

## Diagramy pakietów

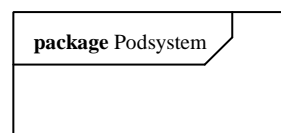
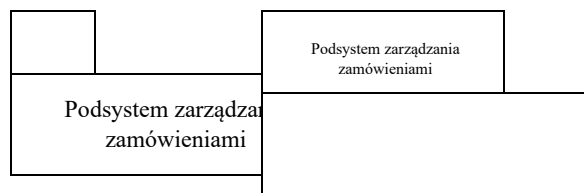
Diagram pakietów służy do tego, by uporządkować strukturę zależności w systemie, który ma bardzo wiele klas, przypadków użycia itp. Przyjmujemy, że pakiet zawiera w sobie wiele elementów, które opisują zadania w miarę dobrze określone. Na diagramie umieszczamy pakiety i wskazujemy na zależności między nimi.

Diagram pakietów:

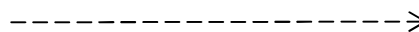
- w miarę wzrostu wielkości modelowanego systemu, rośnie liczba wykorzystywanych elementów (klas, interfejsów, komponentów...),
- można grupować modelowane byty w pakietach,
- dobrze zaprojektowane pakiety składają się z podobnych znaczeniowo i razem zmieniających się bytów. Są luźno powiązane ze sobą, ale silnie spójne wewnętrznie, • dostęp do zawartości pakietów jest kontrolowany,
- nazwę pakietu umieszcza się wewnątrz symbolu pakietu (w zakładce: jeśli wewnątrz pakietu zamieszczono wiele elementów, w nagłówku ramy zastosowanej do opisu pakietu).

Podstawowe definicje:

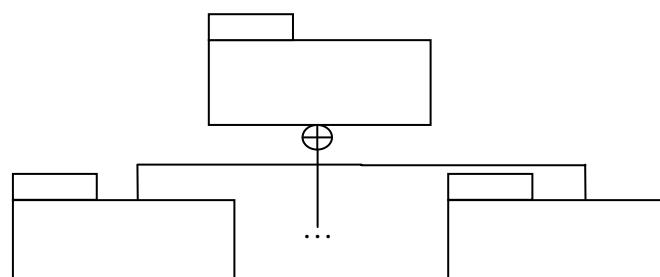
- pakiety - organizowanie elementów w grupy, punktem wyjścia w tworzeniu diagramu jest identyfikacja poszczególnych pakietów



- zależności



- zagnieżdżenie pakietów

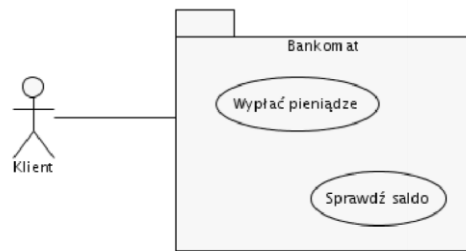


Zasady przy zastosowaniu notacji ramy:

