



Modelowanie Systemu Informatycznego

prezentacja 7

Modelowanie struktury – diagram pakietów, diagram obiektów i diagram struktur złożonych

wersja 1.0

dr inż. Paweł Głuchowski

Wydział Informatyki i Telekomunikacji, Politechnika Wrocławska

Treść prezentacji

1. Diagram pakietów
2. Diagram obiektów
3. Diagram struktur złożonych

1

Diagram pakietów

Diagram pakietów /package diagram/

- Modeluje logiczne uporządkowanie innych elementów:
 - gdzie ten element się znajduje,
 - w jakiej relacji jest ten element z innym elementem.
- Inne diagramy też mogą zawierać pakiety i związki między nimi, np.:
 - diagram wymagań,
 - diagram przypadków użycia,
 - diagram klas,
 - diagram komponentów.

Diagram pakietów

Relacje między pakietami

- Pakiety mogą być **zagnieżdżone**.
- **Relacja dostępu** – elementy jednego pakietu używają publicznych elementów innego pakietu:
 - **relacja «access»** – nazwy elementów wskazanego pakietu dodawane są do przestrzeni nazw używających elementów wskazującego pakietu jako prywatne;
 - **relacja «import»** – nazwy elementów wskazanego pakietu dodawane są do przestrzeni nazw używających elementów wskazującego pakietu jako publiczne.

