



# Modelowanie Systemów Informatycznych (MSI)

dr inż. Mariusz Trzaska  
[mtrzaska@pjwstk.edu.pl](mailto:mtrzaska@pjwstk.edu.pl)

Wykład 6

Diagramy aktywności

<http://www.mtrzaska.com>



# Zagadnienia

## Diagramy aktywności

- Podstawowe pojęcia; notacja
- Aktywność a akcja
- Przepływy decyzyjne
- Przepływy współbieżne
- Łącznik
- Przepływ sterowania a przepływ obiektu
- Wierzchołki typu obiekt
- Akcje związane z sygnałami
- Obszar rozszerzenia
- Obszar przerwania
- Obsługa wyjątków
- Partycje
- Podsumowanie diagramów aktywności



*Wykorzystano materiały z wykładu PRI autorstwa dr inż. Ewy Stemposz oraz prof. Kazimierza Subiety*

# Diagramy aktywności (czynności)

- ✓ **Diagramy aktywności** nie posiadają wyraźnego pierwowzoru w poprzednich pracach Jacobsona, Boocha i Rumbaugh'a. Łącząc idee pochodzące z trzech źródeł: diagramów zdarzeń J. Odell'a, technik modelowania stanów i sieci Petriego są szczególnie użyteczne przy modelowaniu przepływów operacji.
- ✓ **Graf aktywności** – Wierzchołki grafu aktywności odpowiadają stanom wyróżnialnym w trakcie przetwarzania i noszą nazwę aktywności. Aktywność może być interpretowana różnie, w zależności od perspektywy: jako zadanie do wykonania i to zarówno przez człowieka, jak i przez komputer (z perspektywy pojęciowej) czy też np. jako pojedyncza metoda (z perspektywy projektowej). Przejścia między wierzchołkami (krawędzie grafu) są związane z zakończeniem przetwarzania wyspecyfikowanego dla danej aktywności.

Diagramy aktywności mogą być wykorzystywane do modelowania:

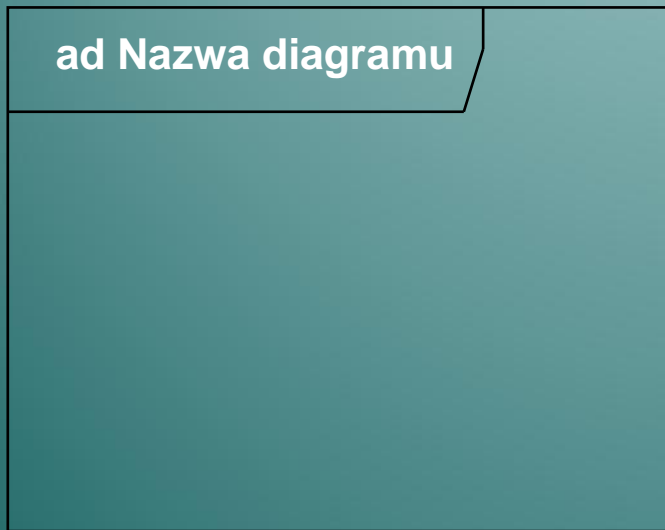
- ✓ procesów biznesowych,
- ✓ scenariuszy przypadków użycia,
- ✓ przetwarzania współbieżnego,
- ✓ pojedynczych operacji, grup operacji,
- ✓ algorytmów.

# Adaptacja notacji BNF

Symbol	Znaczenie
=	struktura danych po lewej stronie symbolu = składa się z elementów wyspecyfikowanych po stronie prawej
+	odpowiada słowu “i”; wykorzystywane do agregowania elementów
[ ... ]	definiowana struktura zawiera tylko jeden spośród elementów zawartych w nawiasach [ ]; kolejne elementy są oddzielane przecinkami
( ... )	elementy zawarte w nawiasach ( ) są opcjonalne, co oznacza, że mają 0..1 wystąpień
{ ... }	definiowana struktura zawiera od 0..* wystąpień elementu zawartego w nawiasach { }; kolejne wystąpienia są oddzielane przecinkami
* ... *	informacje zawarte między * * są traktowane jak komentarz, a więc nie stanowią elementów składowych definiowanej struktury

