

톰 레벨 클래스는 한 파일에 하나만 담으라

밸런스 게임



토퍼벨 클래스는 한 파일에 하나만 담으라

소스 파일 하나에 토퍼벨 클래스를 여러 개 선언하면 자바 컴파일러는 어느 소스파일을 먼저 컴파일하냐에 따라 결과가 달라진다.

토크벨 클래스는 한 파일에 하나만 담으라

Utensil.java

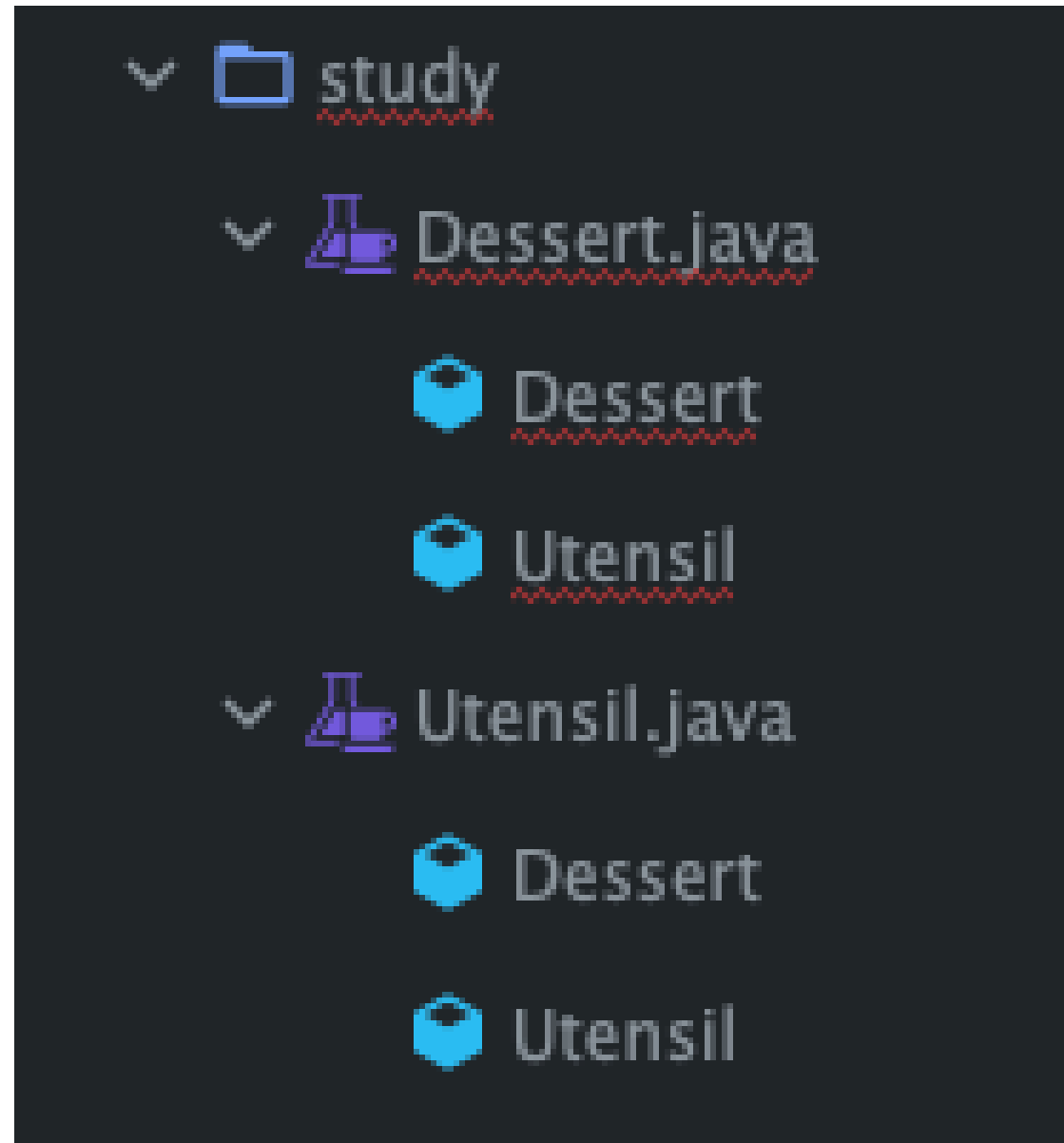
```
class Utensil {  
    static final String NAME = "pan";  
}  
  
class Dessert {  
    static final String NAME = "cake";  
}
```

