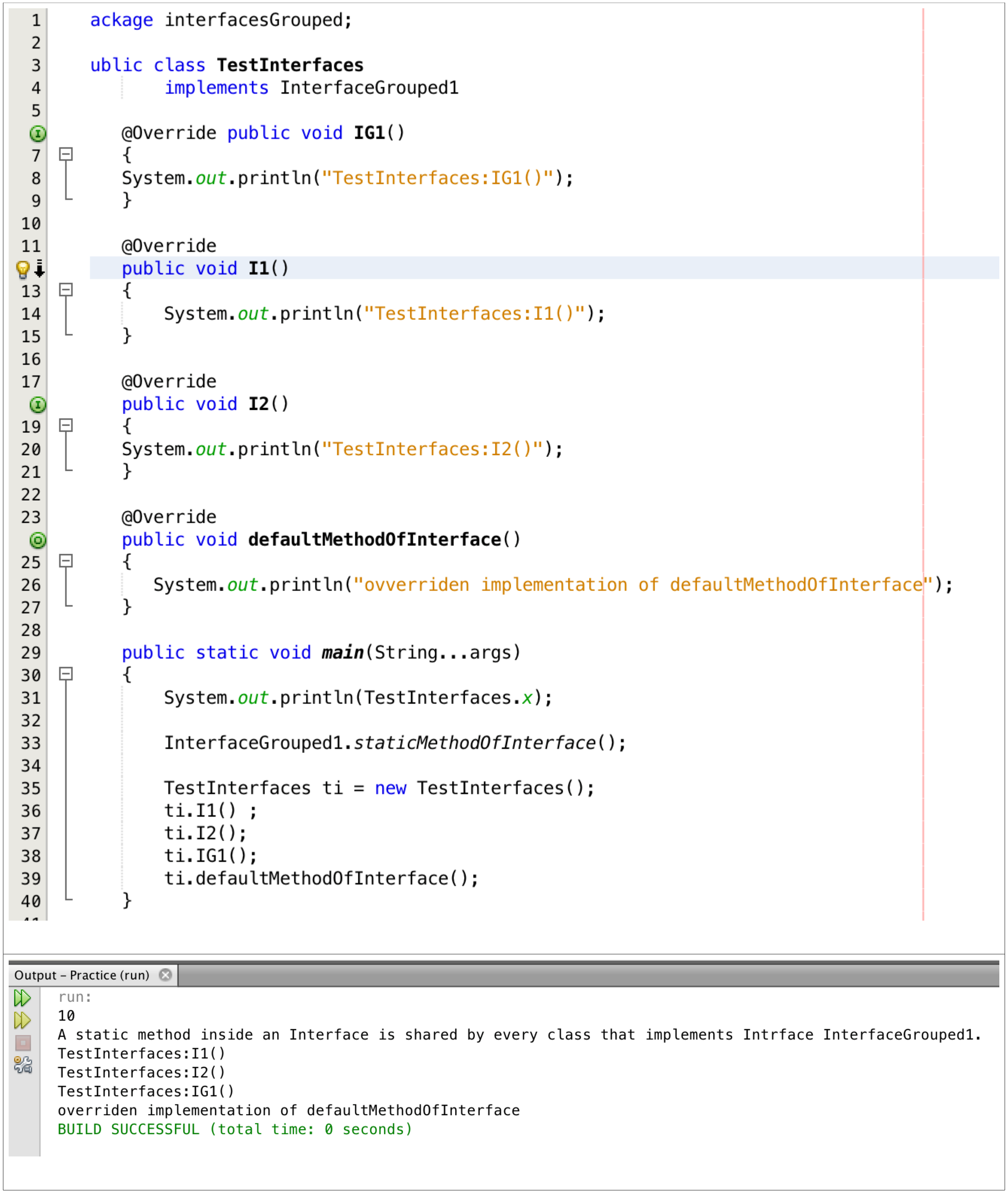


```java
package interfacesGrouped;

public interface Interface2
{
   abstract void I2();
}
```



```java
package interfacesGrouped;

public interface InterfaceGrouped1
        extends Interface1, Interface2
{
    int x = 10;//public static shared by all who implement or extend the interface
    abstract void IG1();

    static void staticMethodOfInterface()
    {
        System.out.println("A static method inside an Interface is shared by every class"
                + " that implements Intrface InterfaceGrouped1. ");
    }
    default void defaultMethodOfInterface()
    {
        System.out.println("The default implementation was used as there was no overriding"
                + "by a class that implemented the Interface InterfaceGrouped1.");
    }
}
```

3. Under the package ***interfacesGrouped*** create the class ***<u>TestIntrafaces</u>*** as shown below: Go line by line from 31 to 39 of main() below, and understand what's going on. Ask your instructor for anything that may puzzle you.



