

Working with Modules



Sander Mak

Java Champion

@Sander_Mak www.javamodularity.com

Strong Encapsulation by Default

module-info.java

```
module mymodule {  
}
```

```
src/  
    com/pluralsight/A.java  
    com/pluralsight/B.java  
    com/pluralsight/util/C.java  
module-info.java
```

Strong Encapsulation by Default

module-info.java

```
module mymodule {  
}
```

```
src/  
    com/pluralsight/A.java  
    com/pluralsight/B.java  
    com/pluralsight/util/C.java  
module-info.java
```



Opt-out on a package basis

module-info.java

```
module mymodule {
```

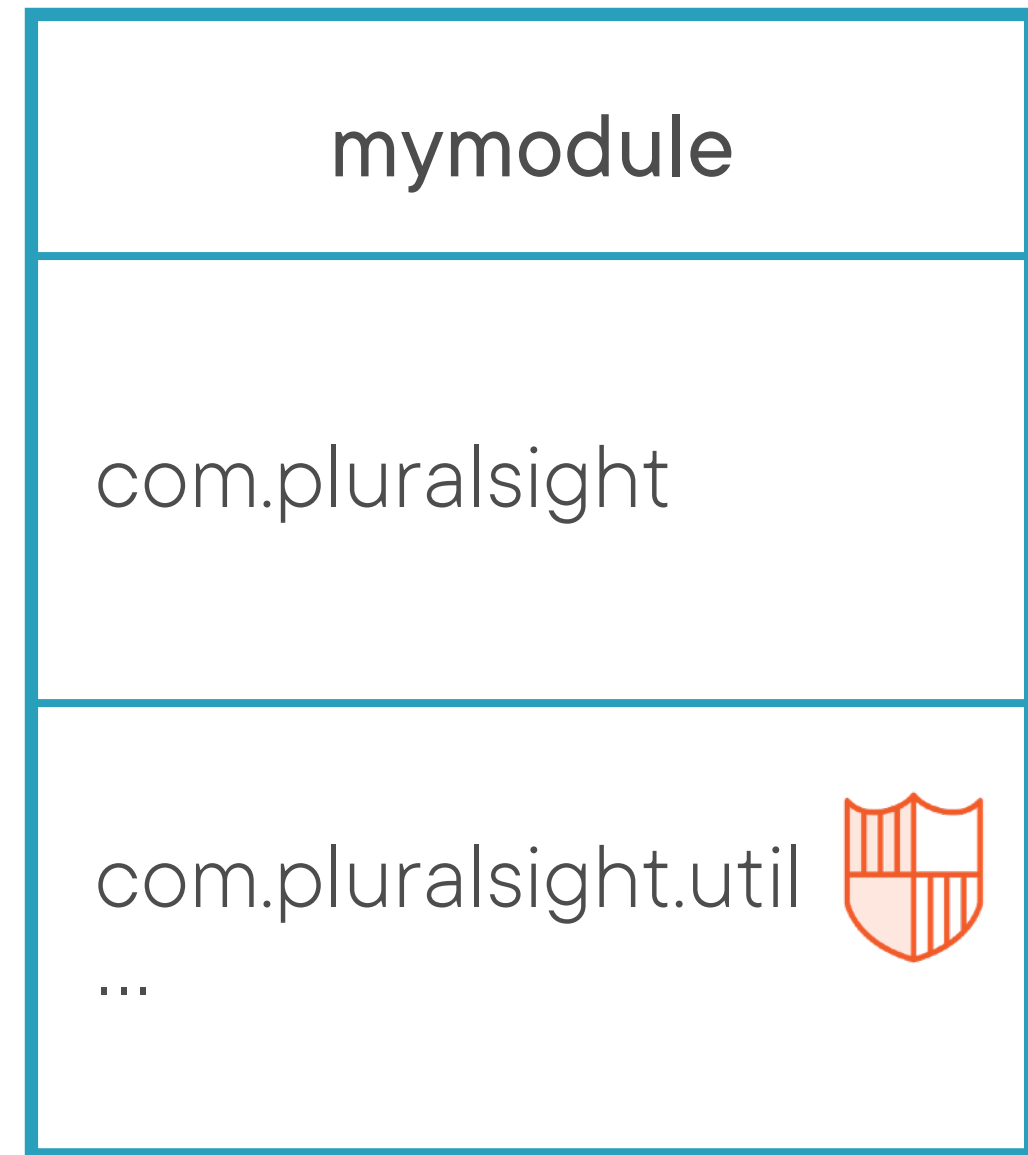
```
}
```

Exporting packages

module-info.java

```
module mymodule {  
  
  
  
  
  
}
```

Exporting packages



module-info.java

```
module mymodule {
```

```
    exports com.pluralsight;
```

```
}
```

