ACCESS MODIFIERS

MODIFIERS
public
private
protected
default

ACCESS MODIFIERS

- Set the accessibility of classes, methods, and other members.
- Classes:
  - Visibility of the class by another classes.

  public class A → can see → public class B ← *Class B is visible by Class A*
- Attributes or methods:
  - El modificador de acceso determina si son ACCESIBLES, ya sea al crear un objeto, al querer usar dichos atributos o métodos si son estáticos o al heredar.

  Clase cObjeto = new Clase();

  cObjeto.metodoPublico(); ← el "*metodoPublico*()" es **accesible**

WHY

- Access modifiers let us control the “Access” to the most important elements of our code
- What would happen if anyone could take control of a plane?
  - The plane still could work, but think about safety.
  - Some elements or our code must be protected.

WHY

- What would happen if everything was innaccesible? A useless airplane
  - Some things must be accesible to be useful.
- Must be a balance.

IMPACT

Access modifiers

- Access modifiers:
  - public
  - protected
  - private
  - (default)
- Other:
  - strictfp
  - final
  - abstract

PUBLIC

- A class declaration with the public keyword gives all classes from all packages access to the public class.
- All classes in the Java Universe (JU) have access to a public class.

DEFAULT

- A class with default access has **no modifier** preceding it in the declaration!
- Think of default access as **package-level** access.
- A class with default access can be seen only by classes within the same package

PROTECTED

- Specifies that a member can only be accessed within its own package (as with default) and, in addition, by a subclass of its class in another package.

PRIVATE (INNER|NESTED CLASSES)

- Members marked private can't be accessed by code in any class other than the class in which the private member was declared.

strictfp

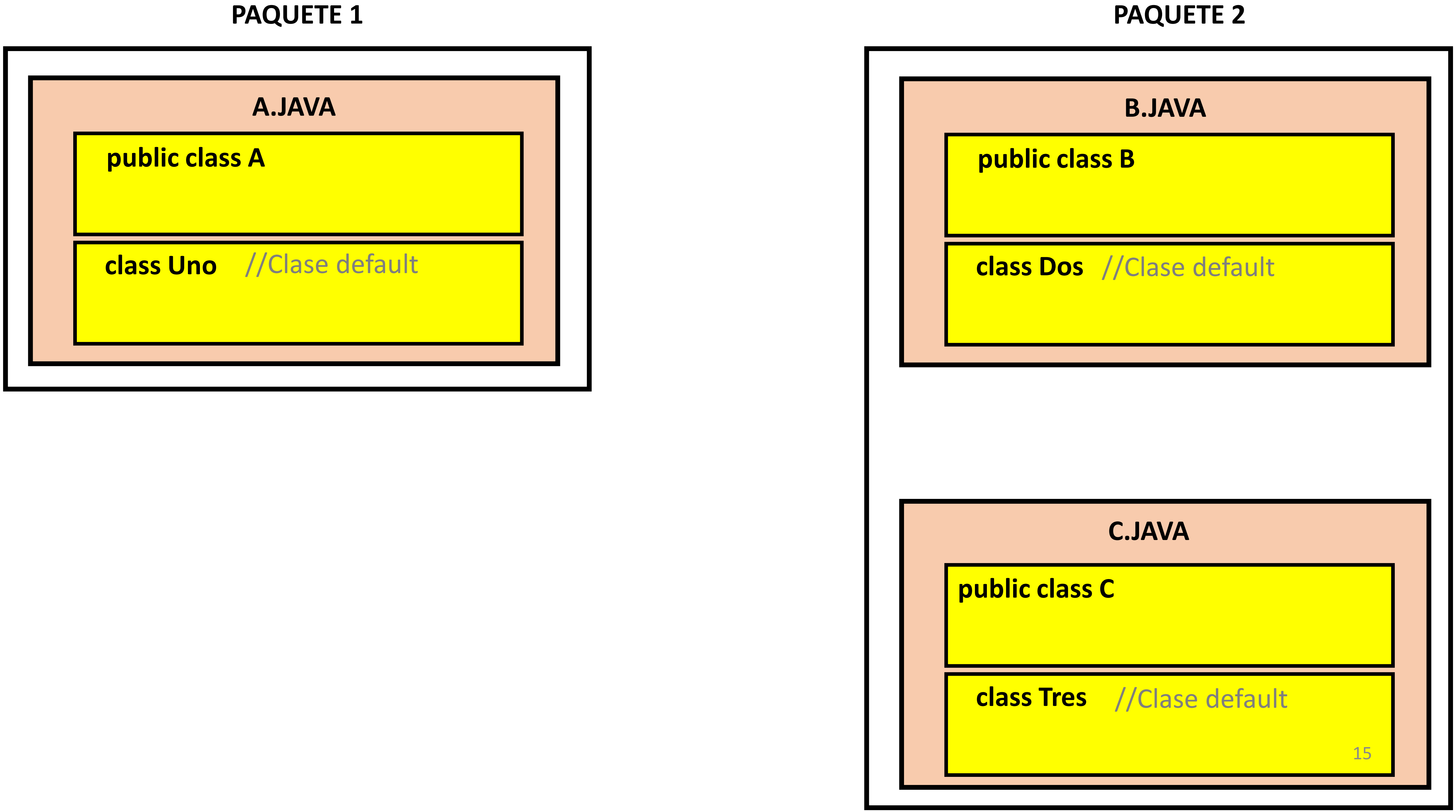
- Restricts floating-point calculations to ensure portability.