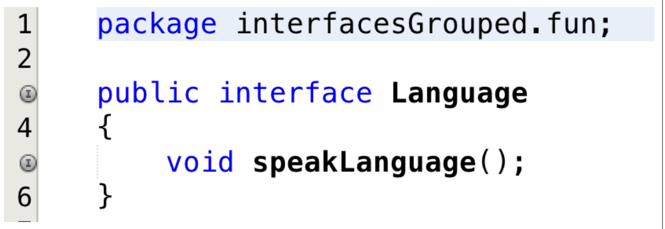
```
 1  ackage interfacesGrouped;
 2  
 3  ublic class TestInterfaces
 4         implements InterfaceGrouped1
 5  
    @Override public void IG1()
 7     {
 8     System.out.println("TestInterfaces:IG1()");
 9     }
10  
11     @Override
       public void I1()
13     {
14         System.out.println("TestInterfaces:I1()");
15     }
16  
17     @Override
       public void I2()
19     {
20     System.out.println("TestInterfaces:I2()");
21     }
22  
23     @Override
       public void defaultMethodOfInterface()
25     {
26        System.out.println("ovverriden implementation of defaultMethodOfInterface");
27     }
28  
29     public static void main(String...args)
30     {
31         System.out.println(TestInterfaces.x);
32  
33         InterfaceGrouped1.staticMethodOfInterface();
34  
35         TestInterfaces ti = new TestInterfaces();
36         ti.I1() ;
37         ti.I2();
38         ti.IG1();
39         ti.defaultMethodOfInterface();
40     }
```

Output - Practice (run)

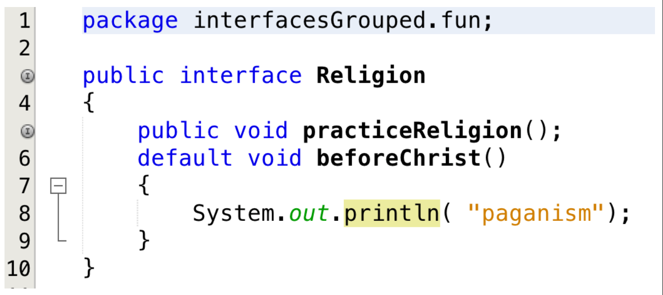
```
run:
10
A static method inside an Interface is shared by every class that implements Intrface InterfaceGrouped1.
TestInterfaces:I1()
TestInterfaces:I2()
TestInterfaces:IG1()
overriden implementation of defaultMethodOfInterface
BUILD SUCCESSFUL (total time: 0 seconds)
```

NEW PROBLEM

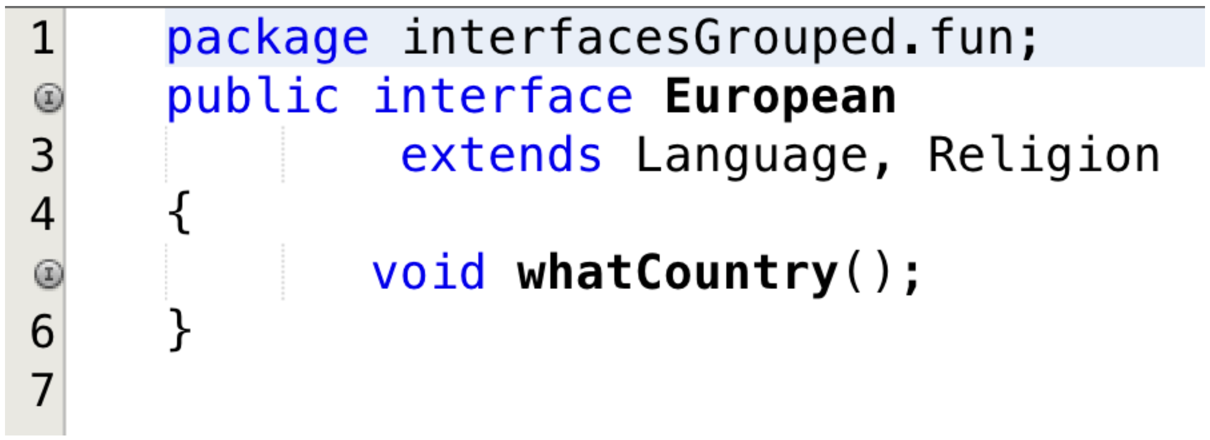
1. **Right click** on package **intrfacesGrouped** and ADD a new package named **fun**. Add to package fun the interfaces shown below. Observe that interface *European* inherits method *speakLanguage* from interface *Language* and methods *practiceReligion* and method *beforeChrist* from interface *Religion*. In addition is has its own abstract method *whatCoutry*.



```java
package interfacesGrouped.fun;

public interface Language
{
    void speakLanguage();
}
```