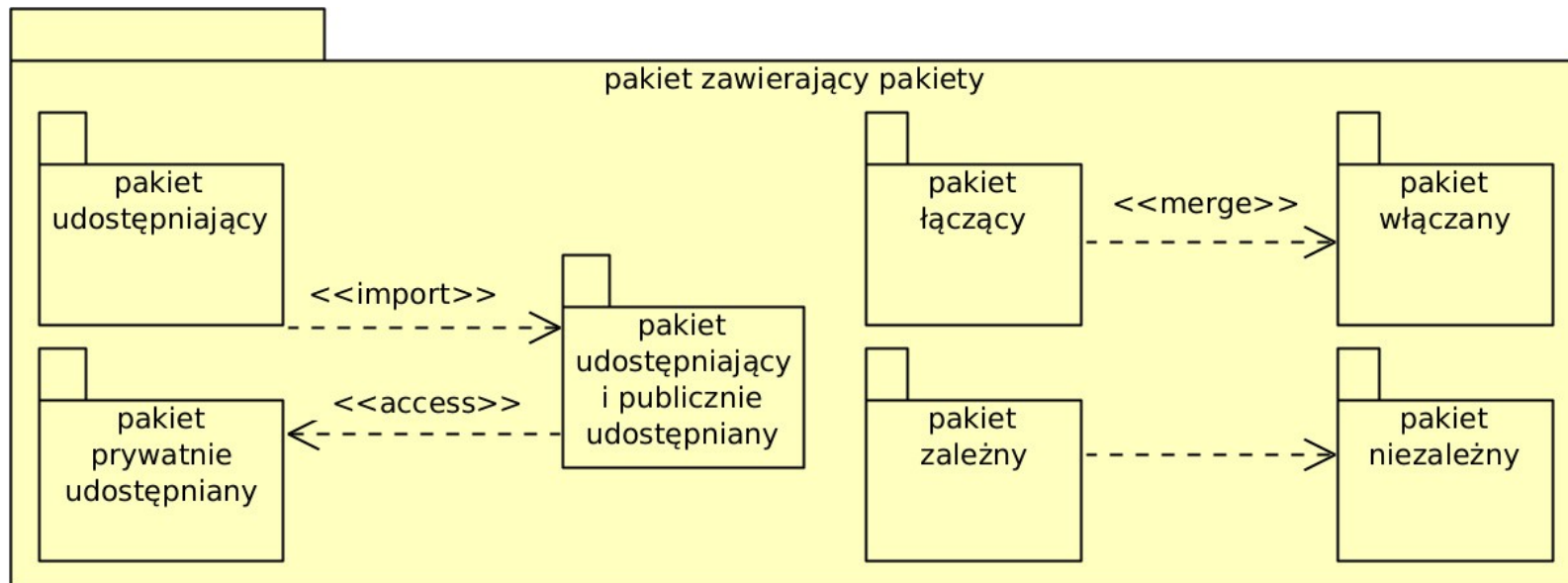


Diagram pakietów

Relacje między pakietami

- **Relacja łączenia** – elementy jednego pakietu są rozbudowywane o nazwę i funkcjonalność odpowiadające im elementy innego pakietu:
 - **relacja «merge»** – zawartość elementów wskazanego pakietu łączona jest z zawartością odpowiadających im elementów wskazującego pakietu.
- **Relacja zależności** – w innym przypadku (słabsza lub nieokreślona relacja):
 - relacja bez stereotypu «» – elementy wskazującego pakietu wymagają specyfikacji lub użycia elementów wskazanego pakietu.

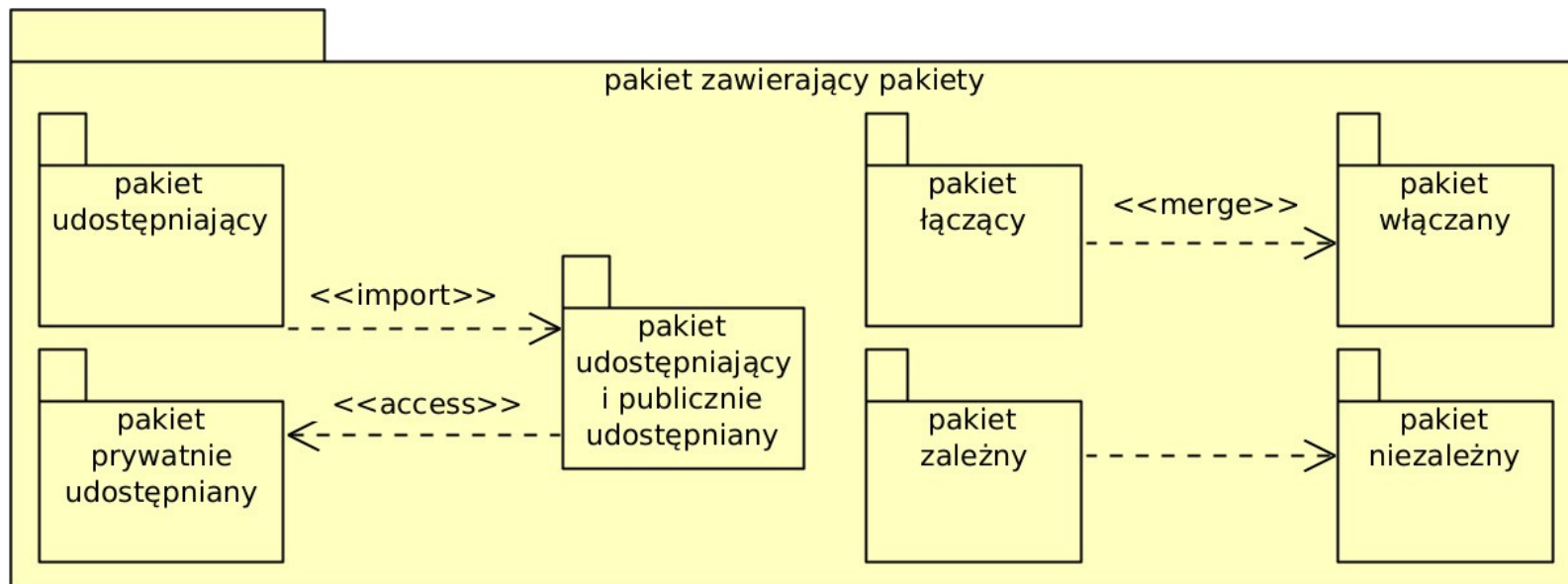


Diagram pakietów

Przykład relacji dostępu

- Założenie: klasy **A**, **B** i **C** są publiczną zawartością swoich pakietów.
- Klasa **B** z pakietu **P2**:
 - ma dostęp do klasy **A** z pakietu **P1** (**P1::A**) jako prywatną klasę **P2::A** ($P2 \rightarrow \ll\text{access}\gg \rightarrow P1$).
- Klasa **C** z pakietu **P3**:
 - nie ma dostępu do klasy **P2::A** (jest prywatna),
 - ma dostęp do klasy **B** z pakietu **P2** (**P2::B**) jako publiczną klasę **P3::B** ($P3 \rightarrow \ll\text{import}\gg \rightarrow P2$).
- Klasa **D** z pakietu **P4**:
 - ma dostęp do klas **P3::B** i **P3::C** (są publiczne).