Dessert.java

```
class Utensil {  
    static final String NAME = "pot";  
}  
  
class Dessert {  
    static final String NAME = "pie";  
}
```

같은 패키지에 클래스 정의

토크레벨 클래스는 한 파일에 하나만 담으라



```
package com.example.test.study;

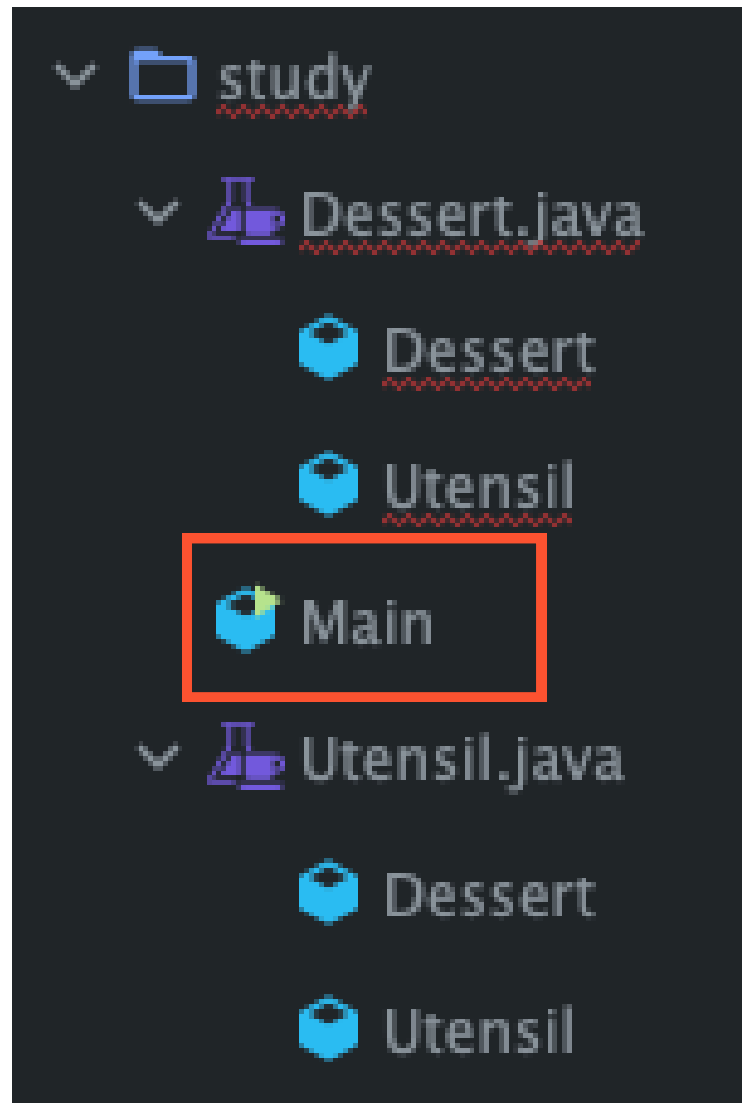
no usages
class Utensil {
    no usages
    static final String NAME = "pot";
}

no usages
class Dessert {
    no usages
    static final String NAME = "pie";
}
```

IntelliJ가 이미 중복된 클래스라고 말해주고 있다.

- ❗ Duplicate class found in the file '/Users/ryu/개발/test/src/main/java/com/example/test/study/Utensil.java' :3
- ❗ Duplicate class found in the file '/Users/ryu/개발/test/src/main/java/com/example/test/study/Utensil.java' :7