Exporting packages

mymodule

com.pluralsight

com.pluralsight.util

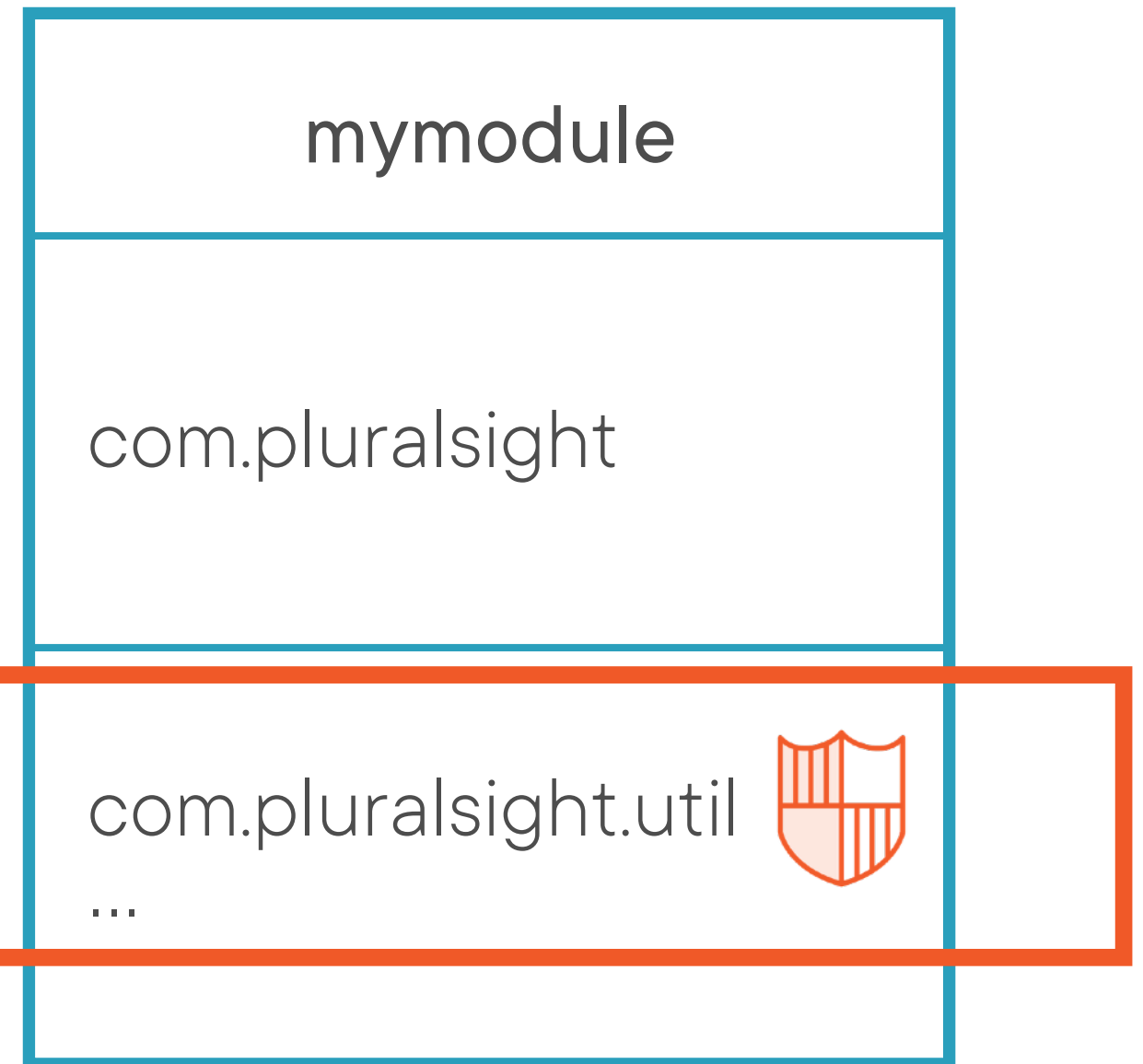
...



module-info.java

```
module mymodule {  
  
    exports com.pluralsight;  
  
}
```