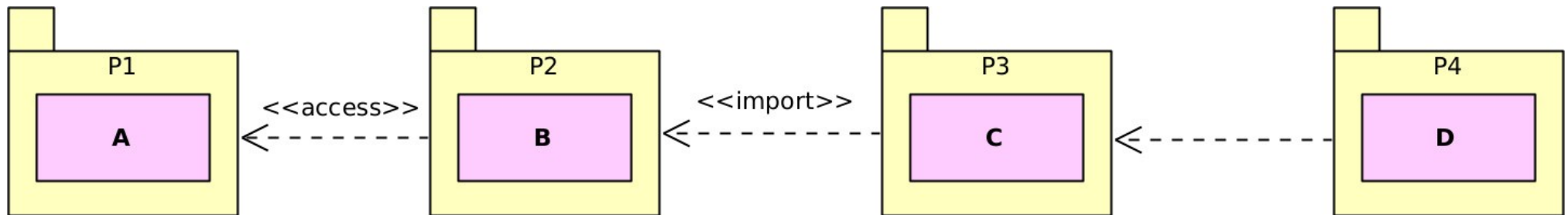
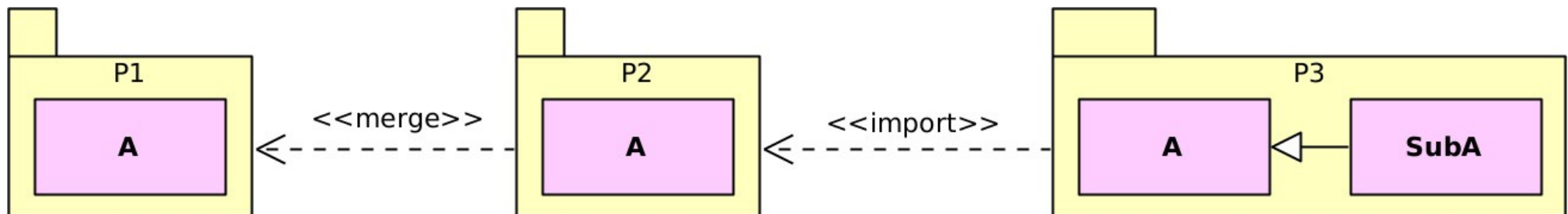


Diagram pakietów

Przykład relacji łączenia i dostępu

na podst. [Unified Modeling Language \(UML\)](#)

- Pakiet **P2** definiuje nadbudowę klasy **A**, która jest zdefiniowana w pakiecie **P1** ($P2 \rightarrow \llmerge\rright \rightarrow P1$).
 - **P2** włącza do siebie zawartość **P1** (włącza **P1::A** do **P2::A**).
- Pakiet **P3** definiuje klasę **SubA** – podklasę klasy **A** z pakietu **P2** (uogólnienie i $P3 \rightarrow \llimport\rright \rightarrow P1$).
 - **A** w pakiecie **P3** (**P3::A**) to **A** z pakietu **P2** (**P2::A**).
 - **A** w pakiecie **P2** (**P2::A**) to wynik włączenia **P1::A** do **P2::A**, a nie sama nadbudowa klasy **A** (**P2::A**).

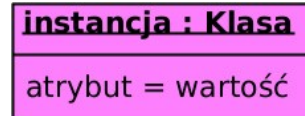


2

Diagram obiektów

Diagram obiektów /object diagram/

- Modeluje instancje klas i relacje między nimi.
- **Obiekt** – instancja klasy:
 - nazwana lub anonimowa (tylko nazwa klasy),
 - kompletna lub częściowa (tylko wybrane atrybuty):
 - atrybuty mają konkretne wartości.