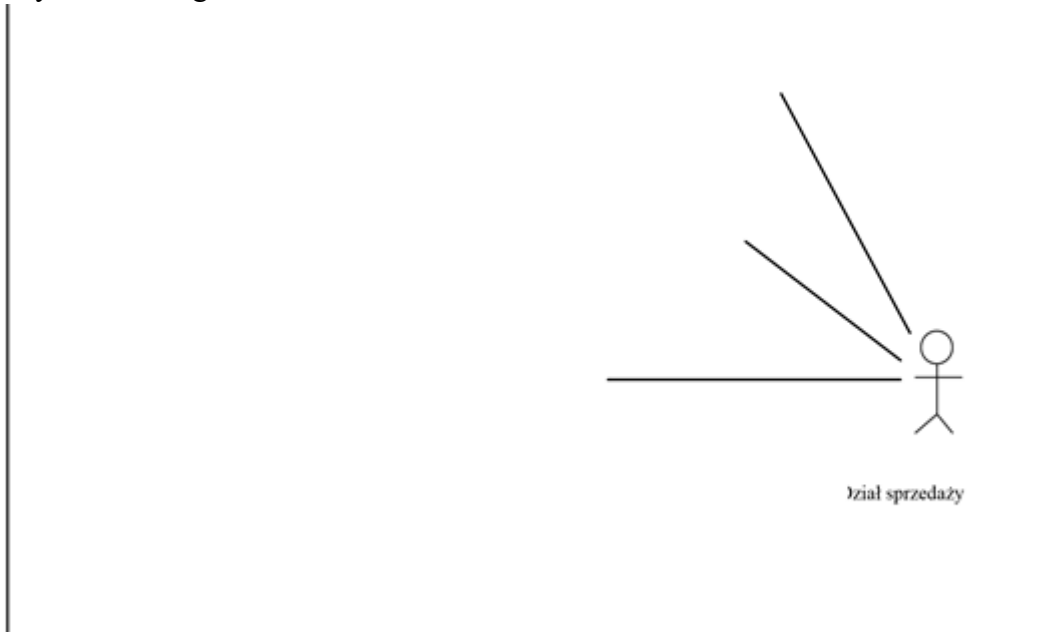
- pojedynczym elementom przypisuje się wyróżnik **package**
- kompletnym diagramom pakietów odpowiednio wyróżnik pd (j.ang.) lub pkt (j.pol.)

Składnikami pakietu mogą być różne elementy:

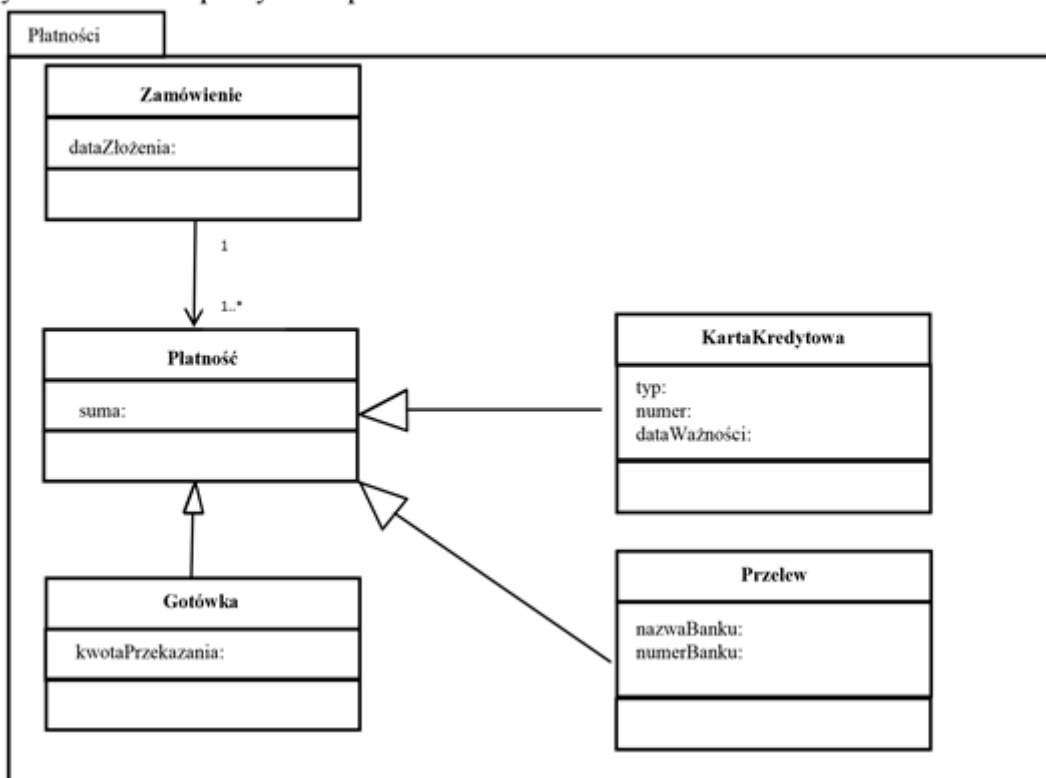
- klasy
- interfejsy
- operacje
- przypadki użycia
- diagramy
- inne pakiety
- komponenty
- ....
- lecz najczęściej są to klasy