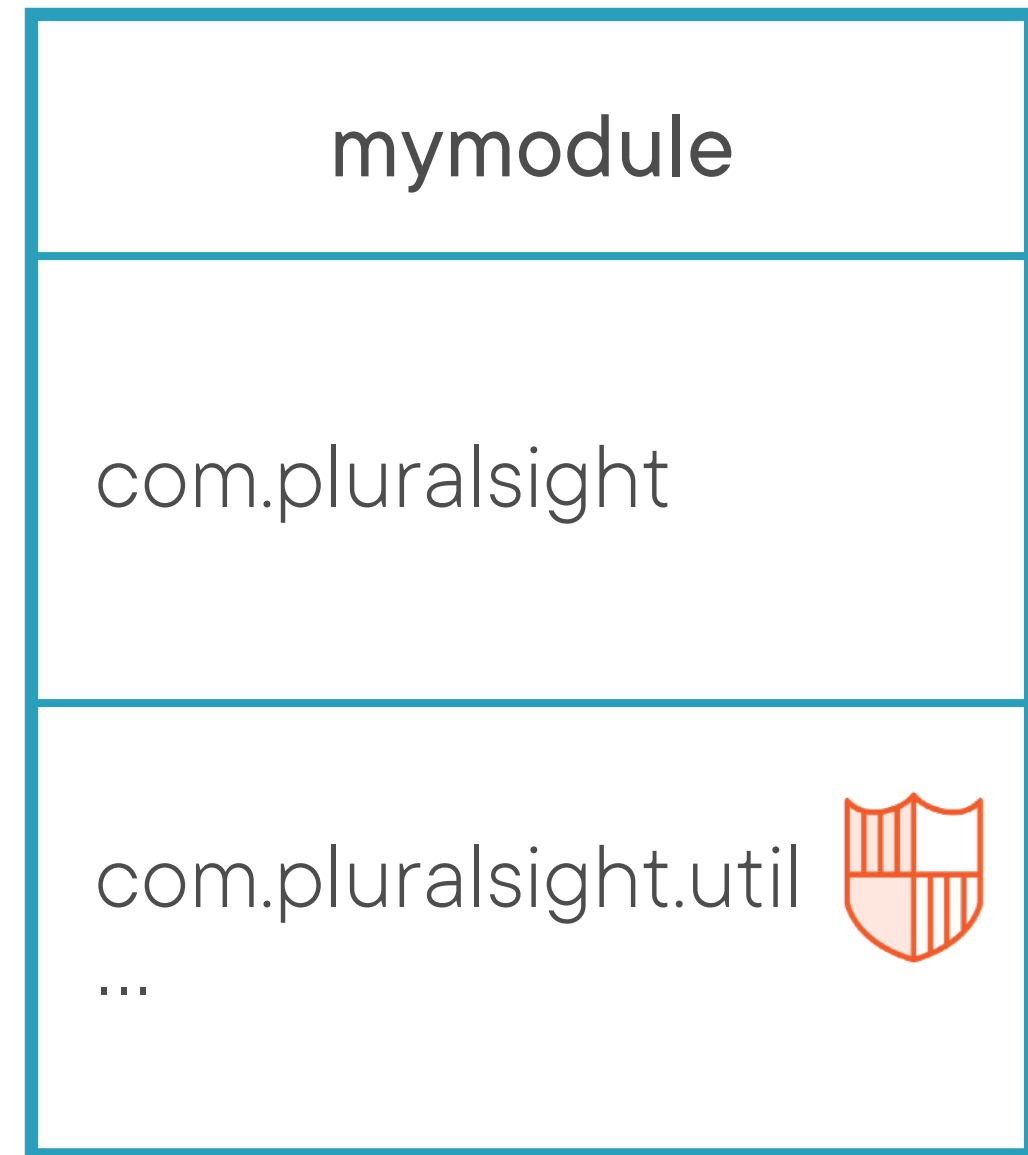
Exporting packages



module-info.java

```
module mymodule {  
  
    exports com.pluralsight;  
  
}
```

Exporting packages



Strong Encapsulation and Reflection

module-info.java

```
module mymodule {  
    exports com.pluralsight;  
}
```

```
src/  
    com/pluralsight/A.java  
    com/pluralsight/B.java  
    com/pluralsight/util/C.java  
module-info.java
```

Strong Encapsulation and Reflection

module-info.java

```
module mymodule {  
    exports com.pluralsight;  
}
```

```
src/  
    com/pluralsight/A.java  
    com/pluralsight/B.java  
    com/pluralsight/util/C.java  
module-info.java
```

```
Class.forName("com.pluralsight.util.C").newInstance()
```

Strong Encapsulation and Reflection

module-info.java

```
module mymodule {  
    exports com.pluralsight;  
}
```

```
src/  
    com/pluralsight/A.java  
    com/pluralsight/B.java  
    com/pluralsight/util/C.java  
module-info.java
```

```
Class.forName("com.pluralsight.util.C").newInstance()
```



IllegalAccessException



Dependencies between Modules

Dependencies between Modules



Dependencies between Modules



```
module mymodule {  
  requires util;  
}
```

Dependencies between Modules



```
module mymodule {  
  requires util;  
}
```

```
module util {  
  exports com.pluralsight.util;  
}
```

Dependencies between Modules



```
module mymodule {  
  requires util;  
}
```

```
module util {  
  exports com.pluralsight.util;  
}
```

requires is followed by a **module name**

Dependencies between Modules



```
module mymodule {  
  requires util;  
}
```

```
module util {  
  exports com.pluralsight.util;  
}
```

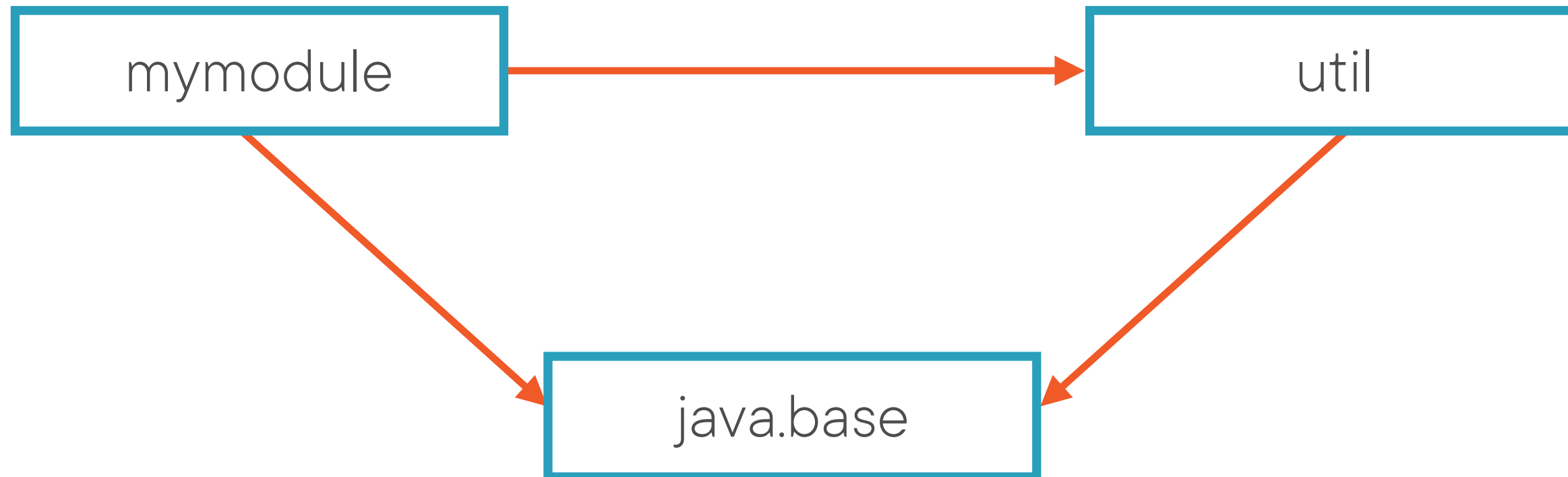
requires is followed by a **module name**