- Nazwa obiektu jest podkreślona:

- typowa składnia:

```
<instance> ::= [<name>] : [<package>::]*<classifiername>
```

- pełna składnia:

```
<instance> ::= {<name> [ '/' <rolename> ] | '/' <rolename> }  
[ ':' [<package>::]*<classifiername> [ ',' <classifiername> ]*
```

<name> – nazwa obiektu;

<rolename> – nazwa roli pełnionej przez obiekt;

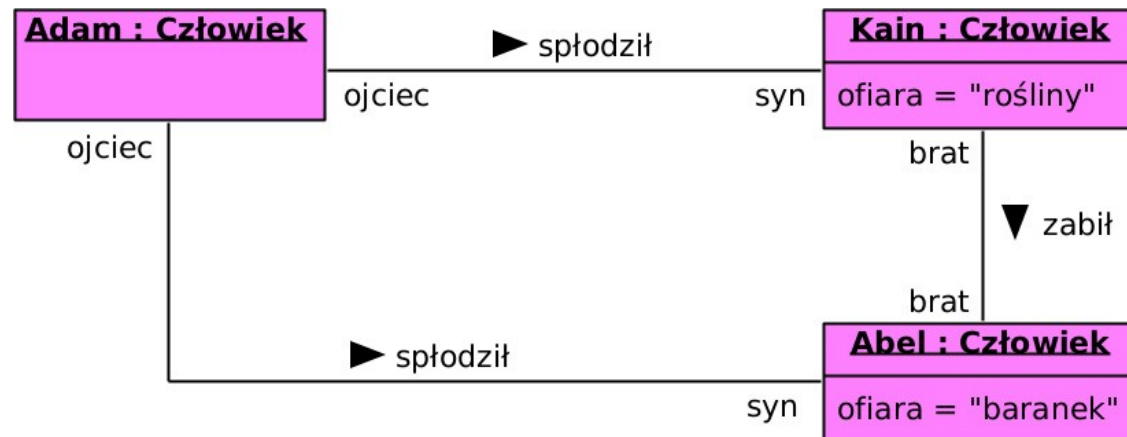
<package> – pakiet;

<classifiername> – nazwa klasy, której obiekt jest instancją.

Diagram obiektów

- Związki między obiektami:
 - zależność,
 - asocjacja 1 do 1.

- **Przykład:**



3

Diagram struktur złożonych

Diagram struktur złożonych

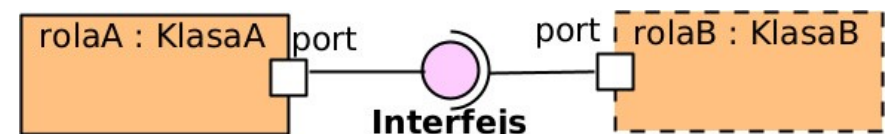
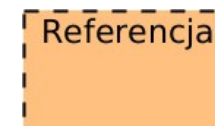
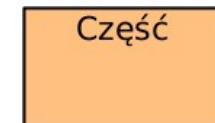
/composite structure diagram/

- Modeluje strukturę złożoną hierarchicznie:
 - wewnętrzną strukturę klasyfikatora (np. klasy),
 - współdziałanie.
- Pakiet grupuje klasy w czasie kompilacji, a struktura złożona grupuje klasy w czasie wykonania programu.
- Główne elementy struktury:
 - **część** /part/ struktury (m.in. jako **rola** /role/ klasy),
 - **referencja** do części /referenced part/,
 - **interfejs** (i **port**) dostępu od i do struktury i jej części,
 - **łącznik** /connector/ – relacja między częściami i części z interfejsem,
 - **klasyfikator** (np. klasa),
 - **współdziałanie** /collaboration/ ról w osiągnięciu celu struktury,
 - **użycie współdziałania** /collaboration use/.

Diagram struktur złożonych

Wewnętrzna struktura /internal structure/

- Budowa klasyfikatora (np. klasy) zawarta w jego przedziale „wewnętrzna struktura”:
 - jego **części** (gdy klasyfikator jest w relacji agregacji lub kompozycji z częścią),
 - **referencje** do NIE jego części (gdy klasyfikator jest w relacji asocjacji z częścią),
 - **role** pełnione przez części,
 - **łączniki** – relacje między nimi i z interfejsami,
 - **liczba** ról i części.
- **Interfejs** modeluje funkcjonalność struktury:
 - bez szczegółów realizacji,
 - w postaci „lizaka” (gdy zapewniany) lub „łapki” (gdy wymagany),
 - może być zaczepiony w porcie.