Przykład: obsługa hurtowni



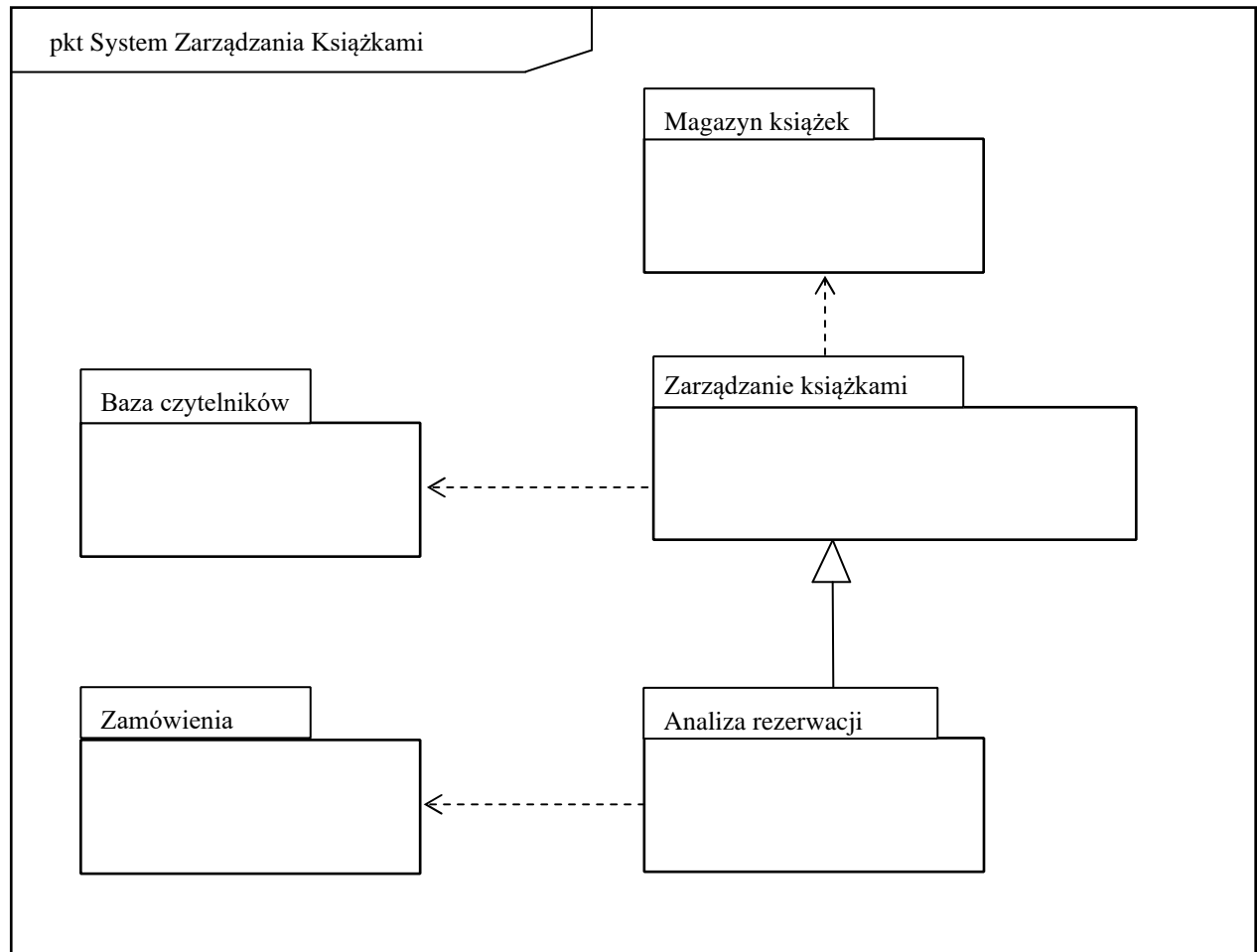
Przykład: struktura podsystemu płatności