exports is followed by a **package name**

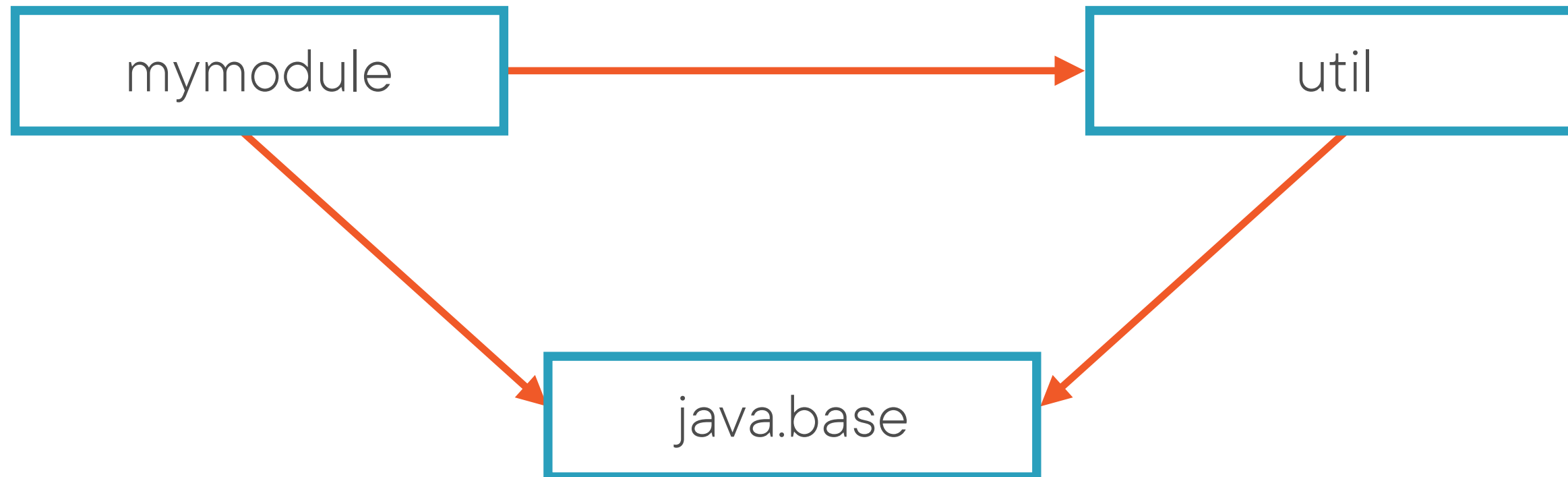
Implicit Dependency on java.base



Implicit Dependency on java.base



Implicit Dependency on java.base



```
module mymodule {  
    requires util;  
    requires java.base;  
}
```