토크레벨 클래스는 한 파일에 하나만 담으라



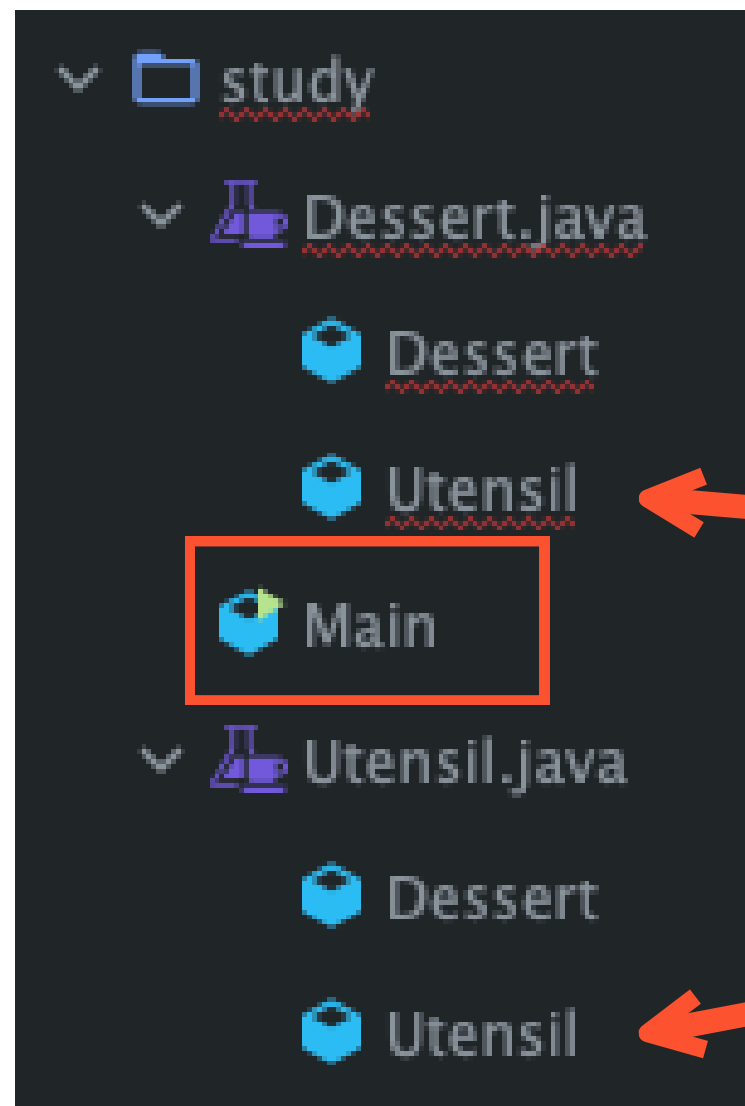
```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(Utensil.NAME + Dessert.NAME);  
    }  
}
```

같은 패키지에 Main 클래스 정의

톱레벨 클래스는 한 파일에 하나만 담으라

같은 패키지 내의 클래스들은 **import**문을 지정하지
않고도 사용 가능하다.

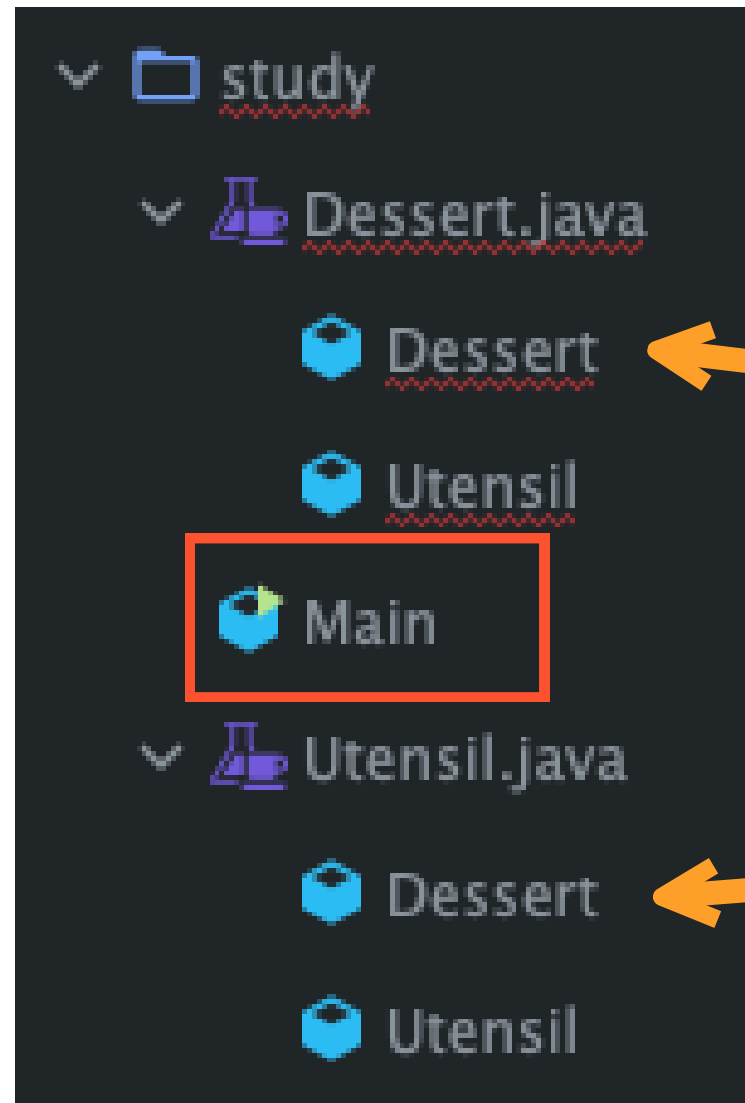
토크레벨 클래스는 한 파일에 하나만 담으라



```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(Utensil.NAME + Dessert.NAME);  
    }  
}
```

