

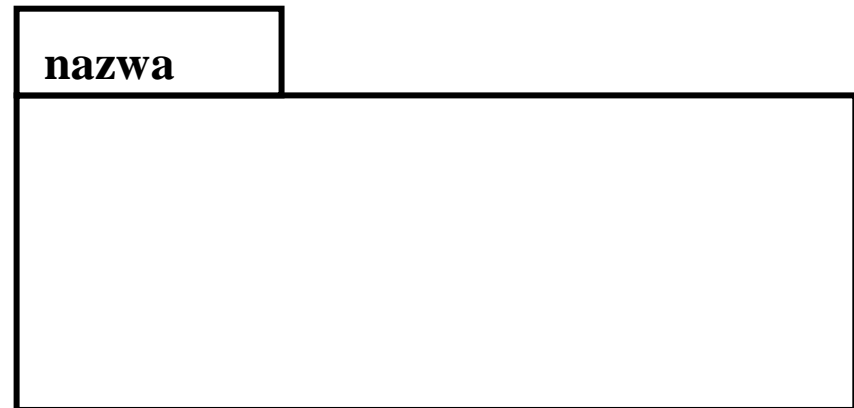
# Pakiety w UML 2.x

---

Dr hab. inż. Ilona Bluemke

# Pakiet

- Pojemnik zawierający byty dowolnego rodzaju (np. klasy, interfejsy, przypadki użycia, inne pakiety itd.)
- Symbol graficzny



## Pakiet - 2

- jest grupą innych bytów: bez tożsamości i egzemplarzy
- określa przestrzeń nazw :  
***pakiet\_zewnetrzny::pakiet\_wewnetrzny::element***
- ogranicza widoczność: publiczne, prywatne, chronione

# Zastosowanie pakietów

## 1. Zarządzanie projektem

- podział systemu na heterogeniczne pakiety, grupujące np. klasy.
- Pakiet zawiera artefakty związane z budową części systemu

## 2. Dekompozycja systemu

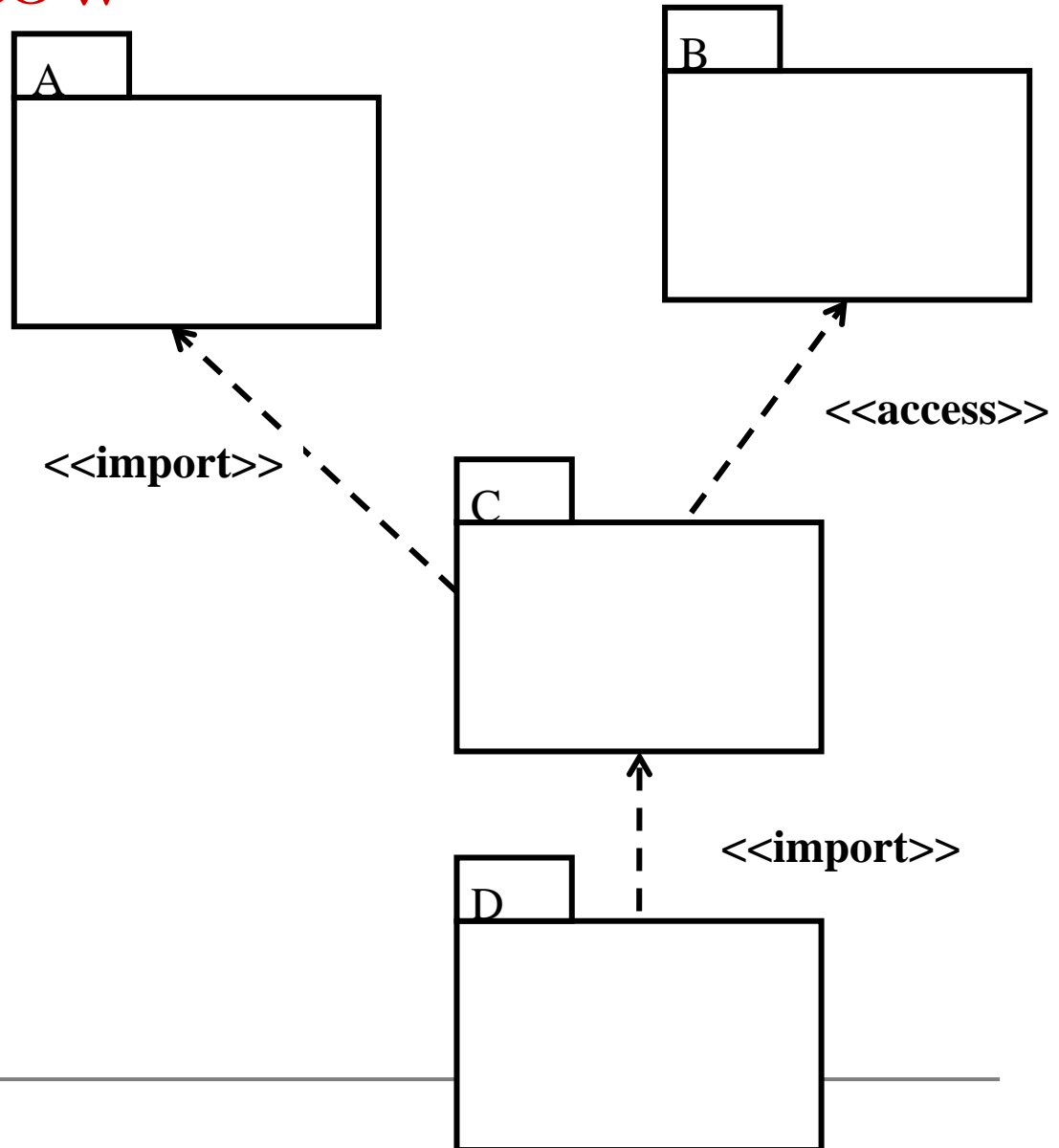
- podział systemu na jednorodne pakiety (klas) opisuje strukturę systemu na wyższym poziomie abstrakcji. Podział na pakiety może poprawić modyfikowalność.