## Zależność

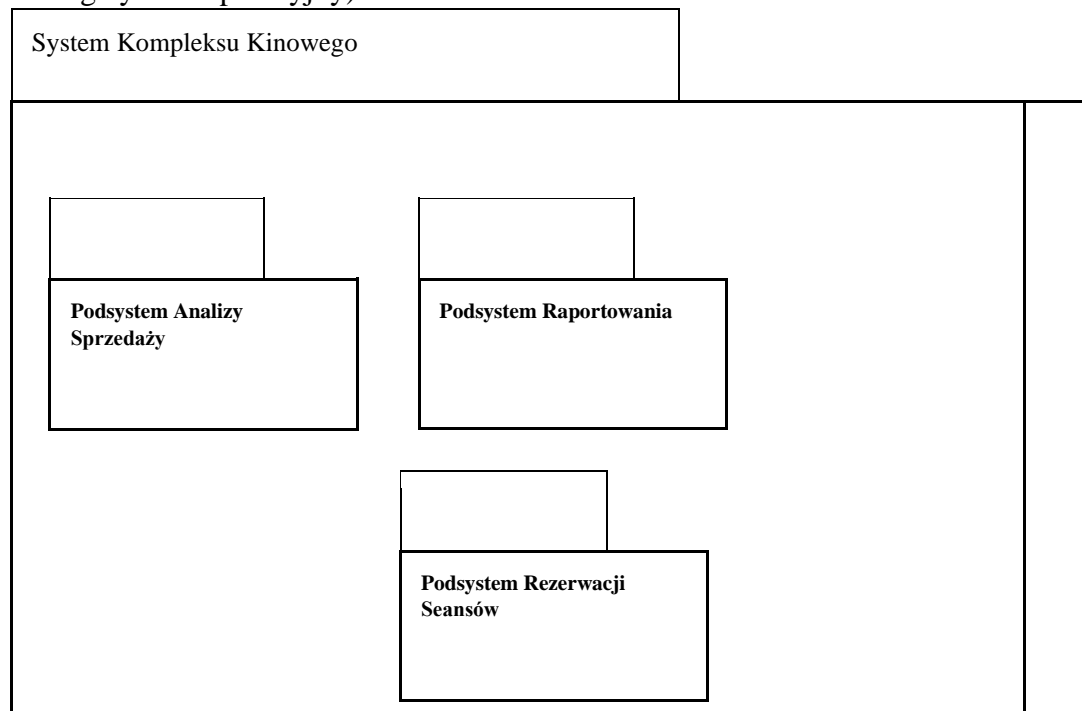
Zależności to sposób łączenia pakietów ze sobą. Dzięki tym związkom powstają diagramy pakietów. Przerywana linia prezentuje zależności między pakietami.