같은 패키지에 Main 클래스 정의

토크레벨 클래스는 한 파일에 하나만 담으라



```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(Utensil.NAME + Dessert.NAME);  
    }  
}
```

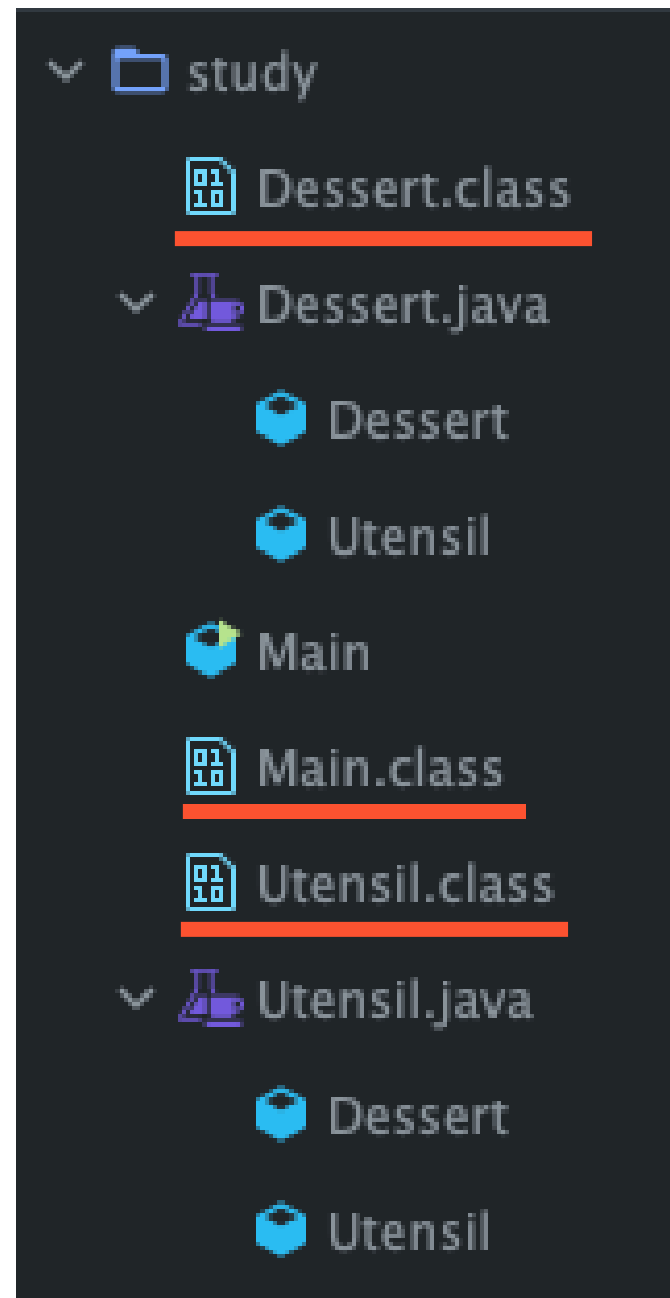
같은 패키지에 Main 클래스 정의

토크레벨 클래스는 한 파일에 하나만 담으라

javac 명령어에 들어가는 **인수에 따라 결과가 달라진다.**

- **javac Main.java Utensil.java**
- **javac Main.java Dessert.java**

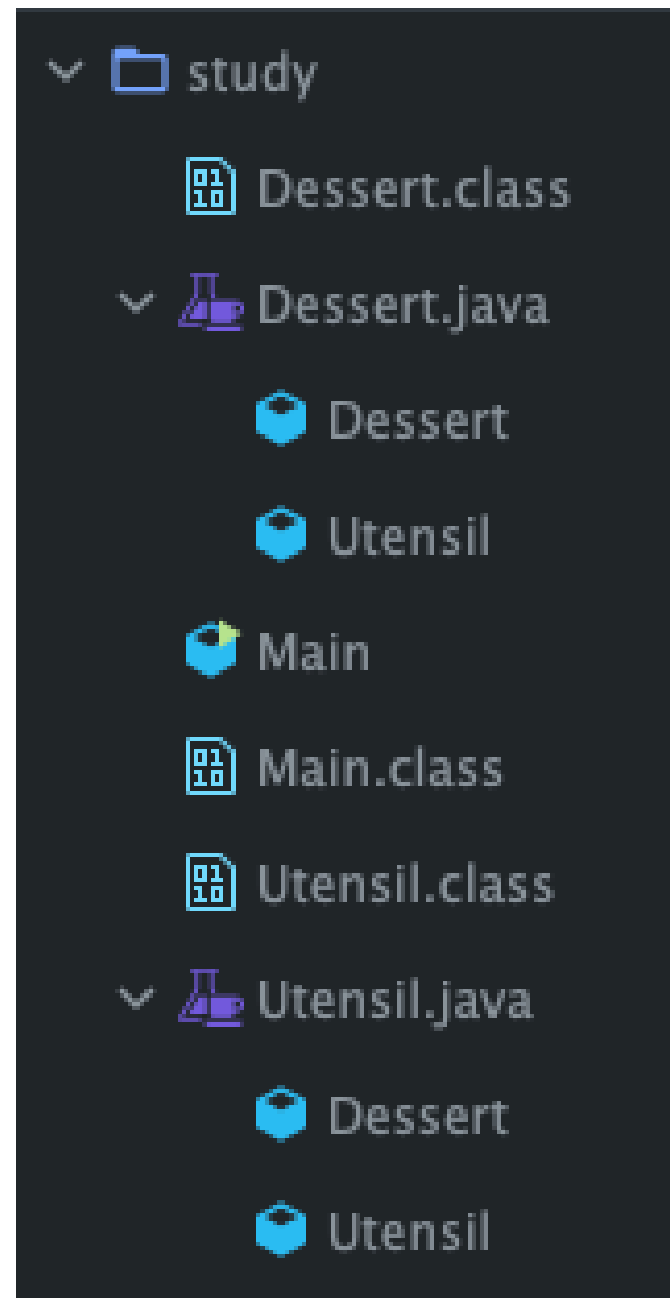
토크레벨 클래스는 한 파일에 하나만 담으라



```
Main.class x
Decompiled .class file, bytecode version: 65.0 (Java 21)
1 //
2 // Source code recreated from a .class file by IntelliJ IDEA
3 // (powered by FernFlower decompiler)
4 //
5
6 package com.example.test.study;
7
8 public class Main {
9     public Main() {
10    }
11
12     public static void main(String[] var0) {
13         System.out.println("pancake");
14     }
15 }
16
```

터미널 -> javac Main.java **Utensil.java**

토퍼벨 클래스는 한 파일에 하나만 담으라



```
Main.class x
Decompiled .class file, bytecode version: 65.0 (Java 21)