# Diagram pakietów

Elementy zdefiniowane w pakiecie A są publicznie importowane do pakietu C.

Elementy w pakiecie B są importowane prywatnie do pakietu C.

Elementy pakietu A i C są dostępne dla pakietu D. Import pakietu nie jest przechodni.

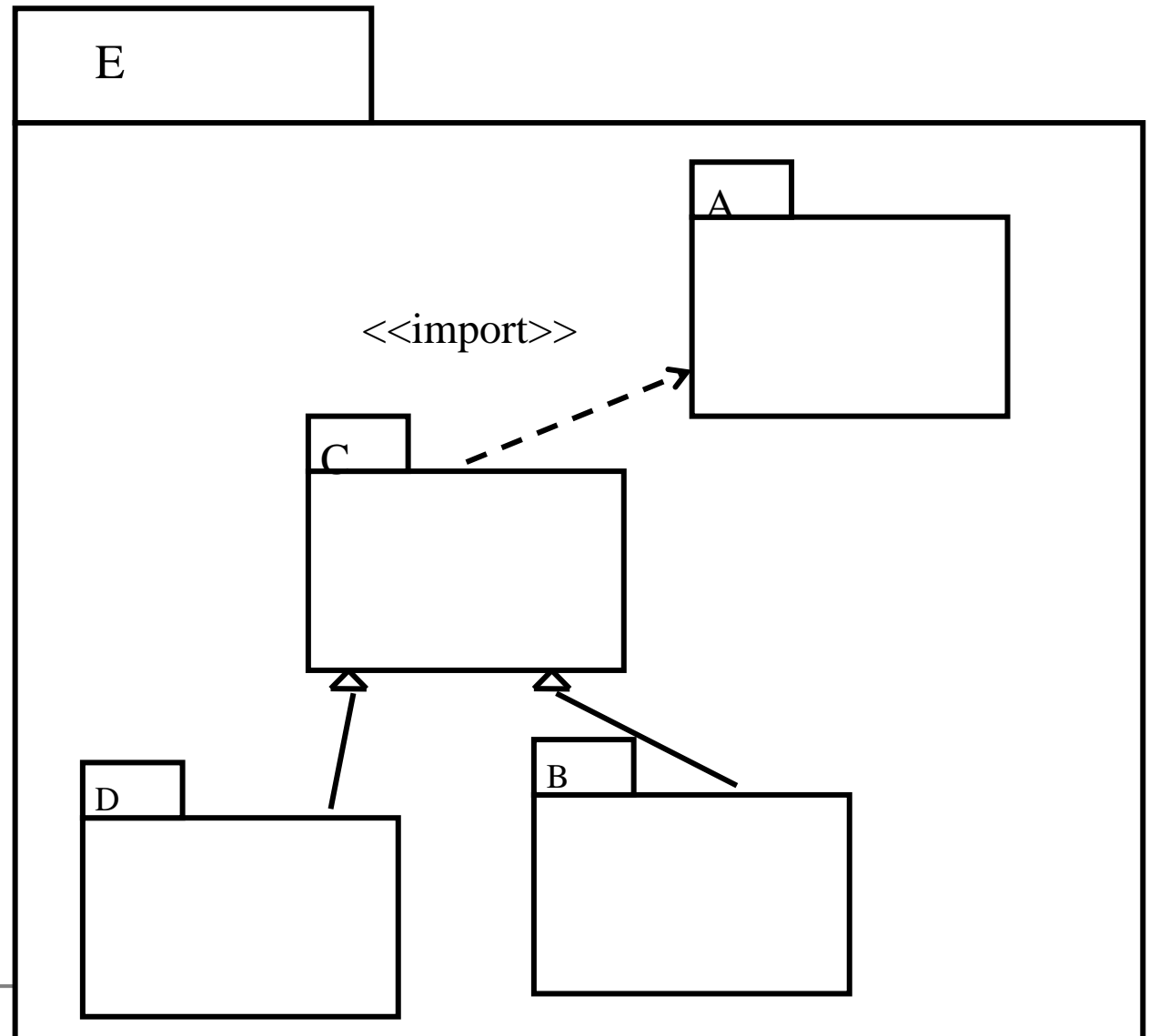


# Przykład pakietów

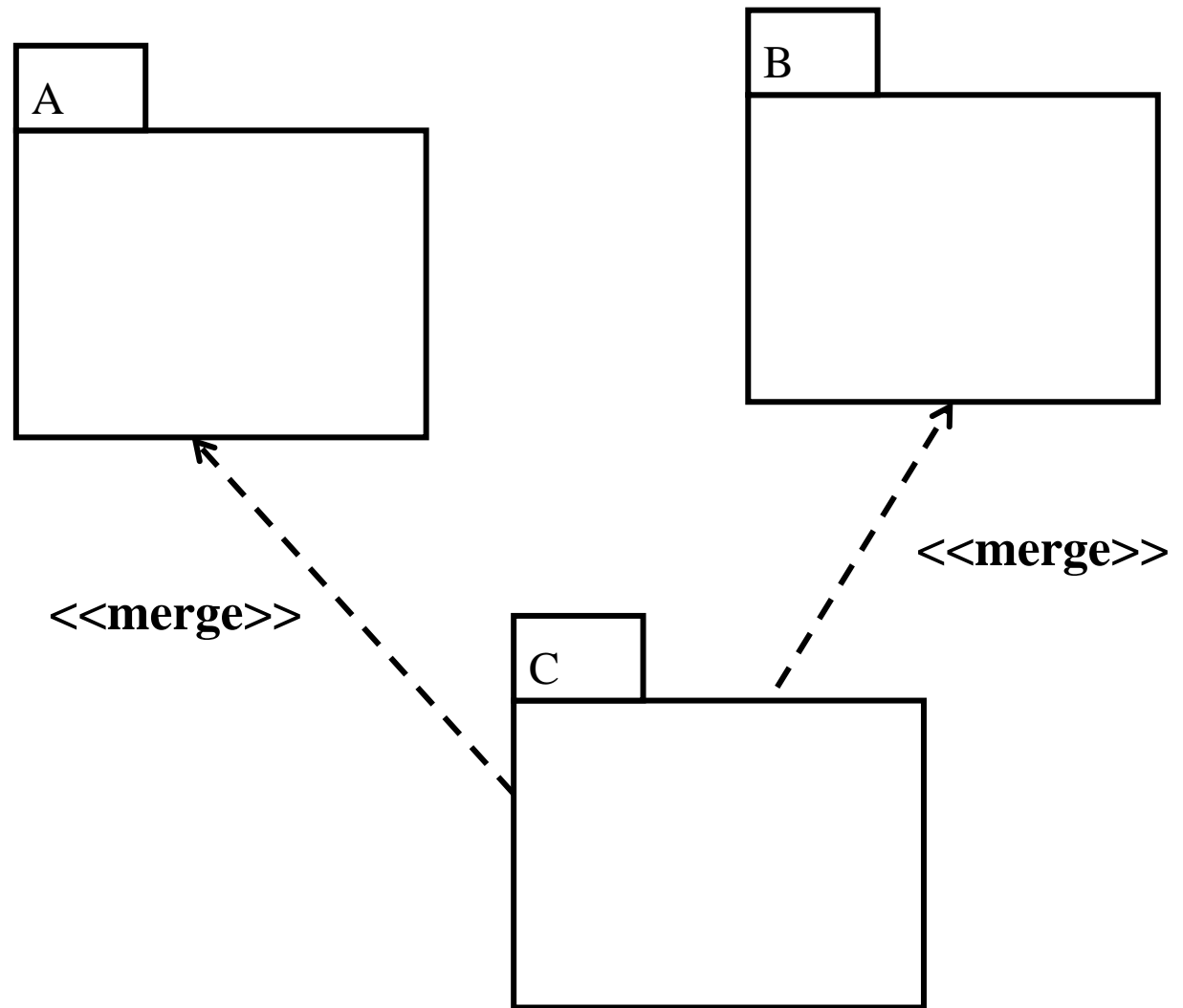
Pakiety **D** i **B**  
są  
specjalizacjami pakietu  
**C**.

Pakiet **C**  
zależy od  
pakietu **A**.