### Przykład: system wypożyczania książek

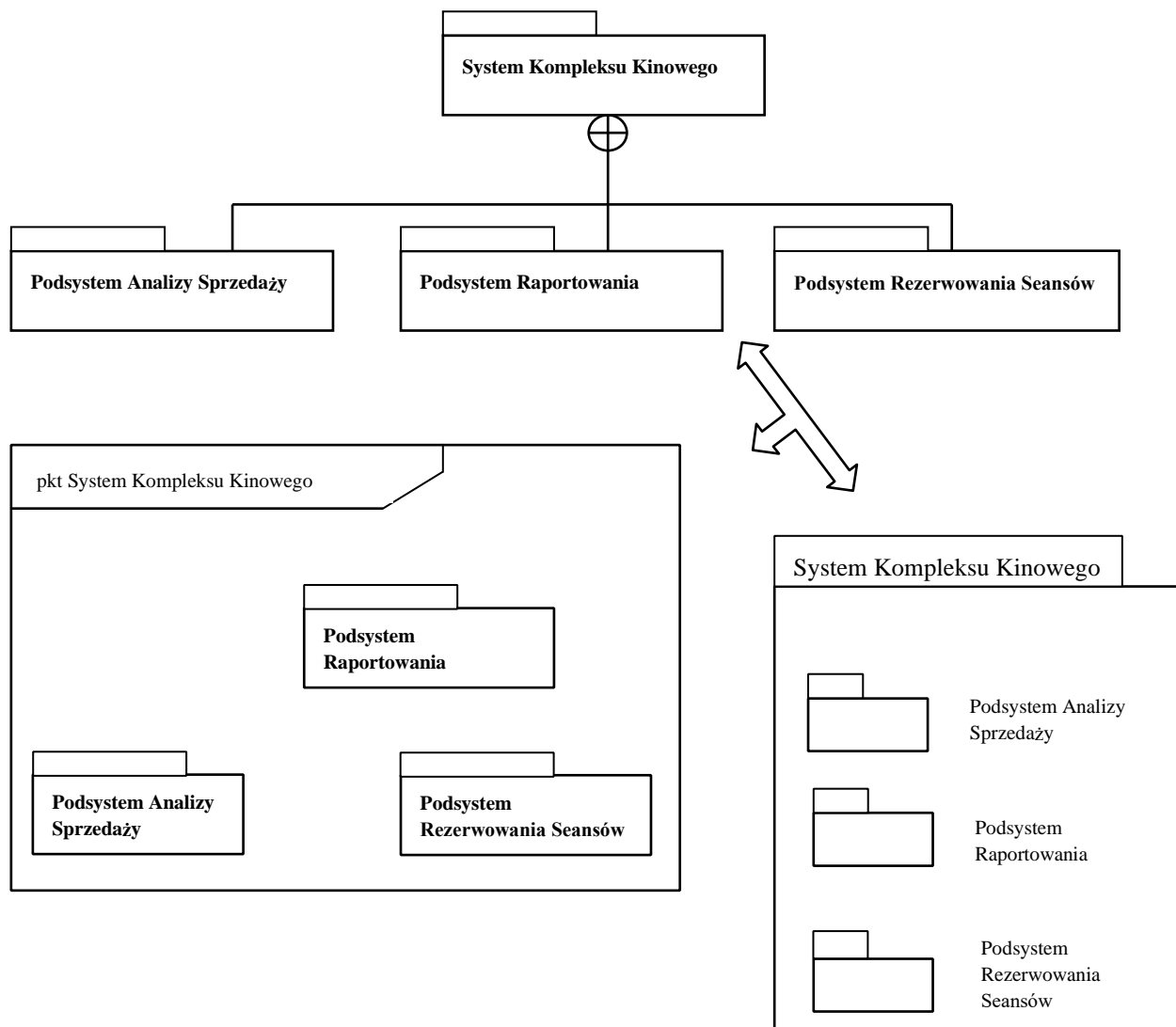


## Zagnieżdżenie pakietów

Naturalnym sposobem organizowania pakietów jest ich zagnieżdżanie (np. plik-folder, katalog-system operacyjny).



Pakiety mogą mieć charakter wielopoziomowy. Alternatywne sposoby dokumentowania zagnieżdżenia pakietów przedstawiono poniżej.



### Zaawansowane składniki diagramu

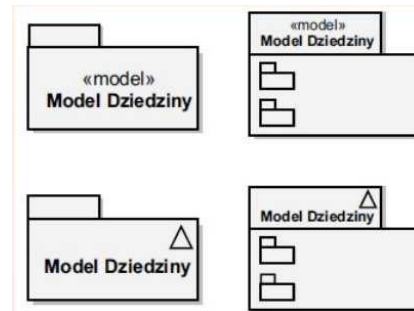
Pakiety jak i zależności różnią się w swej funkcjonalności dlatego w diagramach mogą one być **stereotypowane**:

- pakietów: stereotypy modelu, podsystemu, szablonu, bibliotek klas,
- zależności.

