

SoSe 2018

Seminar

Java 9 Modularity

Preliminary Meeting

Prof. Dr. Alexander Pretschner
Vadim Cebotari

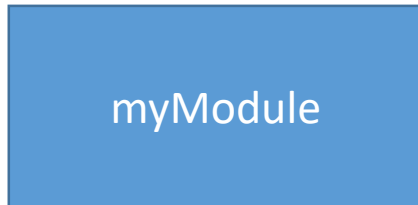
Java Modularity

- Main and most important feature introduced in Java 9
 - New architectural concept in Java language
 - Large-scale impact on Java applications
 - Focus on
 - Strong encapsulation
 - Explicit dependencies
 - Loose coupling

Java Module System

- New way of grouping code and data
 - Readability
 - Accessibility

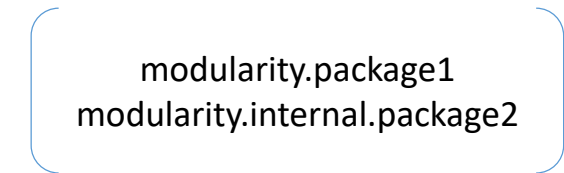
Java Module System



```
module myModule {  
  requires otherModule;  
}
```



```
module myModule {  
  requires java.xml;  
  
  exports modularity.package1;  
}
```



 **Is public still public?**

Java Module System

- Loose coupling between modules
- Improved support for extensibility using services mechanism

Java Module System

services.api

myModule

```
package modularity.services.api;
```

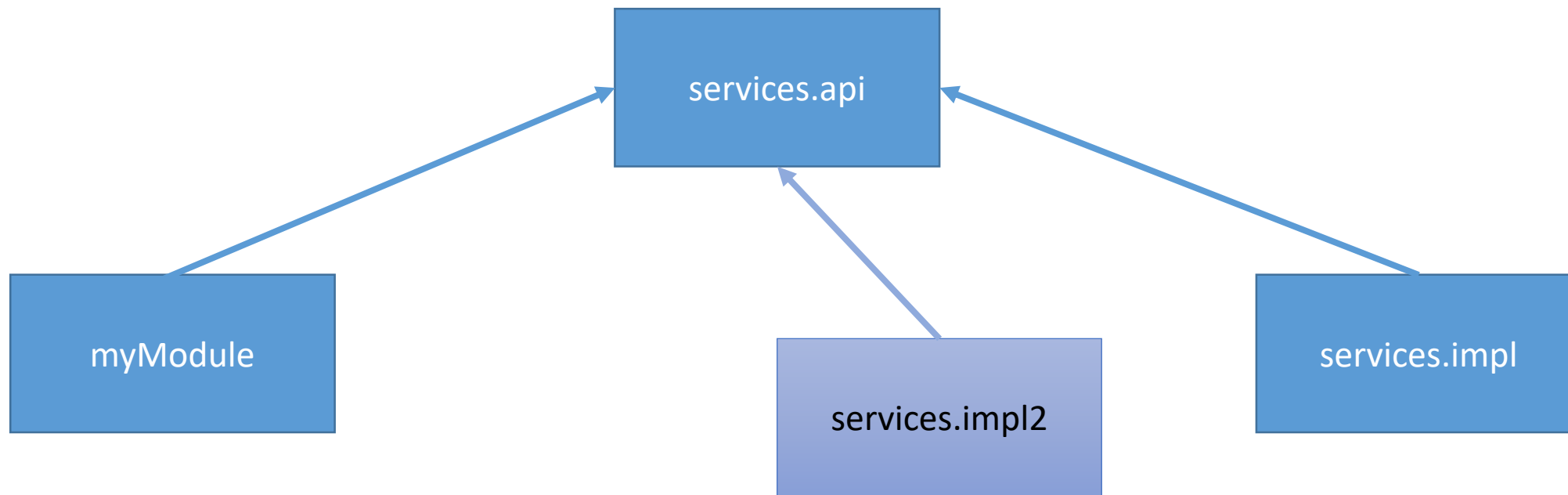
```
public interface Transformer {  
    void transform(String text);  
}
```

services.impl

```
module myModule {  
    requires services.api;  
  
    uses modularity.services.api.Transformer;  
}
```

```
module services.impl {  
    requires services.api;  
  
    provides modularity.services.api.Transformer  
    with modularity.services.impl.TransformerImpl;  
}
```

Java Module System



```
public class Main {  
    public static void main(String... args) {  
        ServiceLoader<Transformer> transformers =  
            ServiceLoader.load(Trasnformer.class);  
  
        // code to select the proper transfomer  
    }  
}
```

```
module services.impl2 {  
    requires services.api;  
  
    provides modularity.services.api.Transformer  
        with modularity.services.impl.TransformerImpl2;  
}
```

Java Module System

- Custom runtime images
 - Customized subset of JRE
 - Optimization of performance and memory usage
 - Optional linking phase via **jlink**

Topics

- Concept of modularity. Advantages and core principles.
- JDK platform modules and application modules.
- Modularity patterns.
- Migration to modules.
- Testing modules.
- Custom runtime images.

Literature

1. Paul Bakker, Sander Mark „Java 9 Modularity“, O‘Reilly Media, 2017
2. Kirk Knoernschild „Java Application Architecture: Modularity Patterns with Examples Using OSGi“, Addison-Wesley Professional, 2012
3. JEP 261: Module System

Administrative

- Rules and policies will be published on the website of chair of software and systems engineering
 - See [Teaching](#)
- Bachelor-/Master Seminar
- Maximum participants: 9
- Registration
 - Via matching.in.tum.de