# Prezentowanie diagramów aktywności

<nagłówek-diagramu> = (<wyróżnik\_diagramu>) + <nazwa-diagramu> + {<parametr>}



**ad** – activity diagram

# Podstawowe pojęcia; notacja

## Podstawowe pojęcia przyjęte w UML 2.0 dla diagramów aktywności:

nazwa  
aktywności/  
akcji



aktywność; czynność (ang. activity) lub akcja (ang. action)

przepływ sterowania: oznacza zakończenie jednej aktywności/akcji i przejście do drugiej; ten sam symbol jest używany dla oznaczenia przepływu obiektu



blok decyzyjny: może rozdzielać jedno przejście na kilka alternatywnych (opatrzonych warunkami) lub łączyć kilka alternatywnych przejść w jedno przejście



sztabka synchronizująca (ang. synchronization bar): może być typu rozwidlenie (rozdzielenie jednej operacji na kilka realizowanych współbieżnie) lub typu scalenie (złączenie kilku operacji współbieżnych w jedną)



aktywność początkowa



aktywność końcowa



zakończenie przepływu

# Aktywność a akcja

**Akcja:** operacja elementarna; nie może zostać przerwana;  
czas realizowania akcji zazwyczaj nie jest brany pod uwagę

x := 100

Wylicz średnią

**Aktywność:** operacja złożona z innych operacji (złożonych lub elementarnych)

Złóż  
zamówienie

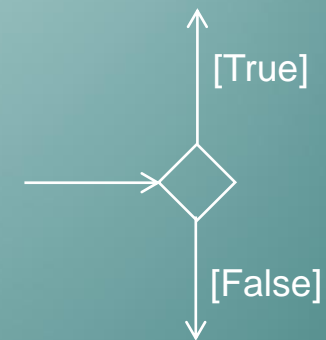
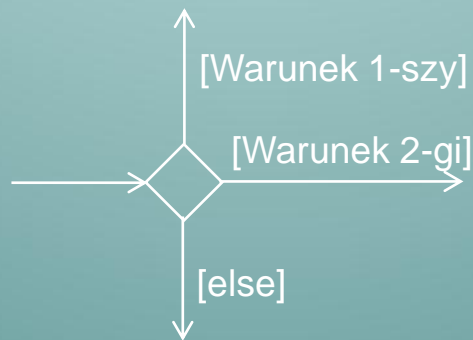
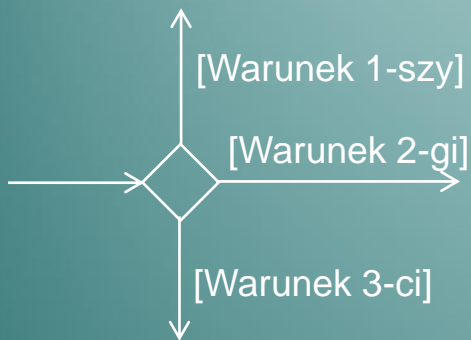
Rejestruj  
klienta

Rejestruj<sup>∞</sup>  
klienta

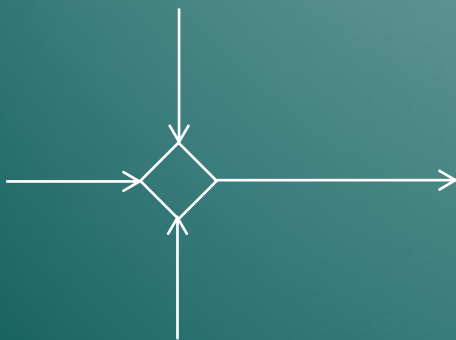
**Uwaga:** Warunek zachowania spójności diagramów wymusza konieczność tożsamości nazw: aktywności nadrzędnej i diagramu uszczegóławiającego.

# Przeptywy decyzyjne

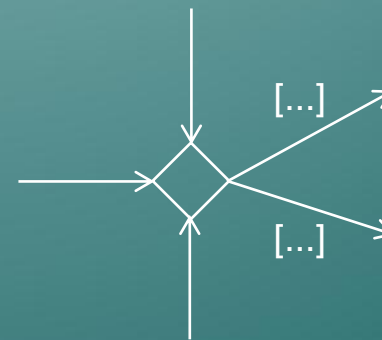
## Decyzja (ang. decision node)



## Złączenie (ang. merge node)



## Decyzja + Złączenie





# Przeptywy współbieżne

Rozwidlenie (ang. fork node)



Scalenie (ang. join node)