## Stereotypy pakietów

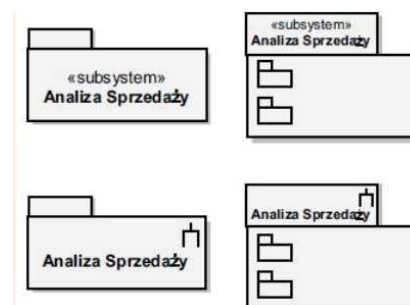
### MODEL

Stereotyp `<<model>>` opisuje pakiety będące modelami, czyli odwzorowaniami, abstrakcjami rzeczywistości. Modele te koncentrują się na całościowym przedstawieniu systemu z określonego punktu widzenia, np. użytkownika, menedżera czy też analityka. Typowe stereotypowane pakiety modeli reprezentują modele przypadków użycia, biznesowe bądź analityczne. Pakiet ten można też stereotypować graficznie za pomocą ikony.



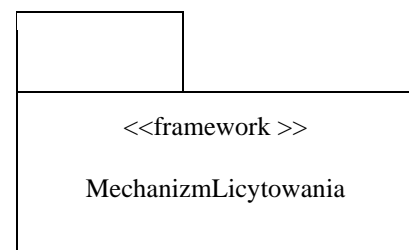
### PODSYSTEM

Stereotyp `<<subsystem>>` jest szczególnie użyteczny w modelowaniu systemów zdekomponowanych na hierarchicznie uporządkowane pakiety. Alternatywnym graficznym stereotypem podsystemu jest ikona.



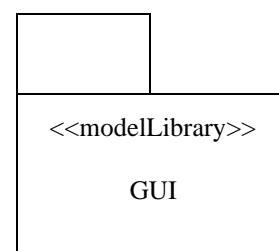
### ZRĄB

Stereotyp `<<framework>>` opisuje pakiet będący wzorcem architektonicznym, czyli elastycznym szablonem rozwiązań problemów z danej dziedziny. Zręby typowo obejmują klasy, wzorce i szablony. Zręb specyficzny dla określonego typu zastosowań nazywany jest zrębem zastosowań.



### BIBLIOTEKA KLAS

Stereotyp `<<modelLibrary>>` opisuje pakiet grupujący elementy systemu informatycznego, które z założenia mają być użytkowane przez inne pakiety. Pakiet taki jest analogią biblioteki klas w obiektowym języku programowania.





**Ograniczenia** - istnieje możliwość odwoływania się do wybranych klasyfikatorów pakietu docelowego wyspecyfikowanych z wykorzystaniem ograniczenia. Ograniczenie to jest umieszczane po nazwie pakietu i alternatywnie wobec stereotypowanego związku zależności.

Ma ono następującą składnię:

```
{import<nazwa-ścieżkowa>}
```

```
{access< nazwa-ścieżkowa>}
```