Cyclic Dependencies Not Allowed

Cyclic Dependencies Not Allowed

Direct cycle

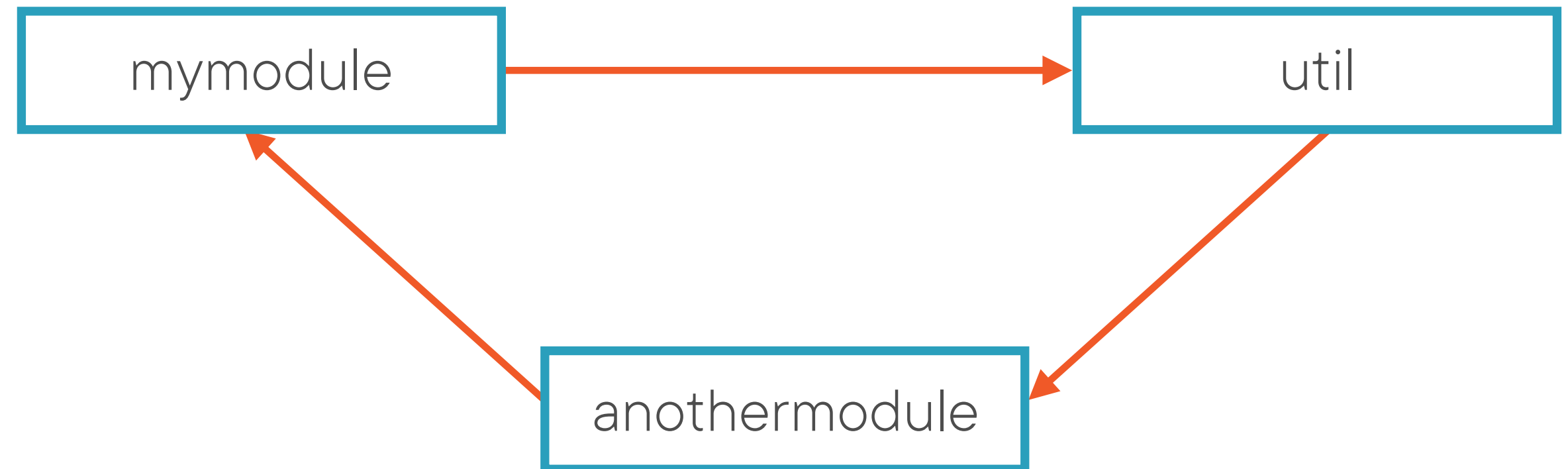


Cyclic Dependencies Not Allowed

Direct cycle

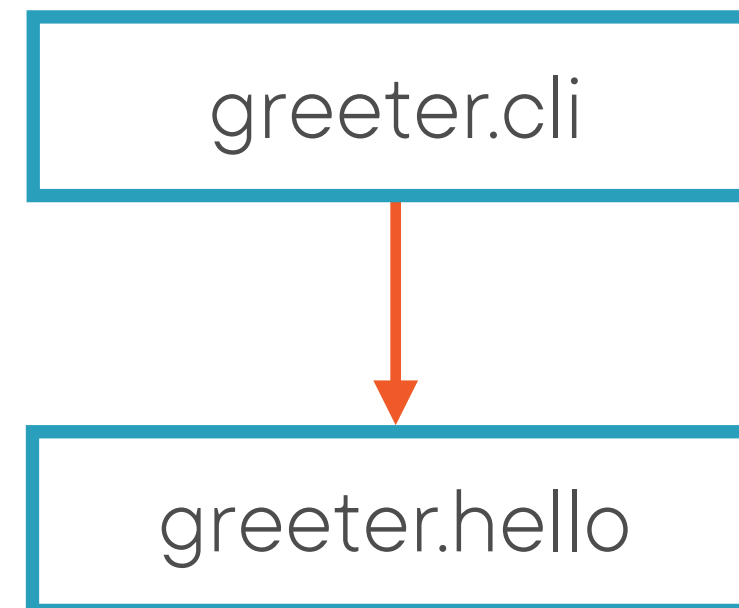


Indirect cycle



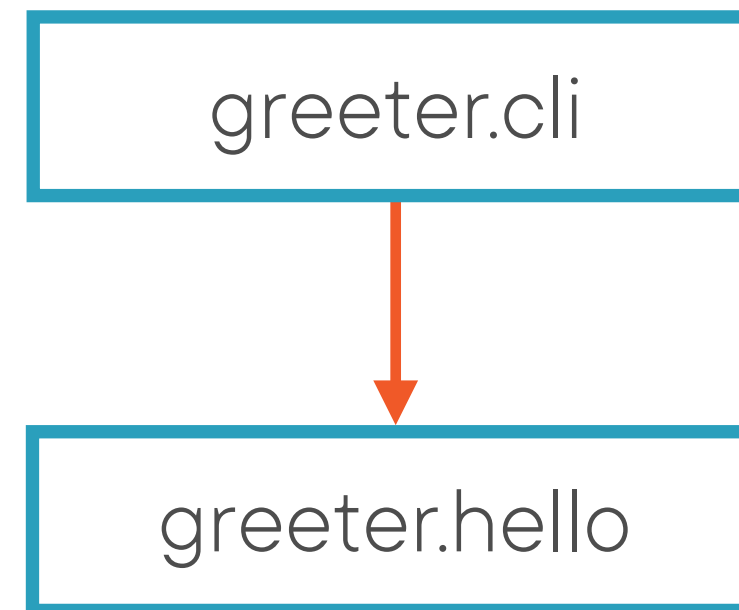
Demo

Working with Two Modules



Demo

Working with Two Modules



Open Modules and Packages

Open Modules and Packages

What if frameworks need
reflective access to strongly
encapsulated classes?

Open Modules and Packages

What if frameworks need **reflective access** to strongly encapsulated classes?

- Spring bean instantiation
- Hibernate entity creation
- Testing frameworks

Open Modules and Packages

What if frameworks need **reflective access** to strongly encapsulated classes?