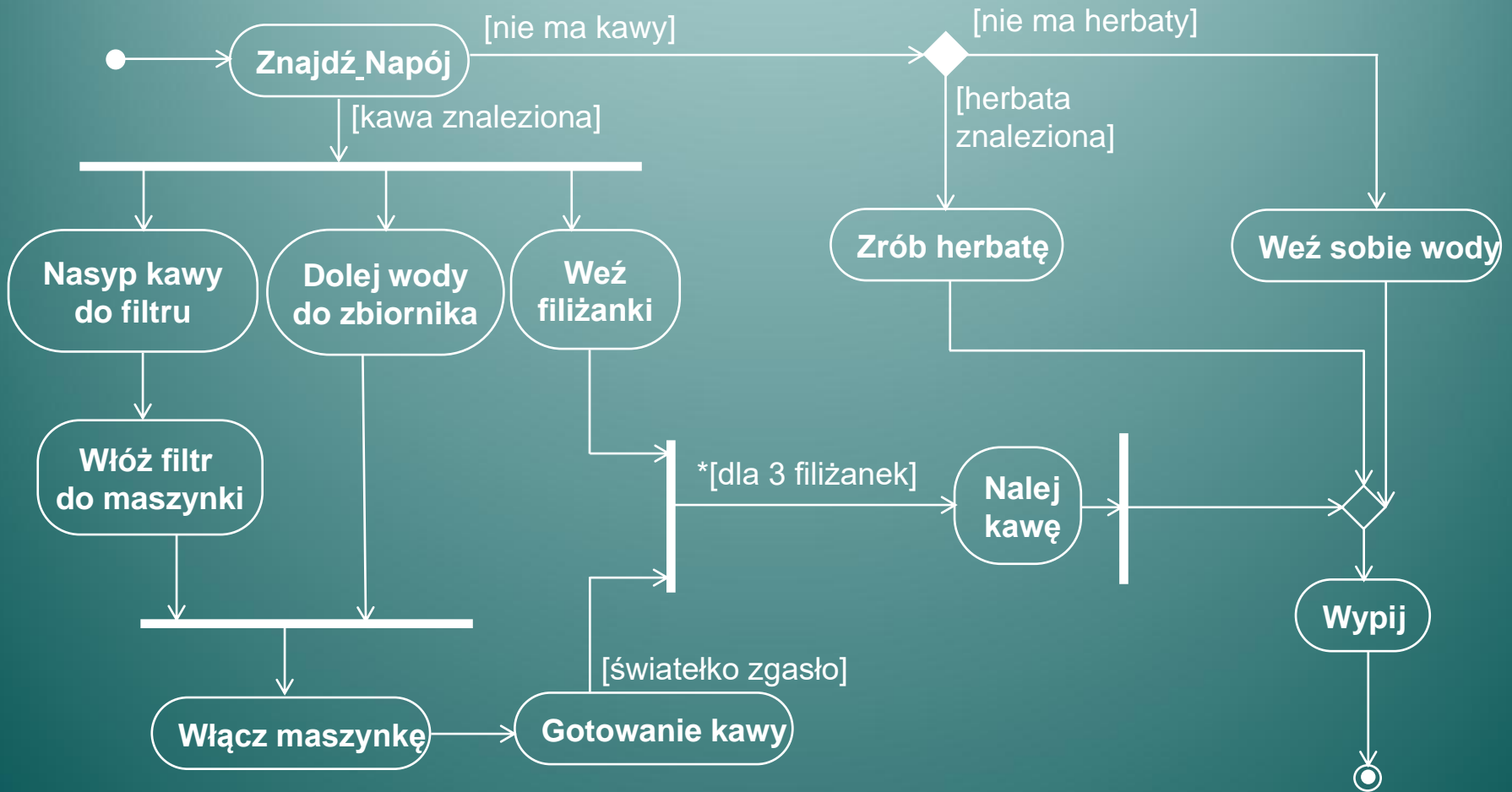


Rozwidlenie + Scalenie