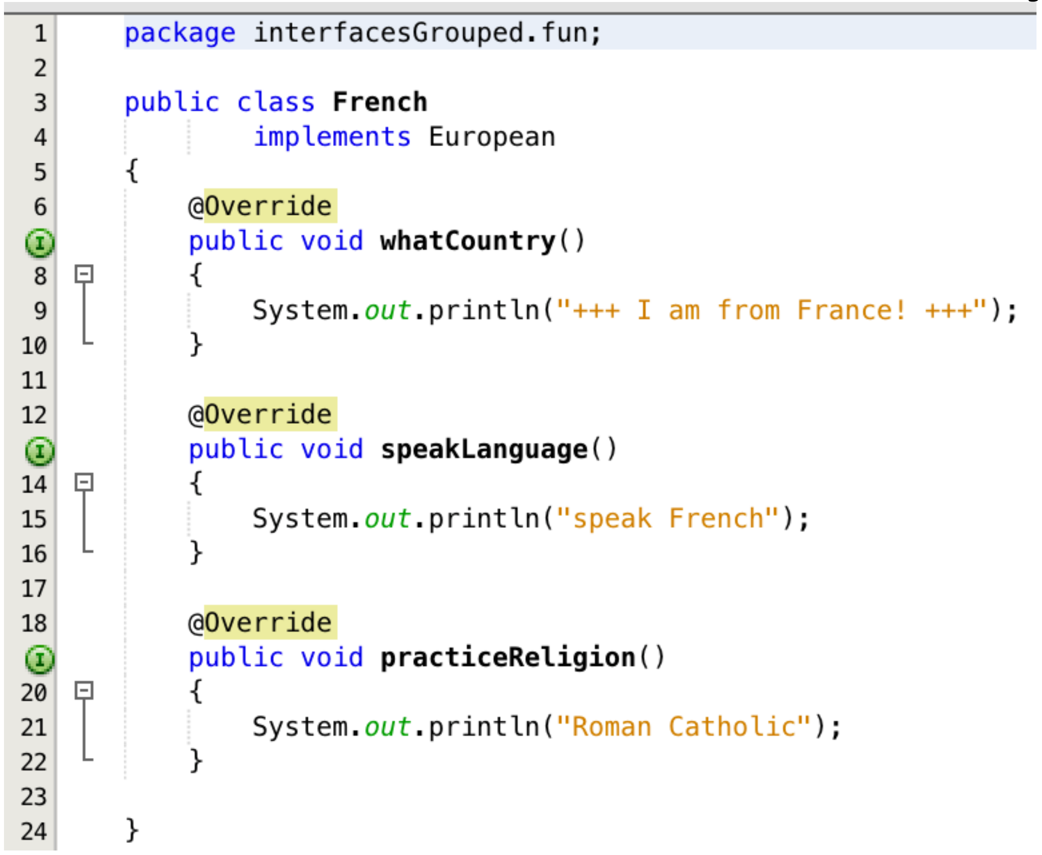


```java
package interfacesGrouped.fun;

public interface Religion
{
    public void practiceReligion();
    default void beforeChrist()
    {
        System.out.println( "paganism");
    }
}
```



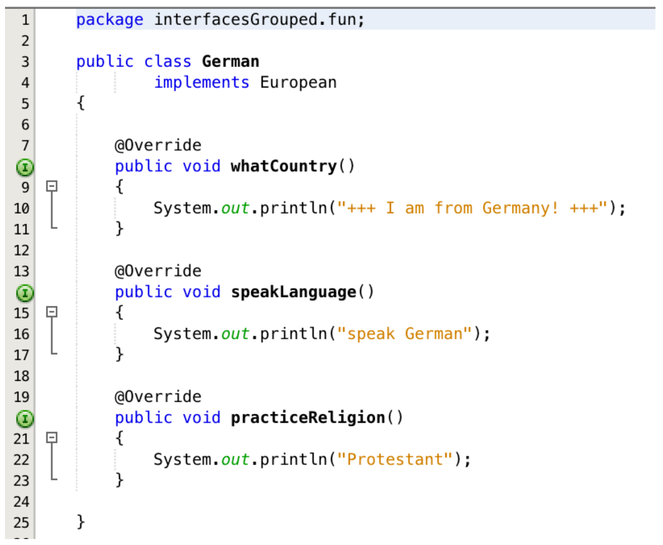
```java
package interfacesGrouped.fun;
public interface European
        extends Language, Religion
{
        void whatCountry();
}
```

2. Now add the class *French* which implements *European*. **Use the lightbulb to implement all abstract methods** as shown below. DO NOT TYPE THE METHOD HEADERS, use the lightbulb.