final

- A final class cannot be subclassed.

abstract

- Abstract classes cannot be instantiated, but they can be subclassed.



PAQUETE 1

A.JAVA

public class A

class Uno //Clase default

PAQUETE 2

B.JAVA

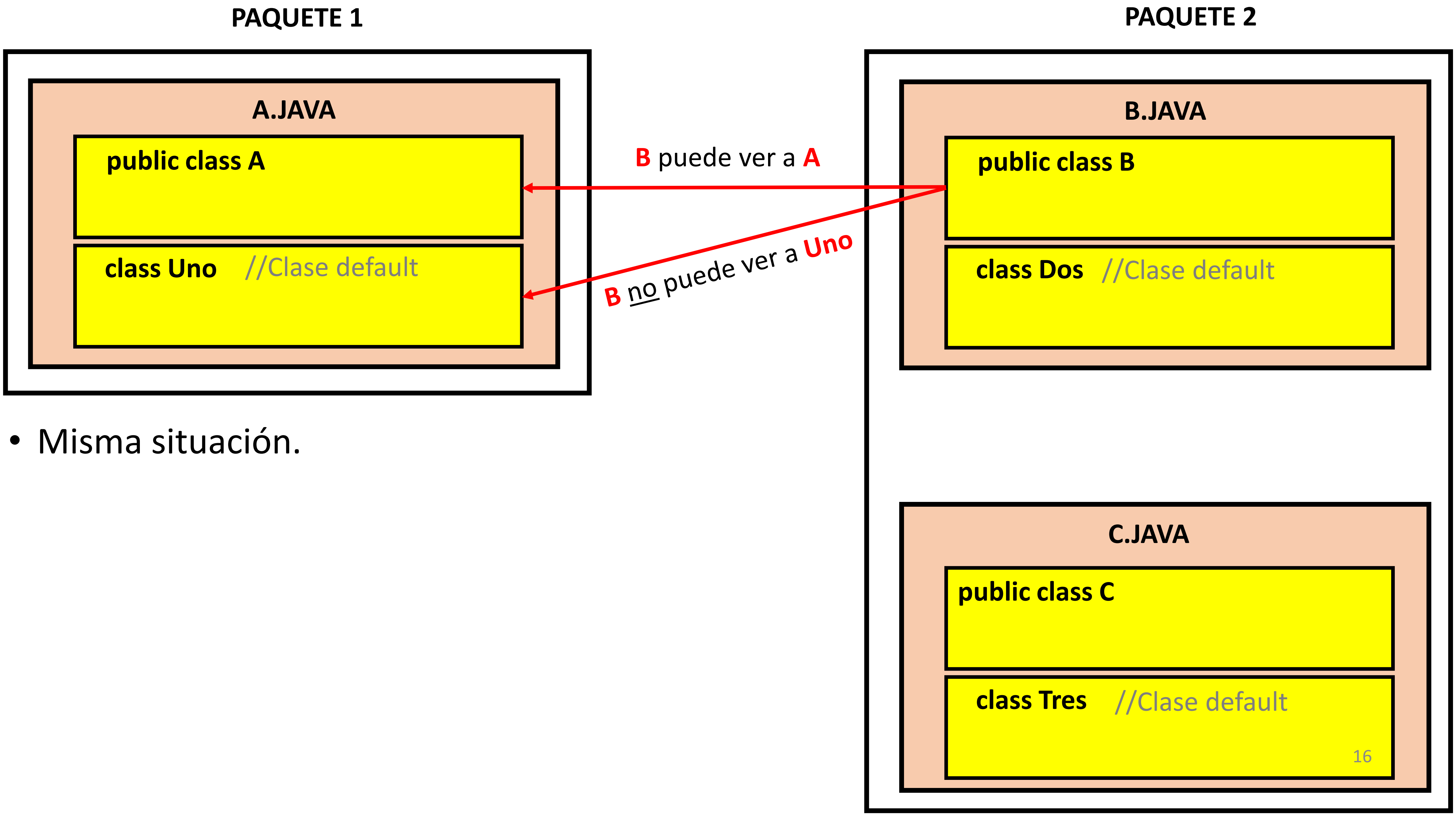
public class B

class Dos //Clase default

C.JAVA

public class C