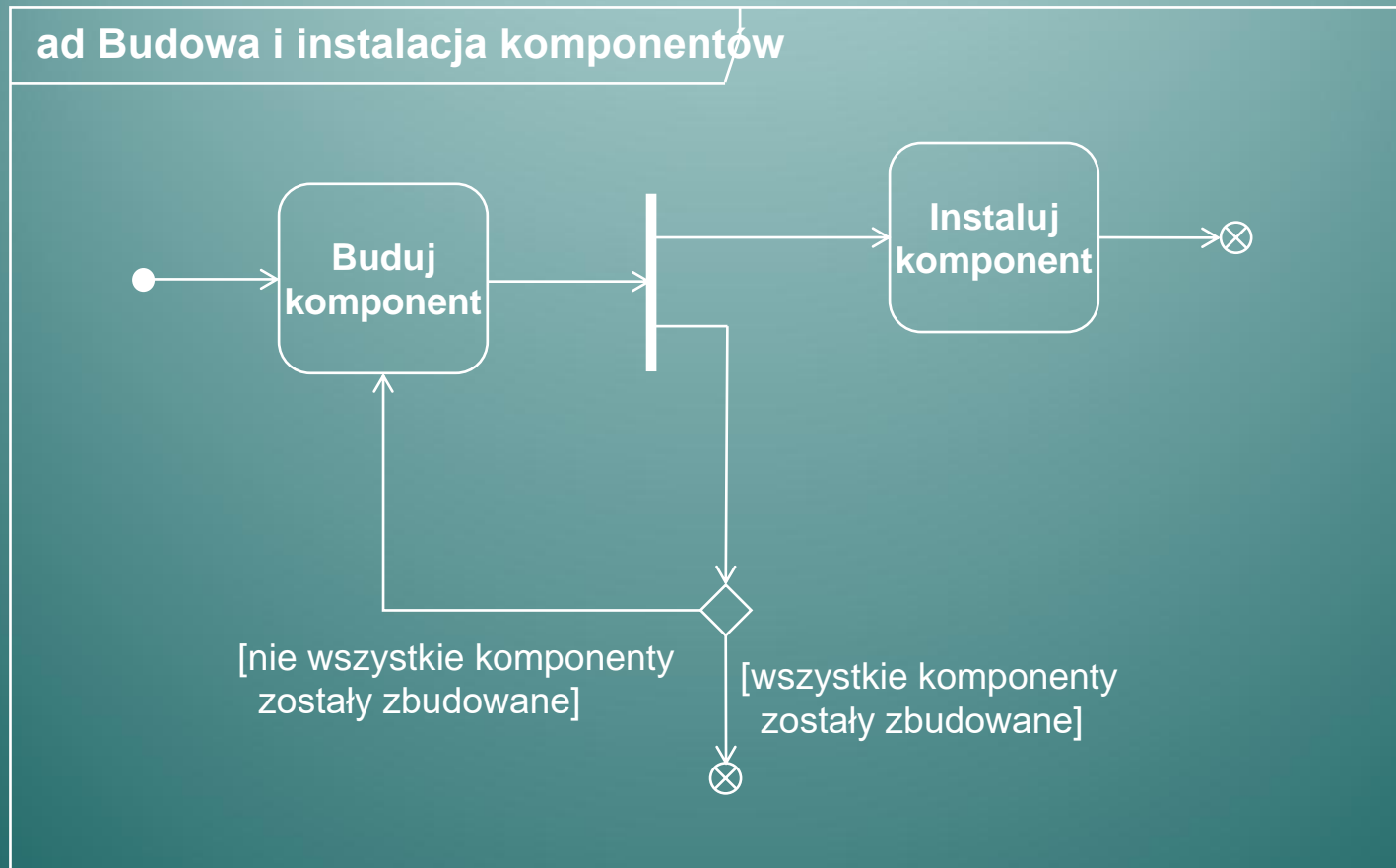


# Diagram aktywności; przykład

ad Przygotowanie Napoju