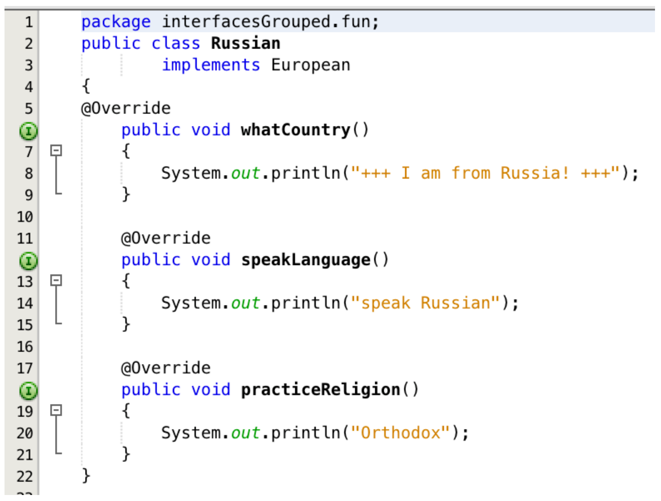


```java
package interfacesGrouped.fun;

public class French
        implements European
{
    @Override
    public void whatCountry()
    {
        System.out.println("+++ I am from France! +++");
    }

    @Override
    public void speakLanguage()
    {
        System.out.println("speak French");
    }

    @Override
    public void practiceReligion()
    {
        System.out.println("Roman Catholic");
    }

}
```

3. Similarly add the classes *German* and *Russian* which implement *European* as shown below; **Observe** that the overridden methods for classes *Russian*, *German* and *French* have different implementations. For example, the overridden method *practiceReligion* in class *Russian* prints Orthodox while the overridden method *practiceReligion* in class *French* prints Roman Catholic and in class *German* prints Protestant.

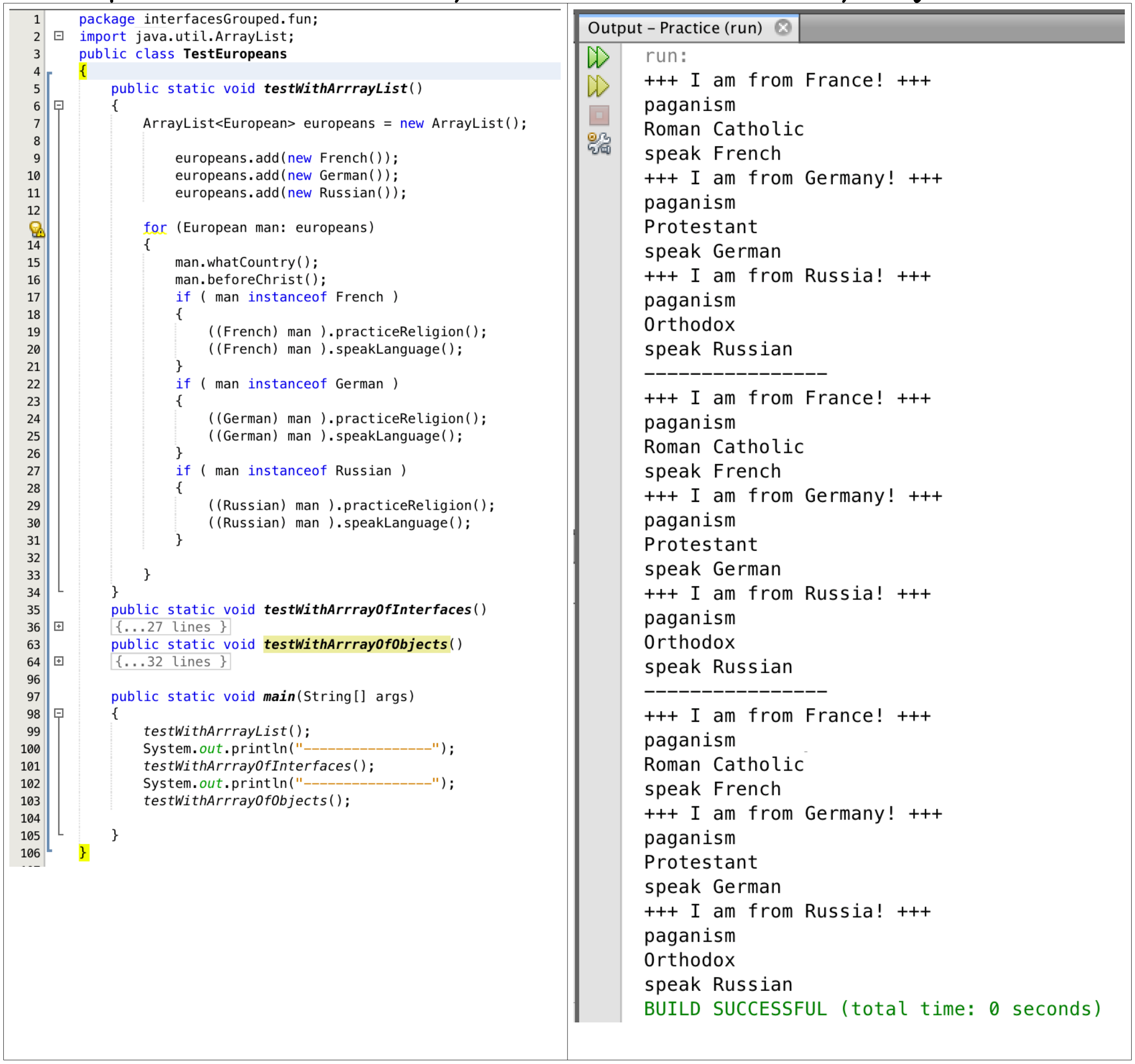


```java
package interfacesGrouped.fun;

public class German
        implements European
{

    @Override
    public void whatCountry()
    {
        System.out.println("+++ I am from Germany! +++");
    }

    @Override
    public void speakLanguage()
    {
        System.out.println("speak German");
    }

    @Override
    public void practiceReligion()
    {
        System.out.println("Protestant");
    }

}
```



```java
package interfacesGrouped.fun;
public class Russian
        implements European
{
@Override
    public void whatCountry()
    {
        System.out.println("+++ I am from Russia! +++");
    }

    @Override
    public void speakLanguage()
    {
        System.out.println("speak Russian");
    }

    @Override
    public void practiceReligion()
    {
        System.out.println("Orthodox");
    }
}
```