1 //
2 // Source code recreated from a .class file by IntelliJ IDEA
3 // (powered by FernFlower decompiler)
4 //
5
6 package com.example.test.study;
7
8 public class Main {
9     public Main() {
10    }
11
12     public static void main(String[] var0) {
13         System.out.println("potpie");
14     }
15 }
```

터미널 -> javac Main.java **Dessert.java**

토퍼벨 클래스는 한 파일에 하나만 담으라

토퍼벨 클래스 중복정의 해결방법

- 토퍼벨 클래스들을 서로 다른 소스 파일로 분리
- 서로 다른 여러 토퍼벨 클래스를 한 파일에 담고 싶다면 정적 멤버 클래스를 사용하는 방법을 고민해보자.

톱레벨 클래스는 한 파일에 하나만 담으라

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println(Utensil.NAME + Dessert.NAME);  
    }  
  
    private static class Utensil{  
        static final String NAME = "pan";  
    }  
  
    private static class Dessert{  
        static final String NAME = "cake";  
    }  
}
```

톱레벨 클래스들을 **정적 멤버 클래스**로 변경

정리

토플레벨 클래스는 한 파일에 하나만 담자

- 클래스 뿐만 아니라 인터페이스도 마찬가지다.

하나의 소스파일에 담아야 한다면 정적 멤버 클래스로 사용하자