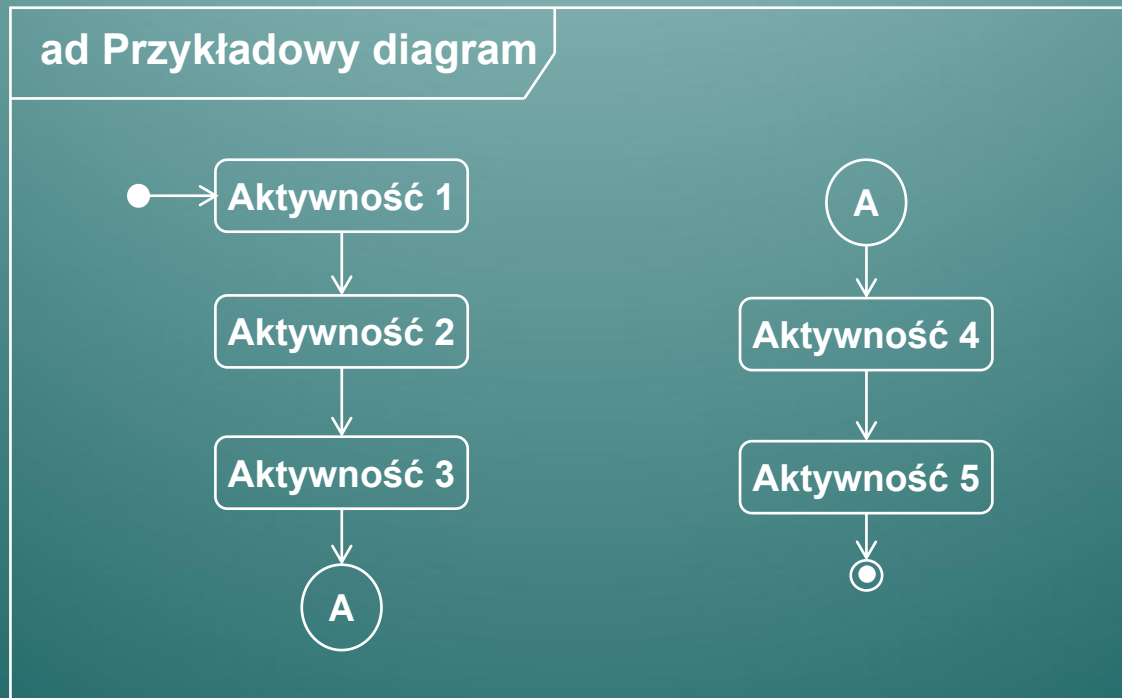
# Przykład wykorzystania zakończenia przepływu