4. Now test the classes *French*, *German* and *Russian* by implementing methods *testWithArrrayOfInterfaces*() and *testWithArrrayOfObjects*() which are almost identical to method *testWithArrrayList* with its code given below. The method *testWithArrrayOfInterfaces* uses an array of interfaces of type *European* ,and the method *testWithArrrayOfObjects* uses an array of type *Object*. **You should produce the exact output shown after you provide your impementation for *testWithArrrayOfInterfaces and testWithArrrayOfObjects*.**



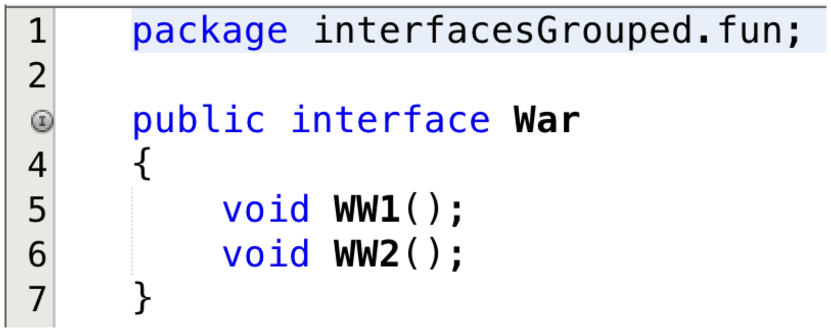
```java
package interfacesGrouped.fun;
import java.util.ArrayList;
public class TestEuropeans
{
    public static void testWithArrrayList()
    {
        ArrayList<European> europeans = new ArrayList();

            europeans.add(new French());
            europeans.add(new German());
            europeans.add(new Russian());

        for (European man: europeans)
        {
            man.whatCountry();
            man.beforeChrist();
            if ( man instanceof French )
            {
                ((French) man ).practiceReligion();
                ((French) man ).speakLanguage();
            }
            if ( man instanceof German )
            {
                ((German) man ).practiceReligion();
                ((German) man ).speakLanguage();
            }
            if ( man instanceof Russian )
            {
                ((Russian) man ).practiceReligion();
                ((Russian) man ).speakLanguage();
            }

        }
    }
    public static void testWithArrrayOfInterfaces()
    {...27 lines }
    public static void testWithArrrayOfObjects()
    {...32 lines }

    public static void main(String[] args)
    {
        testWithArrrayList();
        System.out.println("----------------");
        testWithArrrayOfInterfaces();
        System.out.println("----------------");
        testWithArrrayOfObjects();

    }
}
```