class Tres //Clase default



PAQUETE 1

A.JAVA

public class A

class Uno //Clase default

PAQUETE 2

B.JAVA

public class B

class Dos //Clase default

C.JAVA

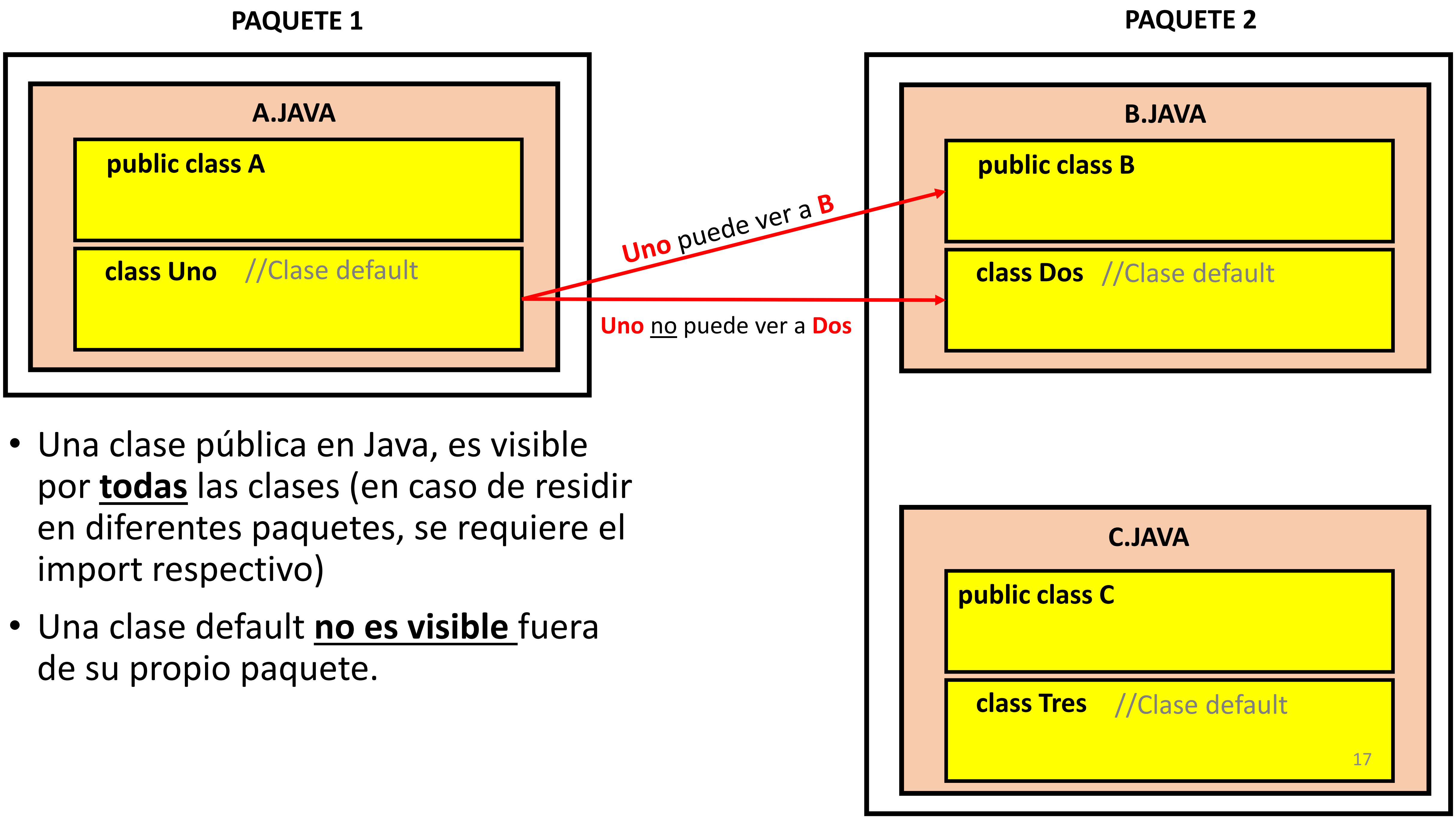
public class C

class Tres //Clase default

B puede ver a A

B no puede ver a Uno

- Misma situación.