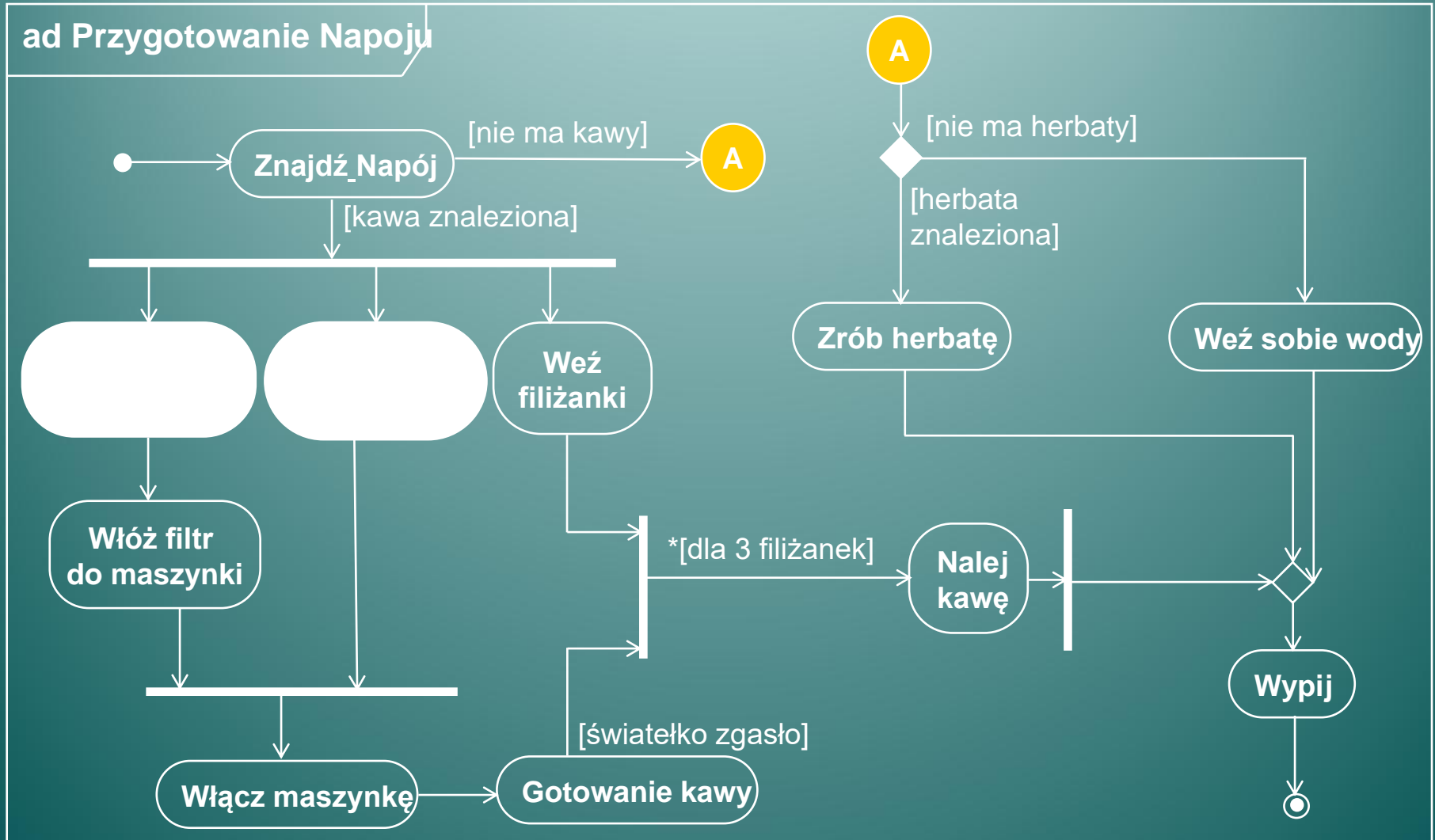
# Łącznik

Łącznik (ang. activity edge connector): umożliwia przerwanie przepływu sterowania w jednym miejscu diagramu i wznowienie go w innym:

- ✓ kolejne łączniki są zwyczajowo oznaczane kolejnymi dużymi literami alfabetu łacińskiego,
- ✓ przepływ sterowania może zostać wznowiony na innym diagramie aktywności.



# Wykorzystanie łącznika; przykład



# Przeptyw sterowania a przepływ obiektu (1)

Przeptyw sterowania  
(ang. control flow edge)



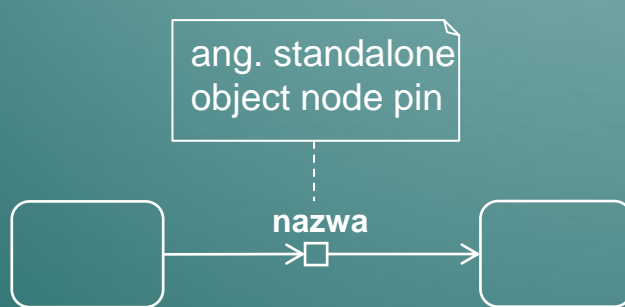
wierzchołek typu obiekt  
(ang. object node)

Przeptyw obiektu  
(ang. object flow edge)



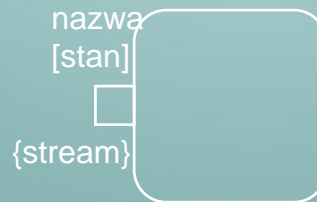
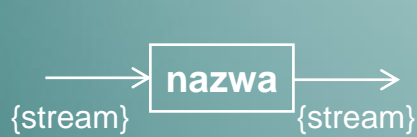
ang. object  
node pin

ang. standalone  
object node pin



Przeptyw obiektu – jako kategoria modelowania – został wprowadzony w celu ilustrowania przeptywu obiektów (lub danych innego rodzaju) pomiędzy aktywnościami/akcjami.

# Przeptyw sterowania a przepływ obiektu (2)



{nonstream} jest domyślne

Alternatywna notacja dla oznaczenia przepływu strumienia danych: zamiast ograniczenia {stream} zamalowano grotę strzałek i wierzchołki we/we operacji



