

Otras características de la modularidad




Exportación específica

- Se puede indicar que un determinado paquete de un módulo sea accesible únicamente por cierto módulo.
- Se emplea la palabra reservada *to*:

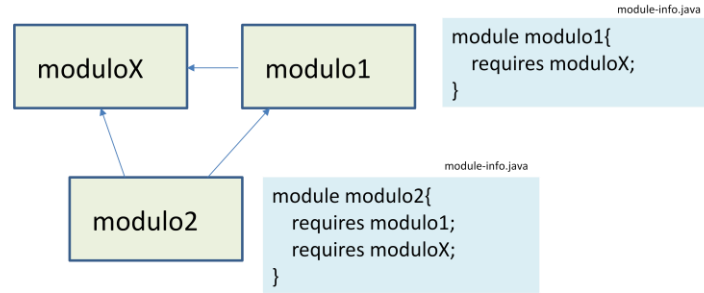
```
module modulo1{  
  exports com.paquete1 to moduloX, modK;  
}
```

El paquete
com.paquete1 sólo
es accesible para
moduloX y modK

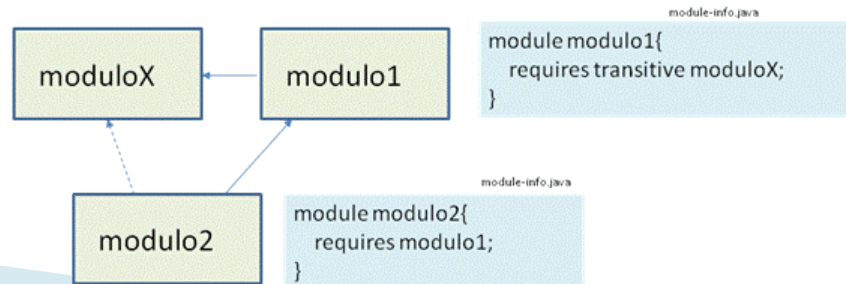


Dependencias transitivas

➤ Evita redundancias a la hora de requerir dos módulos en los que, a su vez, uno requiere al otro:



➤ Con dependencia transitiva:



Servicios

➤ Servicio: Interfaz definida en un módulo:

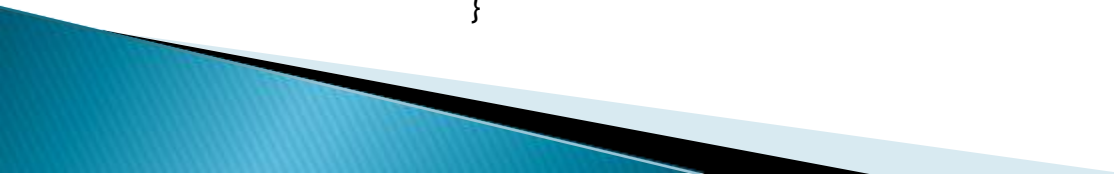
```
module service{  
  exports com.Interfaz1;  
}
```

➤ Proveedor de servicio: Módulo que implementa la interfaz:

```
module proveedor{  
  provides com.Interfaz1 with com.Clase1;  
}
```

➤ Consumidor: Módulo que utiliza el servicio:

```
module consumidor{  
  uses com.Interfaz1;  
}
```



Servicios

➤ Servicio: Interfaz definida en un módulo:

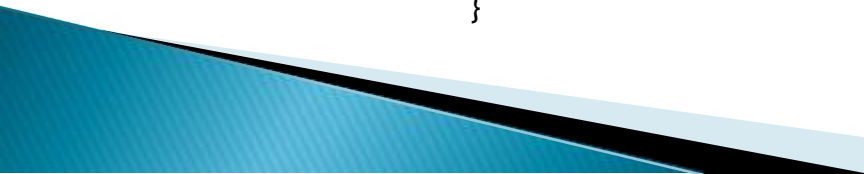
```
module service{  
  exports com.Interfaz1;  
}
```

➤ Proveedor de servicio: Módulo que implementa la interfaz:

```
module proveedor{  
  provides com.Interfaz1 with com.Clase1;  
}
```

➤ Consumidor: Módulo que utiliza el servicio:

```
module consumidor{  
  uses com.Interfaz1;  
}
```



Acceso por reflexión

- Es posible especificar que los paquetes especificados del módulo son accesibles vía reflexión a otros módulos:

```
module moduloX{  
    opens com.paquete;  
}
```

- Incluso, que lo sean solo a ciertos módulos:

```
module moduloX{  
    opens com.paquete to modA, modB;  
}
```