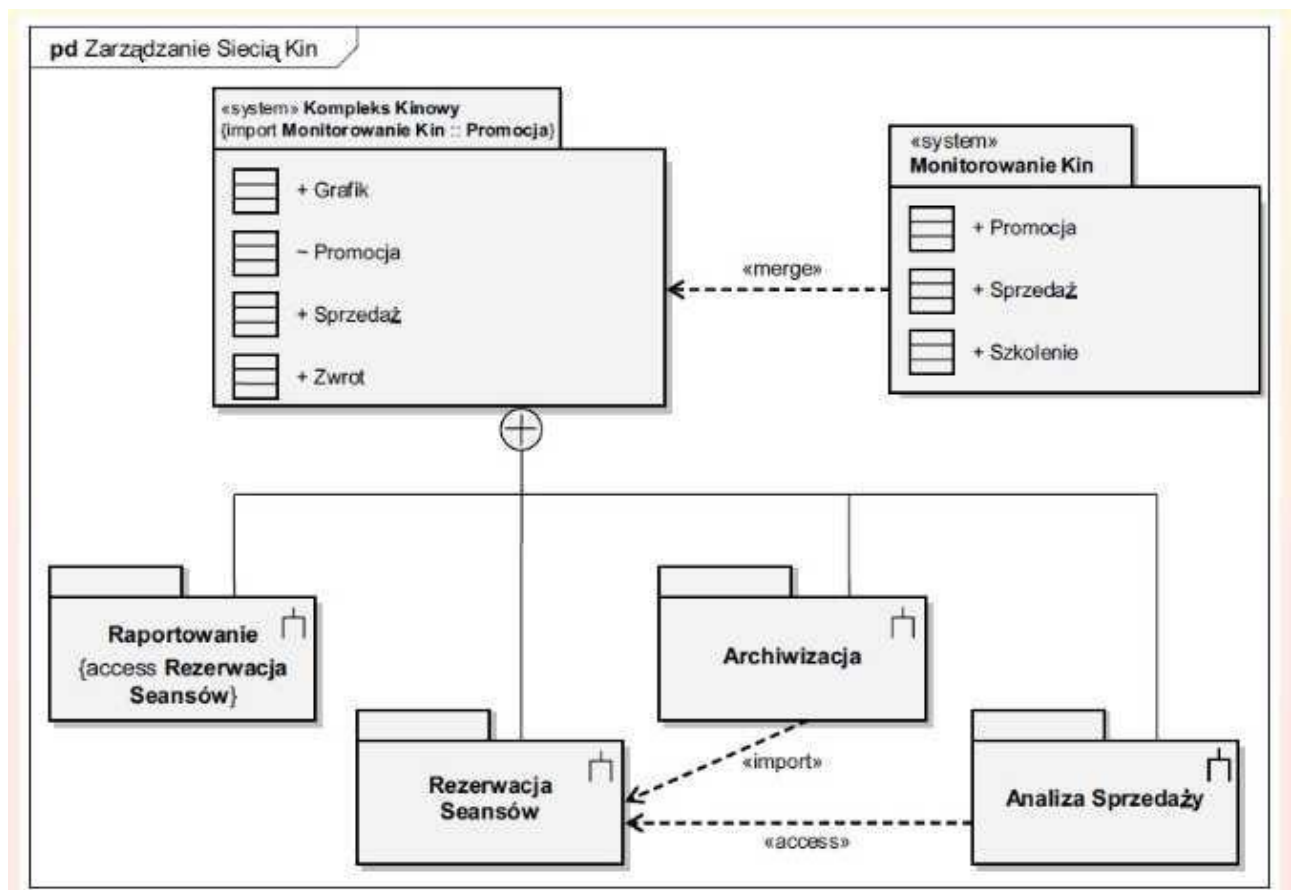
```
{merge <nazwa-ścieżkowa>}
```

**Przykład:** Zarządzanie siecią kin

Kompleks kinowy i jego podsystemy znajduje się na serwerach poszczególnych kompleksów kinowych. Monitorowanie kin – aplikacja zainstalowana w centrali firmy.

System: Kompleks kinowy

Podsystem: raportowanie, rezerwacja seansów, archiwizacja, analiza sprzedaży



**Proces tworzenia diagramu pakietu:**

1. identyfikacja i nazwanie pakietów
2. zakwalifikowanie zidentyfikowanych pakietów z zastosowaniem odpowiednich stereotypów graficznych lub tekstowych:  
subsystem, framework, model



3. określenie zagnieżdżeń pakietów
4. powiązanie pakietów z wykorzystaniem związku zależności 5.  
wyspecyfikowanie zależności poprzez nadanie im stosownych
6. stereotypów: import, merge, Access.

Język UML 2.0 w modelowaniu systemów informatycznych, S. Wrycza, B. Marcinkowski, K. Wyrzykowski, Helion