Wewnętrzna struktura

- **Składnia nazwy części i referencji do części:**

- typowa składnia:

```
<part> ::= [<name>] : [<package>::]*<classifiername>
```

- pełna składnia:

```
<part> ::= {<name> [ '/' <rolename> ] | '/' <rolename>}  
[ ':' [<package>::]*<classifiername> [ ',' <classifiername> ]*]
```

<name> – nazwa części;

<rolename> – nazwa roli pełnionej przez część;

<package> – pakiet;

<classifiername> – nazwa klasy, której część jest instancją.

- **Składnia nazwy łącznika:**

```
<connector> ::= ( [<name>] ':' <associationname> ) |  
                ( [<name>] ':' <associationclassname> ) | [<name>]
```

<name> – nazwa łącznika;

<associationname> – nazwa asocjacji, której łącznik jest instancją;

<associationclassname> – nazwa klasy asocjacyjnej, której łącznik jest instancją.

Diagram struktur złożonych

Przykład złożonej wewnętrznej struktury

- Części klasyfikatora **Android**:
Głowa (a w niej: **Zmysły** (słuch i wzrok)), 2 **Nogi** i 2 **Ręce**.
- **Android** może mieć na sobie (nie w sobie) dowolnie wiele **Ubrań**.
- **Głowa** łączy się z **Nogami**, **Rękami** i otoczeniem **Androida** przez port **mózg**.
- **Android** implementuje interfejs (aby nim sterować) **słuchaj** przez port **ucho**.
- **Android** wymaga interfejsów (aby wpływał na swoje otoczenie) **ruszaj się** przez port **mięśnie** i **patrz** przez port **oko**.

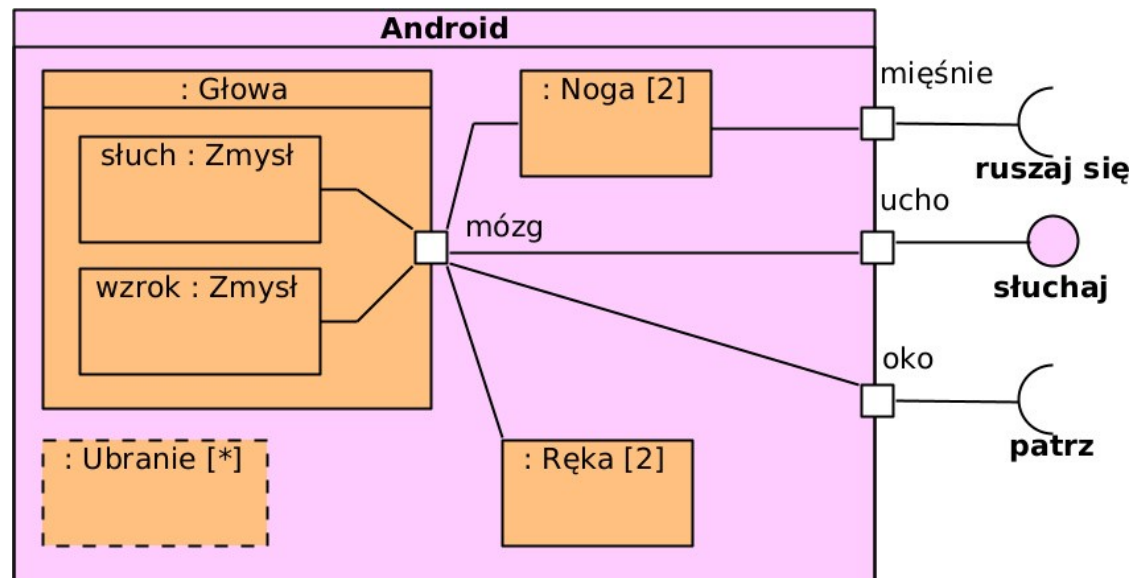


Diagram struktur złożonych

Przykład diagramu wewnętrznej struktury i porównanie z diagramem klas.