Output – Practice (run)

```
run:
+++ I am from France! +++
paganism
Roman Catholic
speak French
+++ I am from Germany! +++
paganism
Protestant
speak German
+++ I am from Russia! +++
paganism
Orthodox
speak Russian
----------------
+++ I am from France! +++
paganism
Roman Catholic
speak French
+++ I am from Germany! +++
paganism
Protestant
speak German
+++ I am from Russia! +++
paganism
Orthodox
speak Russian
----------------
+++ I am from France! +++
paganism
Roman Catholic
speak French
+++ I am from Germany! +++
paganism
Protestant
speak German
+++ I am from Russia! +++
paganism
Orthodox
speak Russian
BUILD SUCCESSFUL (total time: 0 seconds)
```

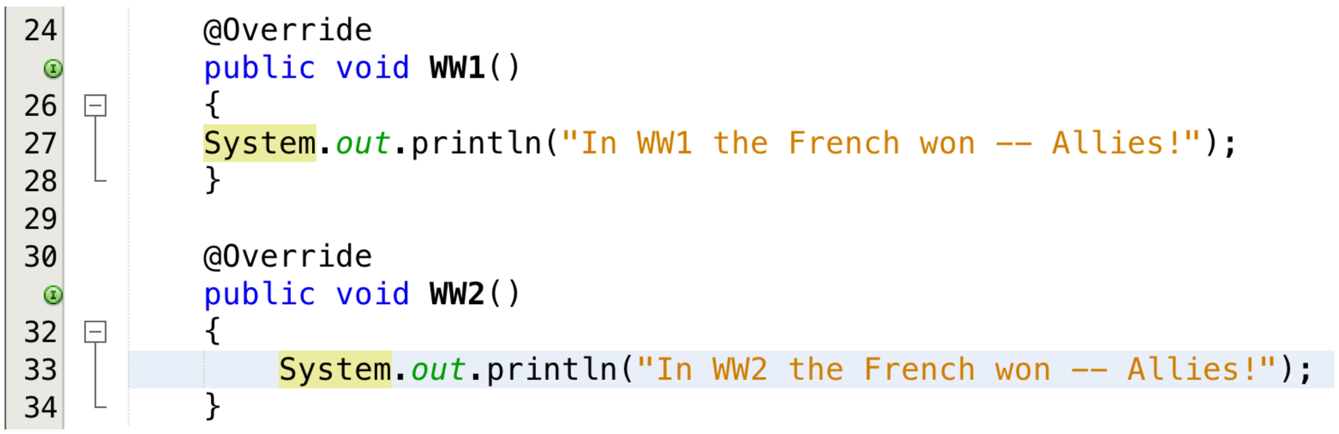
New Problem

1. Under the package **fun** add the interface *War* as shown below with the 2 abstract methods *WW1* and *WW2.*



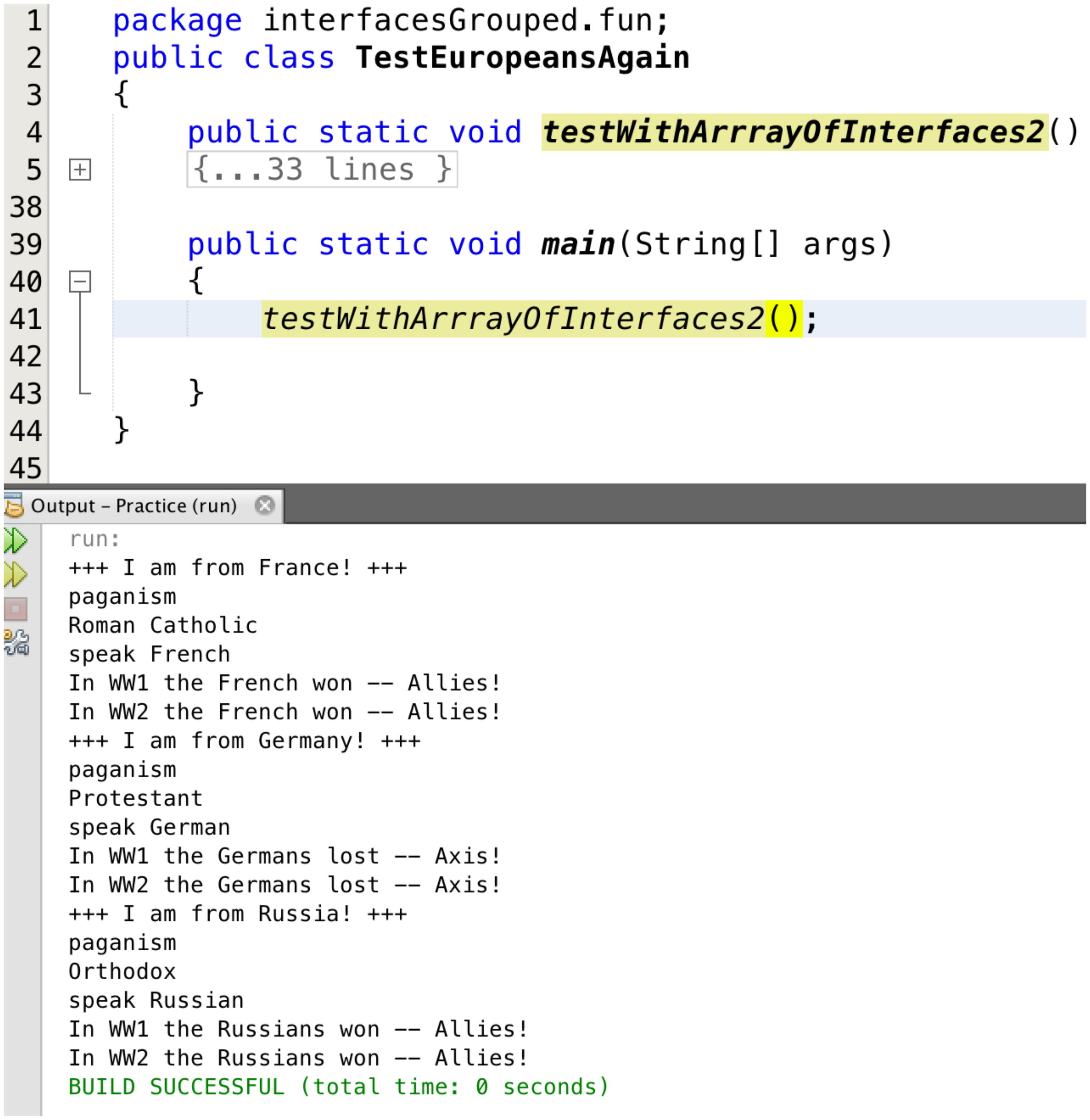
```java
package interfacesGrouped.fun;

public interface War
{
    void WW1();
    void WW2();
}
```

2. **In the extend** of interface *European* add the *War* and implement all abstract methods using the lightbulb.
Now the classes French, German and Russian won't compile as the interface *European* has War as one of its **extend**ed interfaces. Modify the classes *French*, German and *Russian* to reflect the changes of interface *European.* The changes for class French are shown below. Similarly change the classes German and *Russian.* For the shake of completeness, the Germans were with the Axis and lost both wars and the Russians were with the Allies and won both wars.



```java
@Override
public void WW1()
{
System.out.println("In WW1 the French won -- Allies!");
}

@Override
public void WW2()
{
    System.out.println("In WW2 the French won -- Allies!");
}
```

3. Under the package **fun** create the class *TestEuropeansAgain* and implement its method *testWithArrrayofInterfaces2* shown below, to produce the output shown below. Hint: Use an array of type *European* and add the 2 class for WW1 and WW2 where it is appropriate.



```java
package interfacesGrouped.fun;
public class TestEuropeansAgain
{
    public static void testWithArrrayOfInterfaces2()
    {...33 lines }

    public static void main(String[] args)
    {
        testWithArrrayOfInterfaces2();

    }
}
```