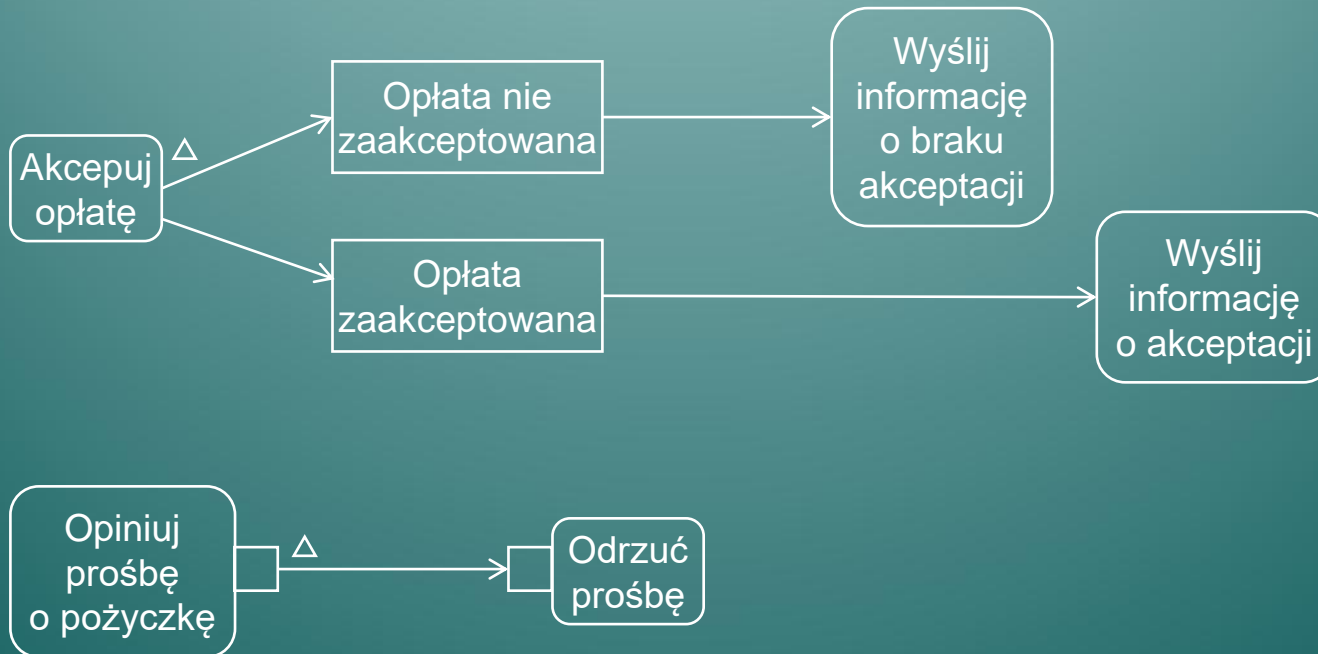
# Przeptyw sterowania a przepływ obiektu (3)

symbol trójkąta jest wykorzystywany dla oznaczenia przepływu danych wykorzystywanych w obsłudze „nie rutynowych” sytuacji

sytuacja rutynowa



sytuacja nierutynowa





# Notacja dla wierzchołka typu obiekt (1)

**Wierzchołek typu obiekt:** oznacza taki wierzchołek w grafie aktywności, w którym dostępne jest wystąpienie/wystąpienia określonego klasyfikatora;



gdzie: *nazwa* jest nazwą wierzchołka

- ✓ w czasie run-time'u wierzchołek typu obiekt może „zawierać” tylko takie dane,
  - które są zgodne z typem danych określonych dla wierzchołka
- ✓ jeśli typ danych dla wierzchołka nie został określony, wierzchołek może „zawierać” dane dowolnego typu
- ✓ wystąpienie/wystąpienia mogą mieć wyspecyfikowane stany
- ✓ jest możliwe określenie maksymalnej liczby wystąpień, które może „zawierać”
  - dany wierzchołek; ograniczenie {upperBound=2}; możliwe jest wykorzystanie \*

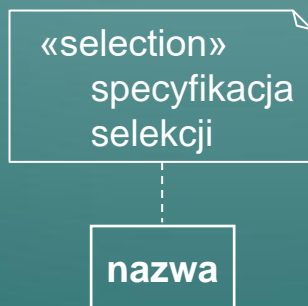
# Notacja dla wierzchołka typu obiekt (2)

## Specjalne rodzaje wierzchołków typu obiekt

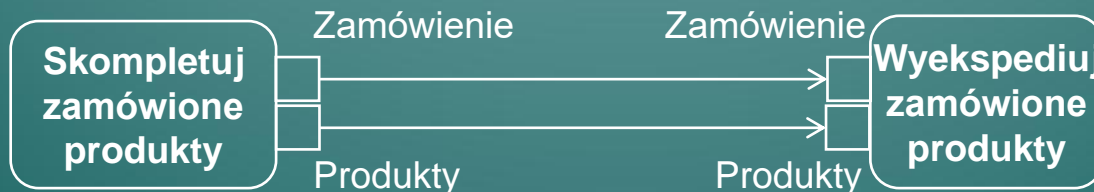
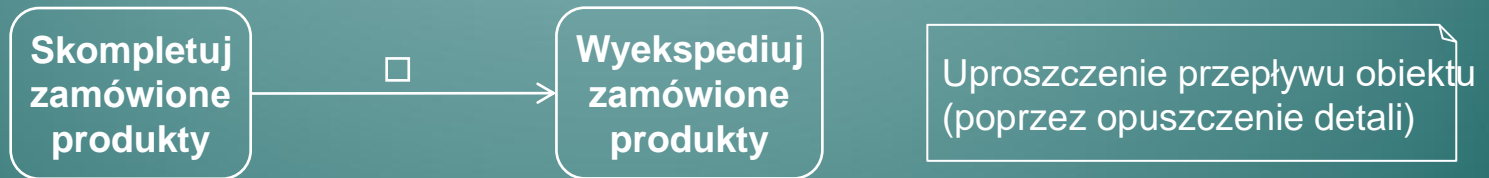