na podst. [Unified Modeling Language \(UML\)](#)

- Diagram klas pokazuje:
 - **Samochód** posiada 2 **Koła** w roli **tylne**.
 - **Samochód** związany jest (nie posiada) z 1 **Silnikiem** w roli **elektryk**.
 - Asocjacja **Oś** łączy **Koło** i **Silnik** w stosunku wiele do wielu.
- Diagram wewnętrznej struktury pokazuje (dla konkretnego **Samochodu**):
 - Częściami **Samochodu** są: 2 **Koła** w roli **tylne** i 1 **Silnik** w roli **elektryk**.
 - **Tylne** z **elektrykiem** łączy łącznik **o** (asocjacja **Oś**) w stosunku 2 do 1.

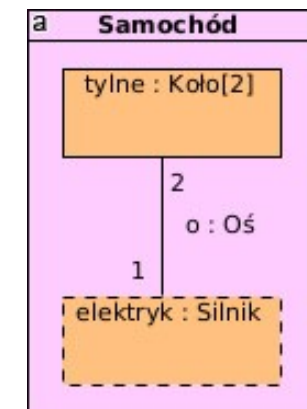
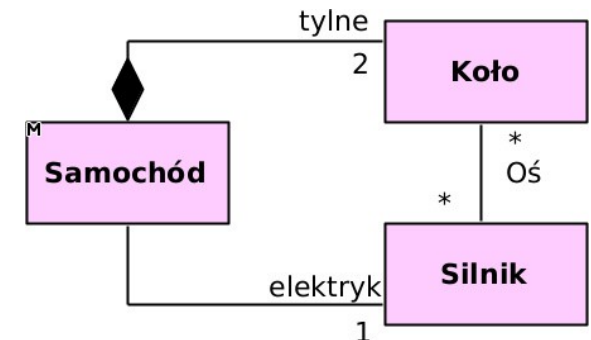


Diagram struktur złożonych

Współdziałanie /collaboration/

- **Modeluje** współdziałanie wybranych ról klas w celu osiągnięcia wspólnych celów (zadań).
- **Zawiera** części (jako **role** klas) i **łączniki** (relacje między nimi):
 - TYLKO te, które są potrzebne do osiągnięcia celów współdziałania;
 - pozostałe (jeśli istnieją) nie są pokazane.
- Klasa może pełnić różne role i uczestniczyć w wielu współdziałaniach.
- **Reprezentacja:** owal z przerywaną krawędzią, nazwą i wewnętrzną strukturą (lub zwykły klasyfikator ze stereotypem «*collaboration*»).
- **Użycie współdziałania** (owal z przerywaną krawędzią i nazwą):
 - w relacji asocjacji z klasami pełniącymi role we współdziałaniu (na diagramie struktur złożonych),
 - w relacji zależności z klasami pełniącymi role we współdziałaniu (na diagramie klas).

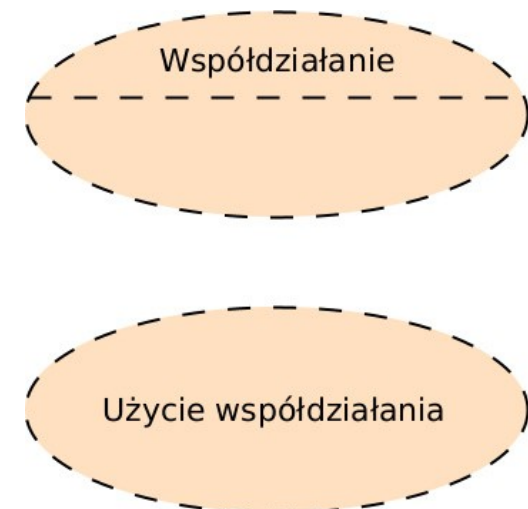


Diagram struktur złożonych

Przykład współdziałania i jego użycia

Diagram konceptualny współdziałania **Wizyta** →

- Diagram klas pokazuje:
 - **Człowiek** w roli **lekarza** jest związany z co najmniej 1 Człowiekiem w roli **pacjenta**.
 - to ogólna relacja.
- Współdziałanie **Wizyta** pokazuje:
 - **Człowiek** w roli **lekarza** jest związany z dokładnie 1 Człowiekiem w roli **pacjenta**,
 - to szczególna relacja użyta do wykonania zadań **Wizyta**.