- Spring bean instantiation
- Hibernate entity creation
- Testing frameworks

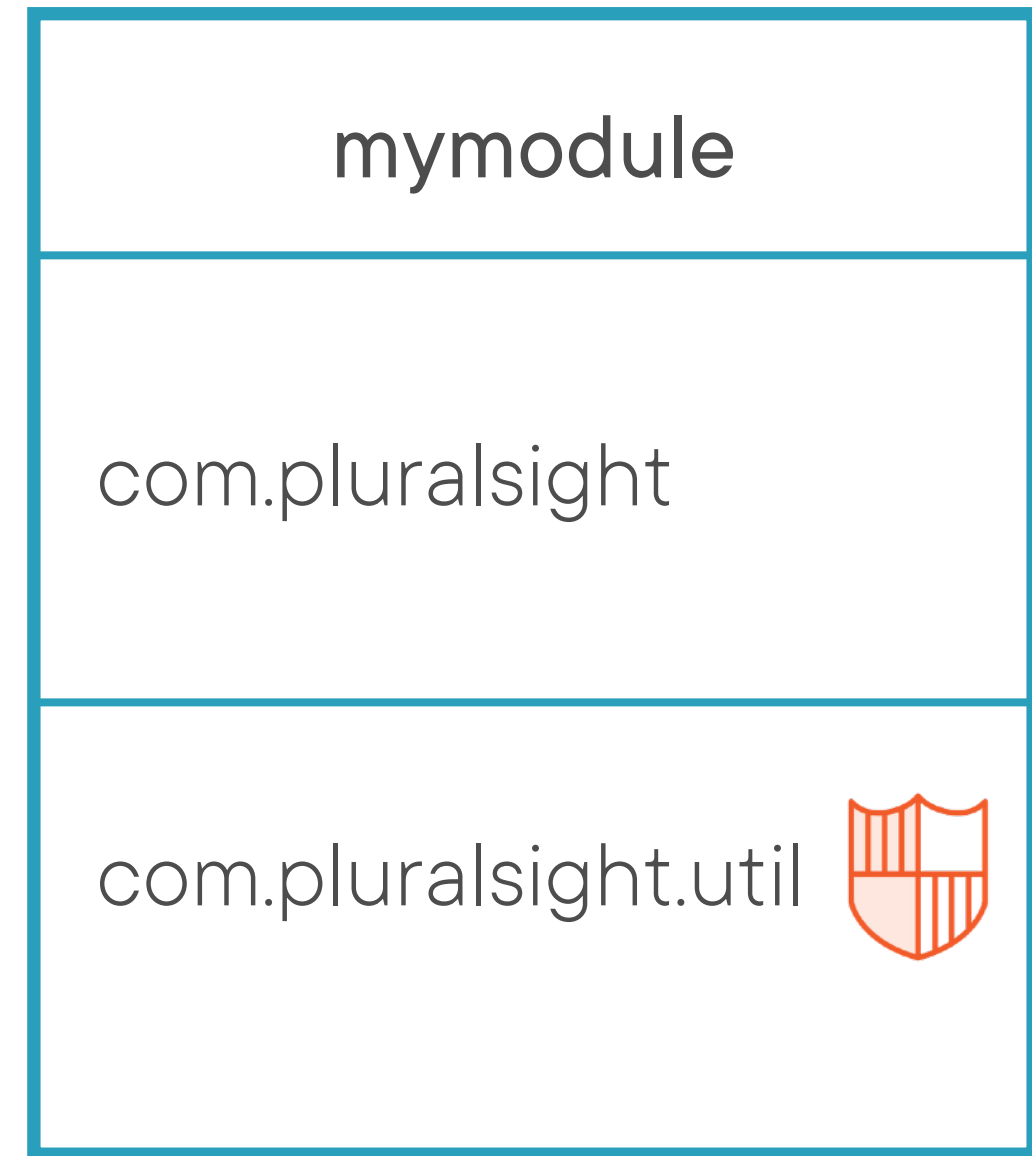
Compile-time vs. run-time

Open Modules and Packages

What if frameworks need **reflective access** to strongly encapsulated classes?

- Spring bean instantiation
- Hibernate entity creation
- Testing frameworks

Compile-time vs. run-time

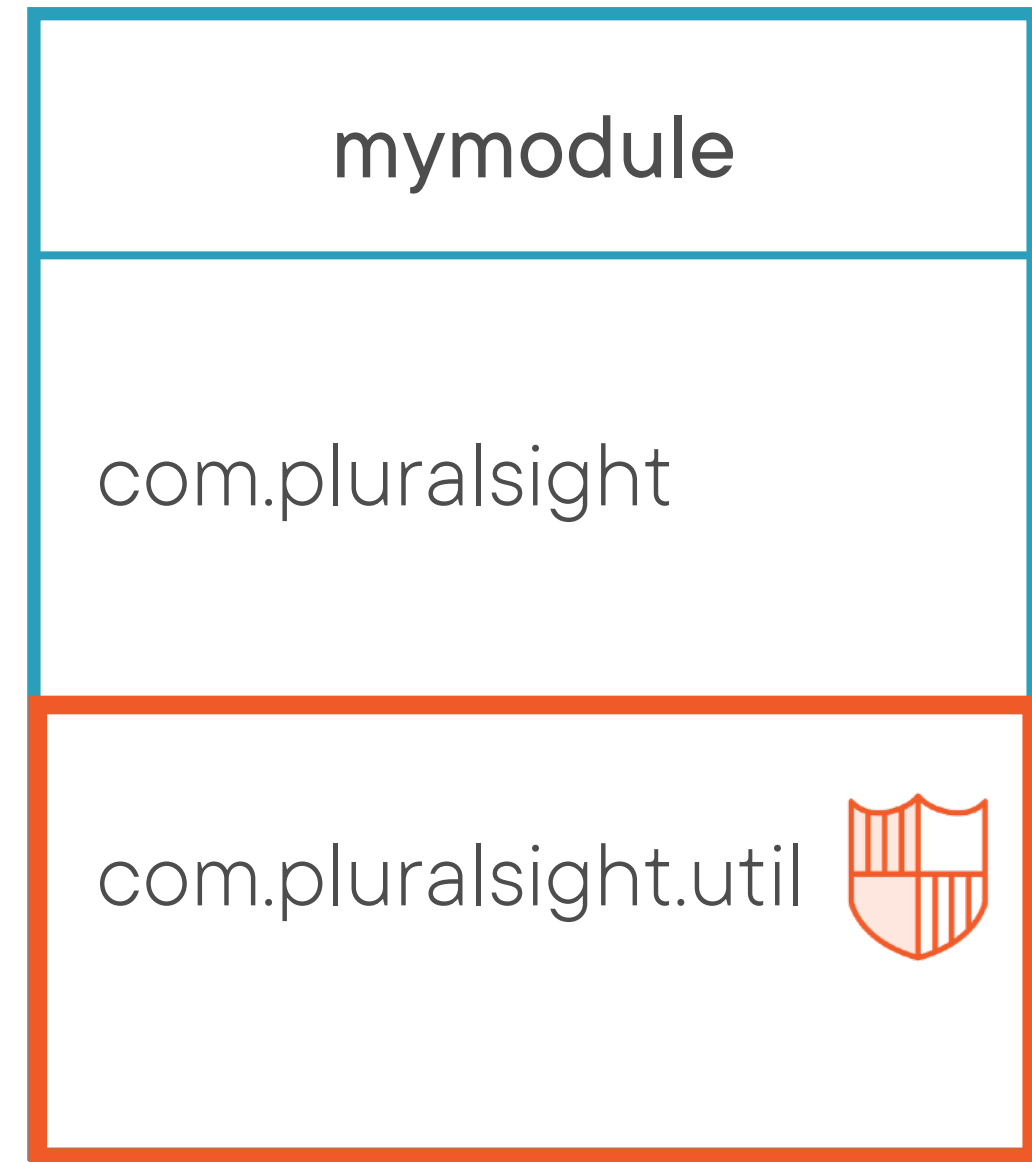


Open Modules and Packages

What if frameworks need **reflective access** to strongly encapsulated classes?