PAQUETE 1

A.JAVA

public class A

class Uno //Clase default

PAQUETE 2

B.JAVA

public class B

class Dos //Clase default

C.JAVA

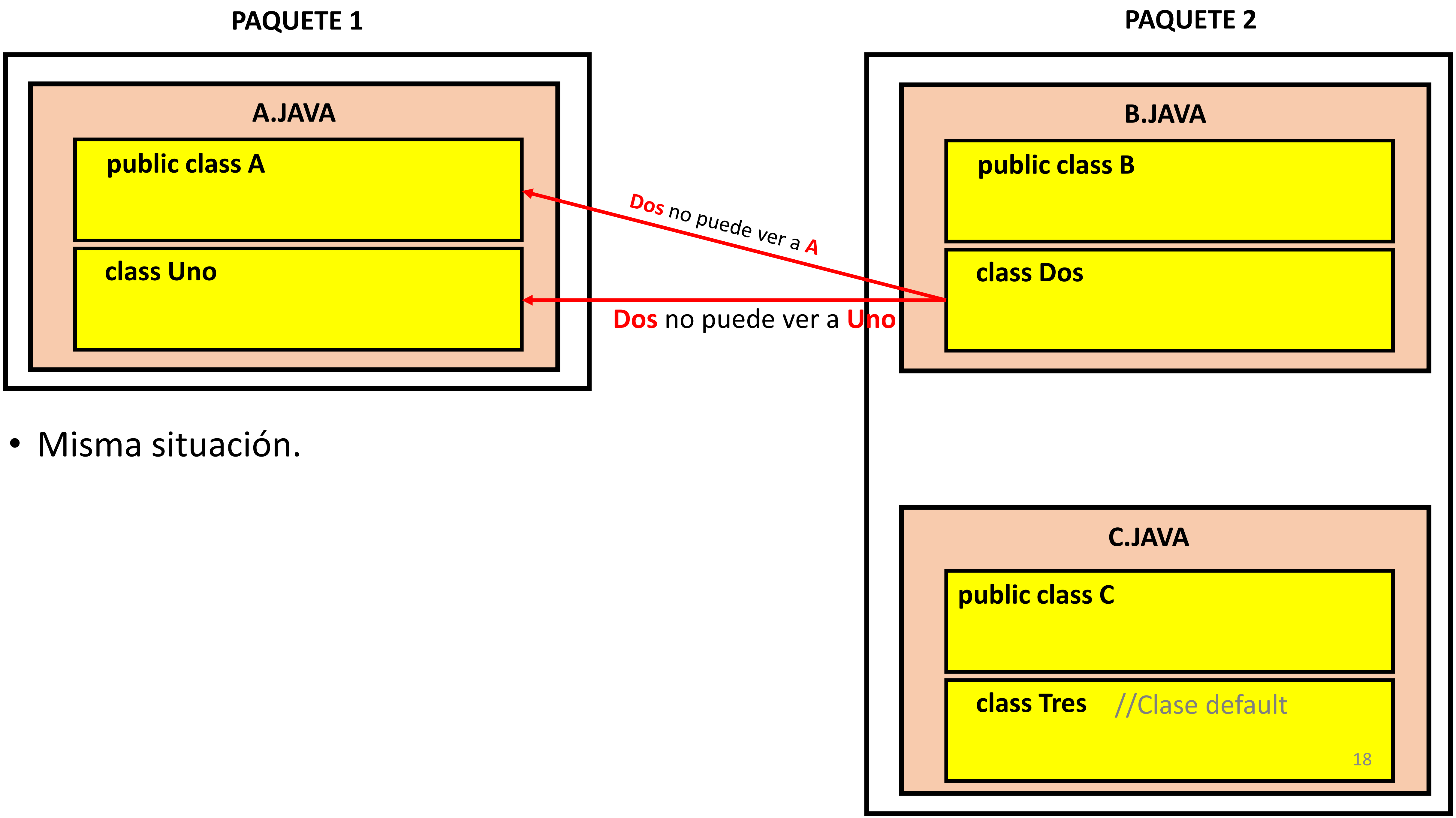
public class C

class Tres //Clase default

Uno puede ver a **B**

Uno <u>no</u> puede ver a **Dos**

- Una clase pública en Java, es visible por <u>**todas**</u> las clases (en caso de residir en diferentes paquetes, se requiere el import respectivo)
- Una clase default <u>**no es visible**</u> fuera de su propio paquete.