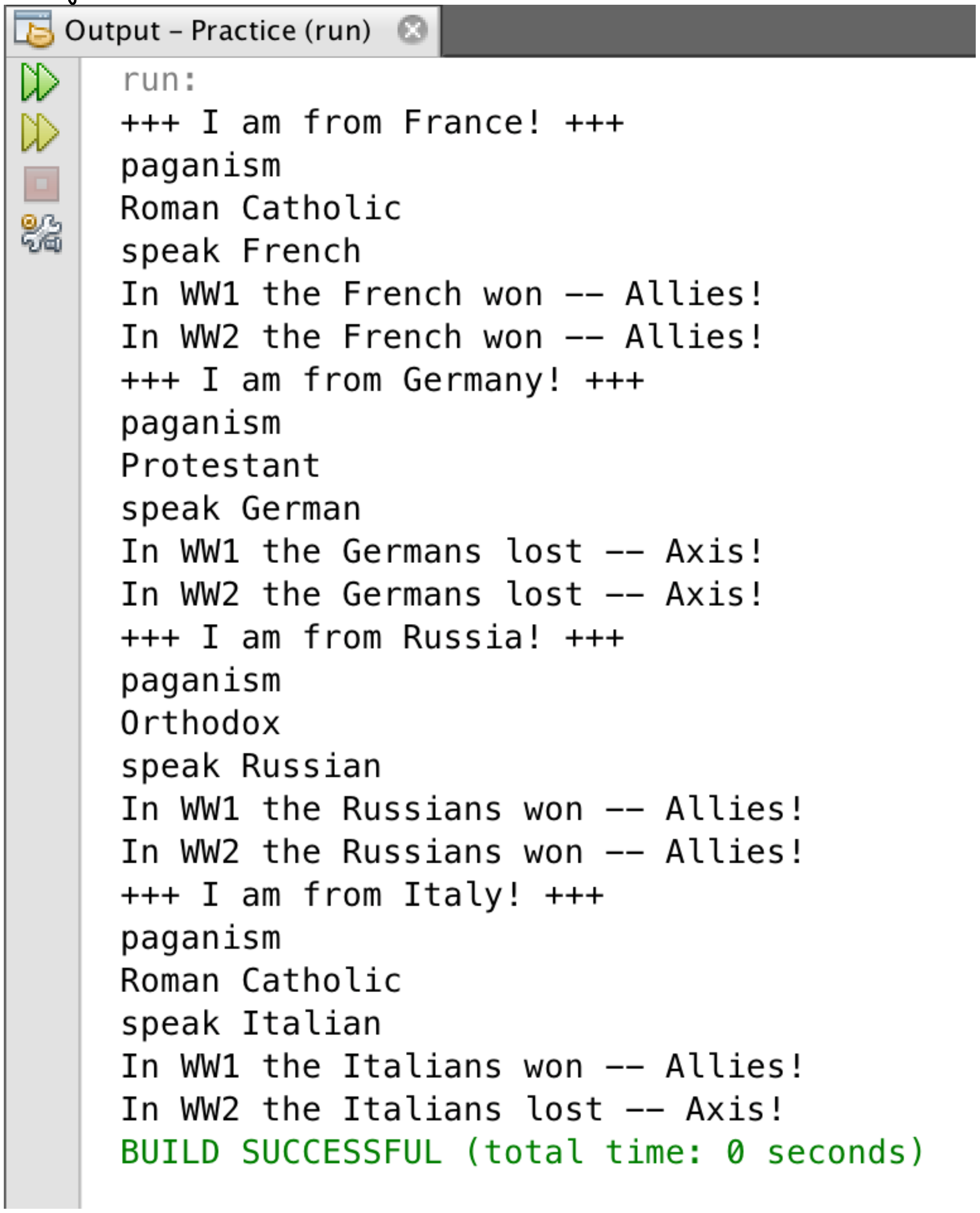
Output – Practice (run)

```
run:
+++ I am from France! +++
paganism
Roman Catholic
speak French
In WW1 the French won -- Allies!
In WW2 the French won -- Allies!
+++ I am from Germany! +++
paganism
Protestant
speak German
In WW1 the Germans lost -- Axis!
In WW2 the Germans lost -- Axis!
+++ I am from Russia! +++
paganism
Orthodox
speak Russian
In WW1 the Russians won -- Allies!
In WW2 the Russians won -- Allies!
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Under the package **fun** create the class *Italian* which **extends** European. Click the light bulb to implement all abstract methods. The Italians speak Italian, are Roman Catholic and were with the winning side in WW1 and the losing side in WW2.

5. Modify your class *TestWithEuropeansAgain* and add one more object to you array of type *Italian.*

6. Modify your method *testWithArrrayofInterfaces2 to* produce the following output which displays the Italian object last as shown below.



```
run:
+++ I am from France! +++
paganism
Roman Catholic
speak French
In WW1 the French won -- Allies!
In WW2 the French won -- Allies!
+++ I am from Germany! +++
paganism
Protestant
speak German
In WW1 the Germans lost -- Axis!
In WW2 the Germans lost -- Axis!
+++ I am from Russia! +++
paganism
Orthodox
speak Russian
In WW1 the Russians won -- Allies!
In WW2 the Russians won -- Allies!
+++ I am from Italy! +++
paganism
Roman Catholic
speak Italian
In WW1 the Italians won -- Allies!
In WW2 the Italians lost -- Axis!
BUILD SUCCESSFUL (total time: 0 seconds)
```