- Spring bean instantiation
- Hibernate entity creation
- Testing frameworks

Compile-time vs. run-time



Open Packages

```
module mymodule {  
  exports com.pluralsight;  
  opens com.pluralsight.util;  
  
}
```


Open Packages

Reflective access

```
module mymodule {  
    exports com.pluralsight;  
    opens com.pluralsight.util;  
  
}
```

Open Packages

Reflective access

```
module mymodule {  
  exports com.pluralsight;  
  opens com.pluralsight.util;  
  
}
```

```
Class.forName("com.pluralsight.util.C").newInstance()
```

Open Packages

Reflective access

Bypass access restrictions

```
module mymodule {  
    exports com.pluralsight;  
    opens com.pluralsight.util;  
  
}
```

```
Class.forName("com.pluralsight.util.C")  
    .getDeclaredField("secret")  
    .setAccessible(true)
```

Open Packages

Reflective access

Bypass access restrictions

Export & open

```
module mymodule {  
  exports com.pluralsight;  
  opens com.pluralsight;  
  opens com.pluralsight.util;  
}
```

Open Modules

Open Modules

```
open module mymodule {  
  exports com.pluralsight;  
  
}
```

Open Modules

```
open module mymodule {  
  exports com.pluralsight;  
  
}
```

An **open module** implicitly opens **all packages**

Open Modules

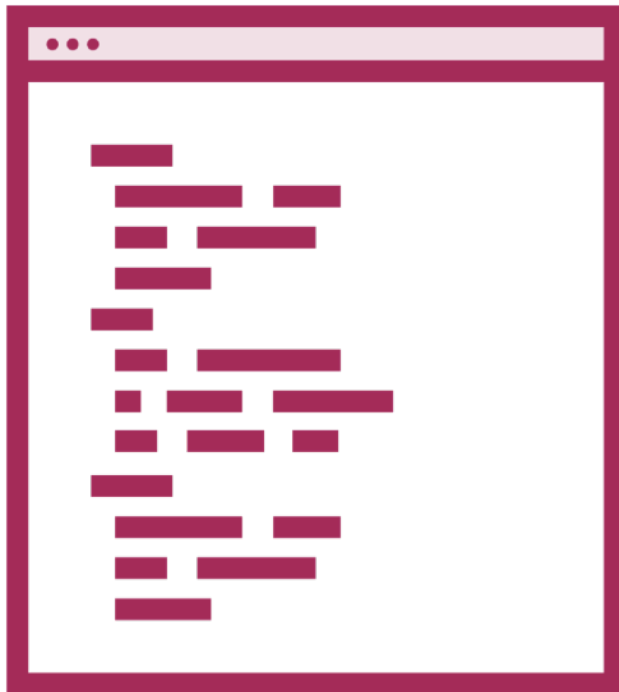
```
open module mymodule {  
    exports com.pluralsight;  
  
}
```

```
module mymodule {  
    exports com.pluralsight;  
    opens com.pluralsight;  
    opens com.pluralsight.util;  
}
```

An **open module** implicitly opens **all packages**

Overview: Module Declaration Syntax

Overview: Module Declaration Syntax



```
[open] module <module_name> {  
  exports <package>;  
  opens <package>;  
  requires <module_name>;  
}
```

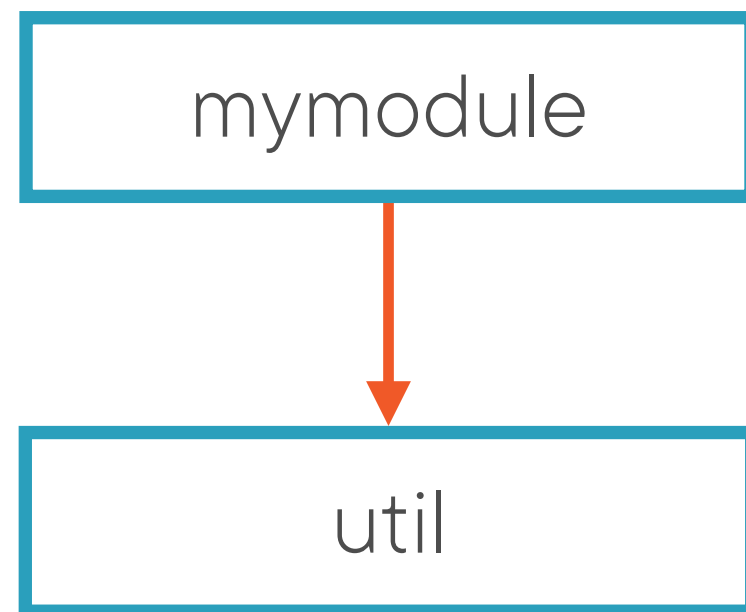
Qualified Exports

Qualified Exports

```
module util {  
  exports com.pluralsight.util to mymodule;  
}
```