Pakiety A, B,  
C, D  
znajdują się  
w pakiecie  
**E**.



# Scalanie pakietów

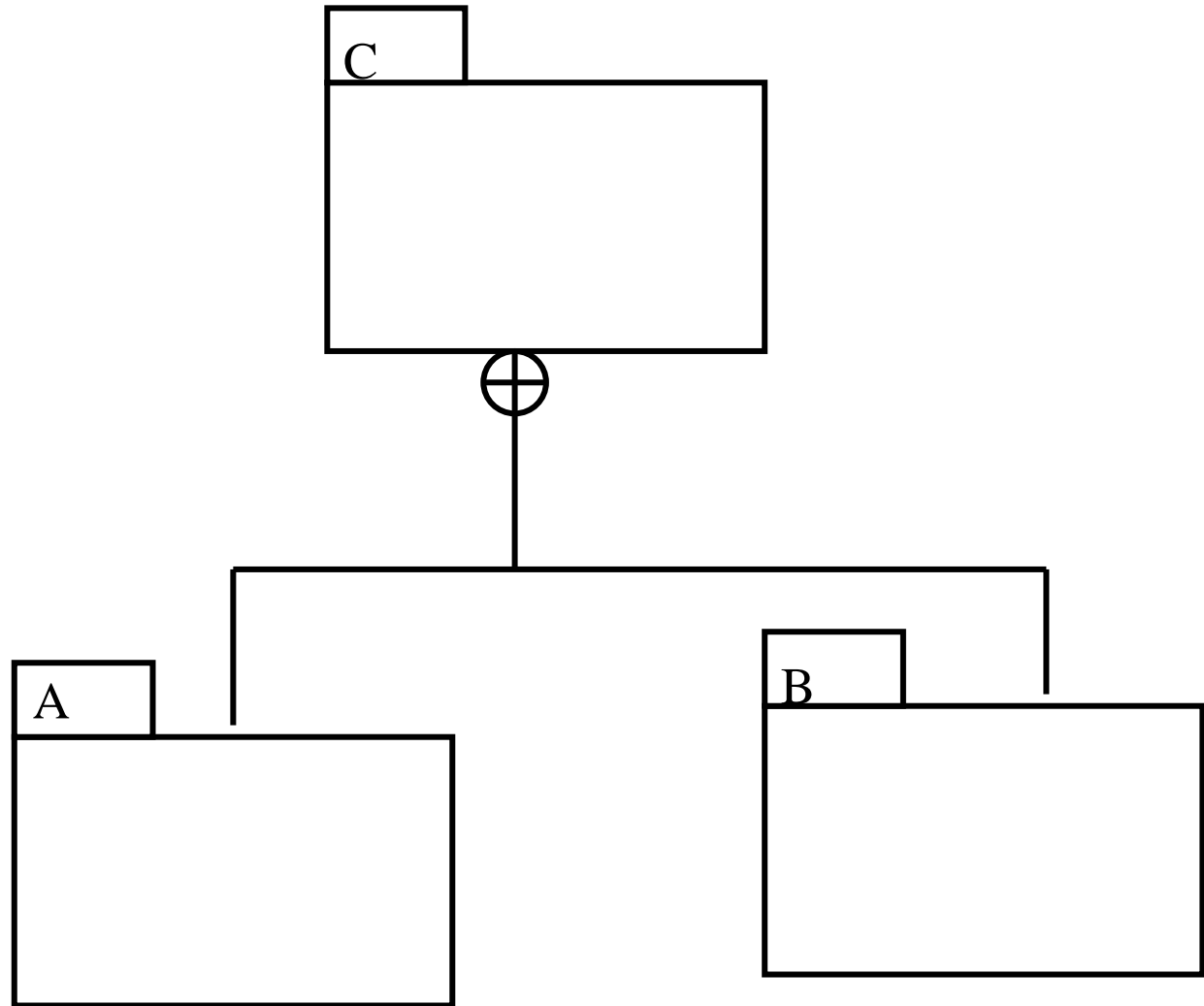


# stereotypy w diagramie pakietów

- **<<system>>** domyślny pakiet obejmujący cały system
- **<<subsystem>>** niezależna część systemu
- **<<facade>>** pakiet opisujący interfejs innego pakietu
- **<<stub>>** reprezentant (symulator) innego pakietu
- **<<framework>>** zrab - pakiet parametryzowanych wzorców architektonicznych
- **<< modelLibrary>>** pakiet grupujący elementy, które mają być użytkowane przez inne pakiety



# Alternatywna notacja zawierania pakietów



# Proces tworzenia diagramu pakietów

- identyfikacja i nazwanie pakietów
- zakwalifikowanie pakietów (*subsystem, framework, model*)
- określenie zagnieżdżeń
- określenie zależności
- wyspecyfikowanie zależności (*import, access, merge*)