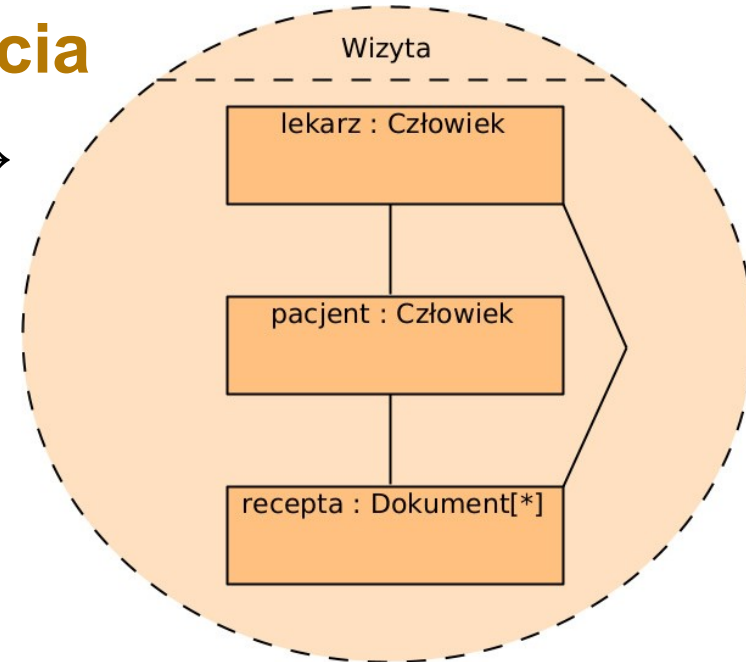
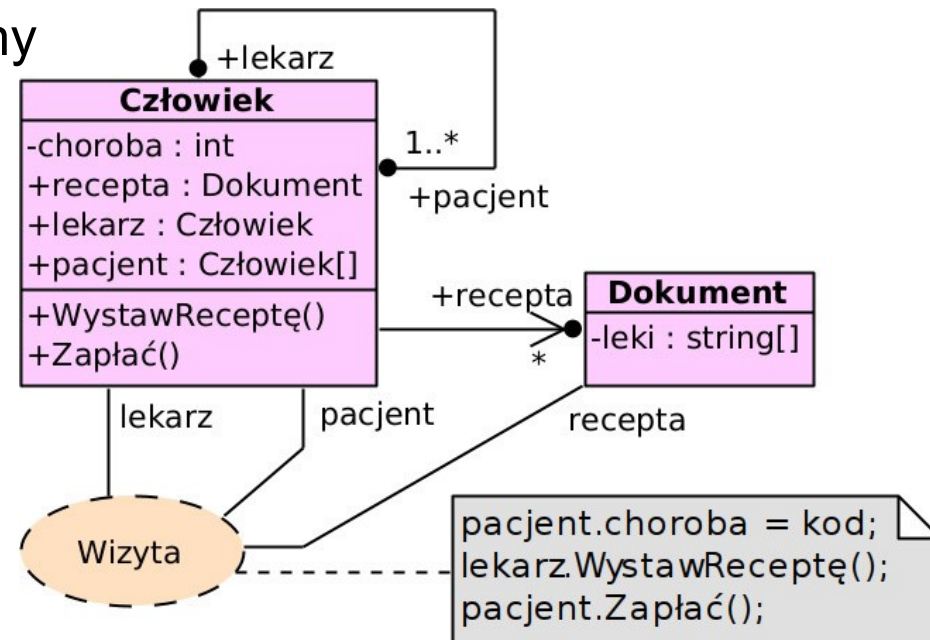


Diagram implementacyjny użycia współdziałania **Wizyta** →



Temat następnej prezentacji

Modelowanie struktury
– diagram komponentów
i diagram wdrożenia