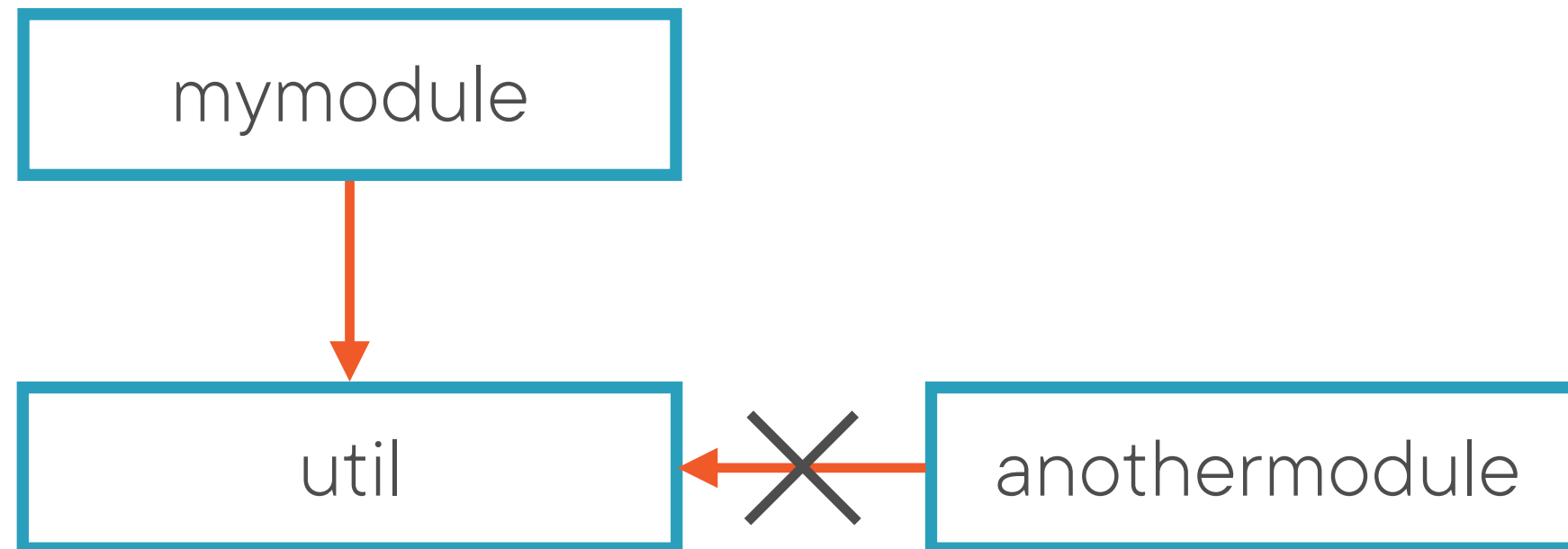
Qualified Exports

```
module util {  
  exports com.pluralsight.util to mymodule;  
}
```



Qualified Exports

```
module util {  
  exports com.pluralsight.util to mymodule;  
}
```



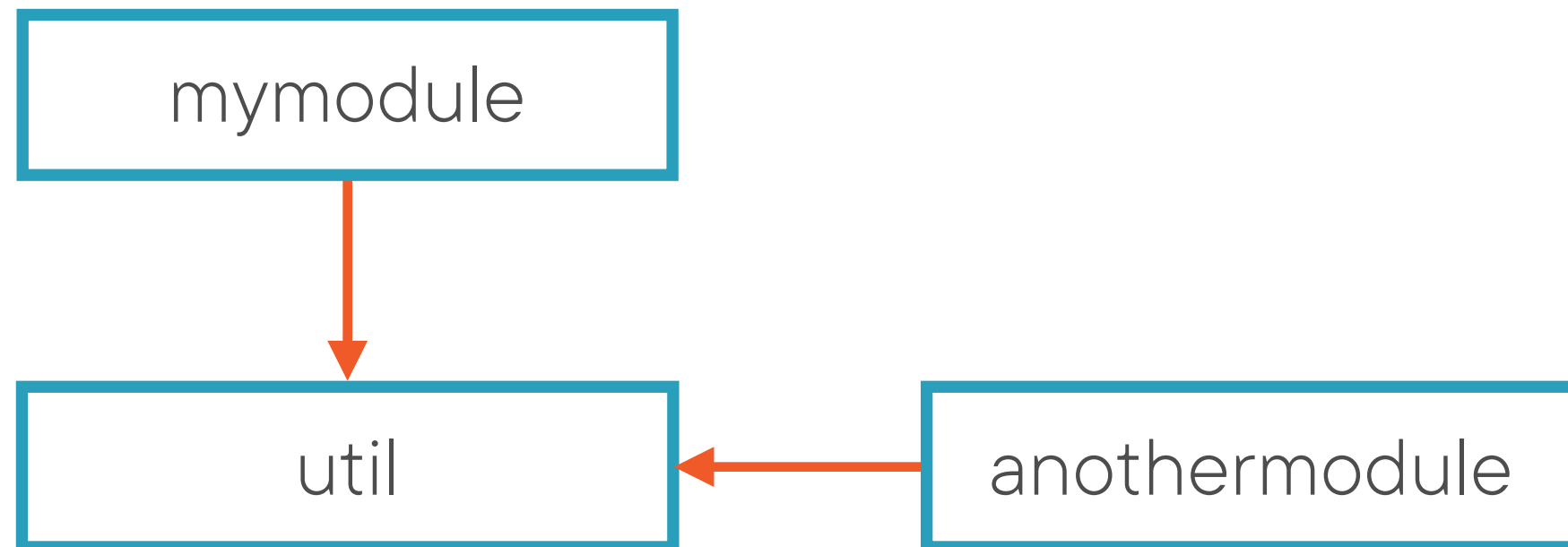
Qualified Opens

Qualified Opens

```
module util {  
    opens com.pluralsight.util to mymodule, anothermodule;  
}
```


Qualified Opens

```
module util {  
  opens com.pluralsight.util to mymodule, anothermodule;  
}
```



Summary

Summary

Summary

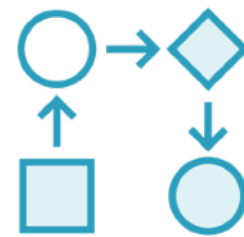


Exporting packages

Summary



Exporting packages

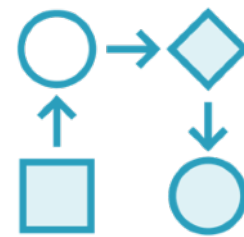


Requiring modules

Summary



Exporting packages



Requiring modules



Open packages and modules