PAQUETE 1

A.JAVA

public class A

class Uno

PAQUETE 2

B.JAVA

public class B

class Dos

C.JAVA

public class C

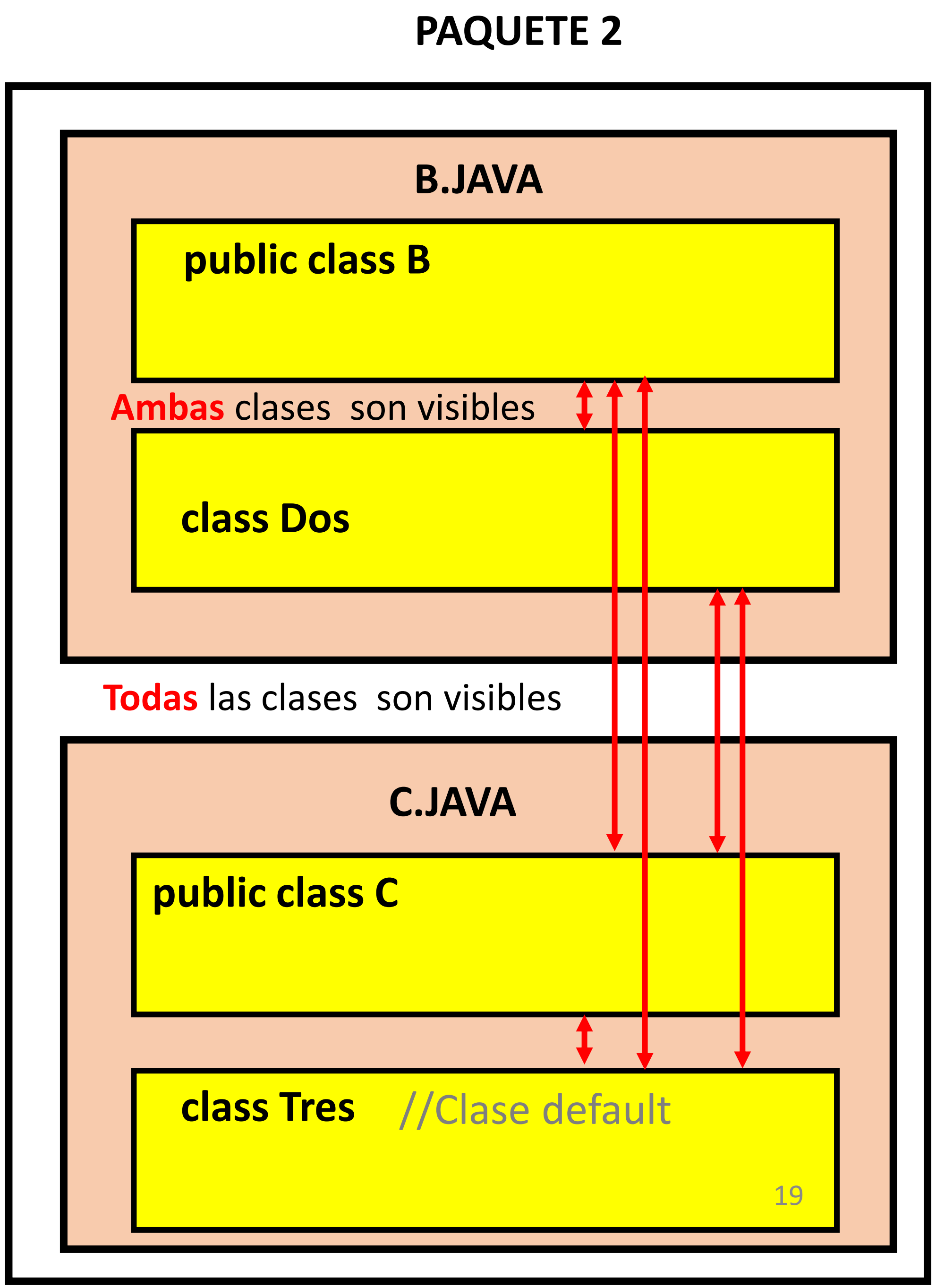
class Tres //Clase default

Dos no puede ver a A

Dos no puede ver a Uno

- Misma situación.

- En un mismo paquete, las clases default y public son visibles entre si.


PAQUETE 2
B.JAVA
public class B
Ambas clases son visibles
class Dos
Todas las clases son visibles
C.JAVA
public class C
class Tres //Clase default