**«selection»:** zachowanie (ang. behavior) wykorzystywane w celu umożliwienia wyboru danych związanych z wierzchołkiem typu obiekt; selection jest umieszczane w polu adnotacji

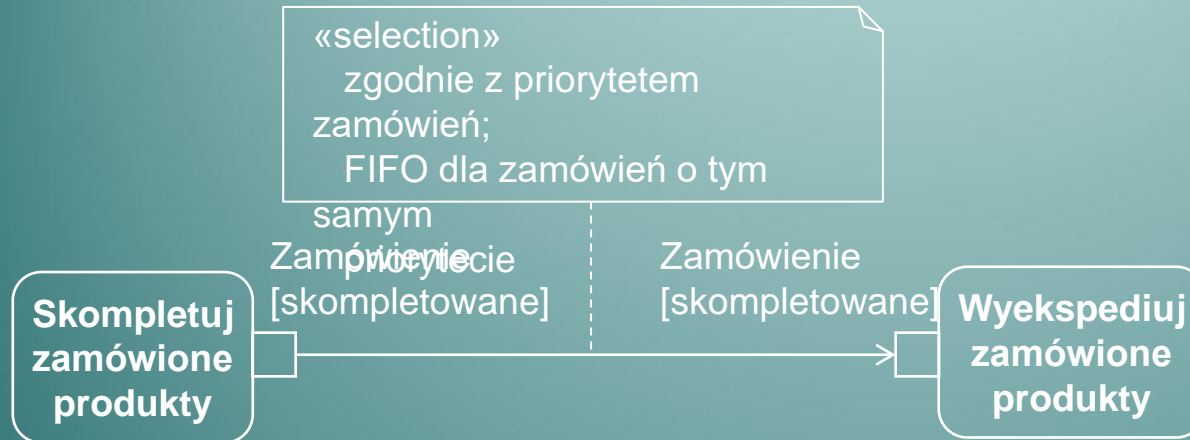
określa sposób selekcjonowania danych dla przepływów wychodzących z wierzchołka typu obiekt



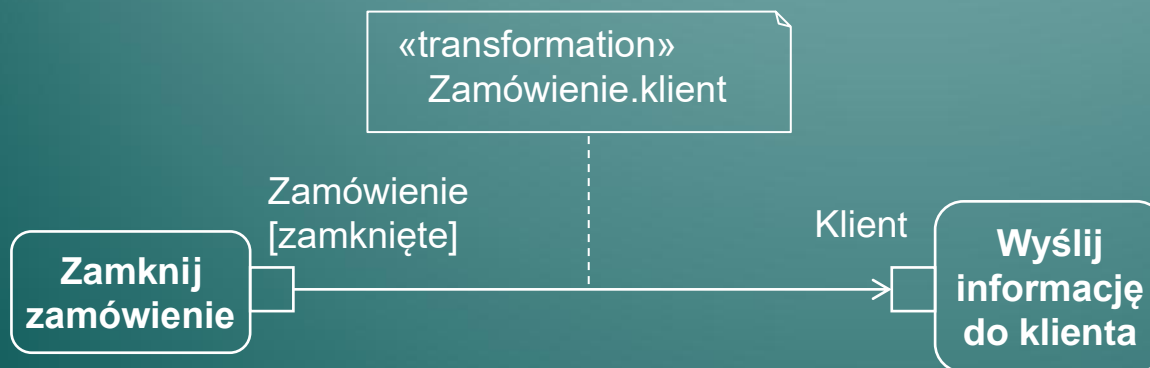
# Przeptyw obiektu; przykłady



# Specyfikacja selekcji; przykłady



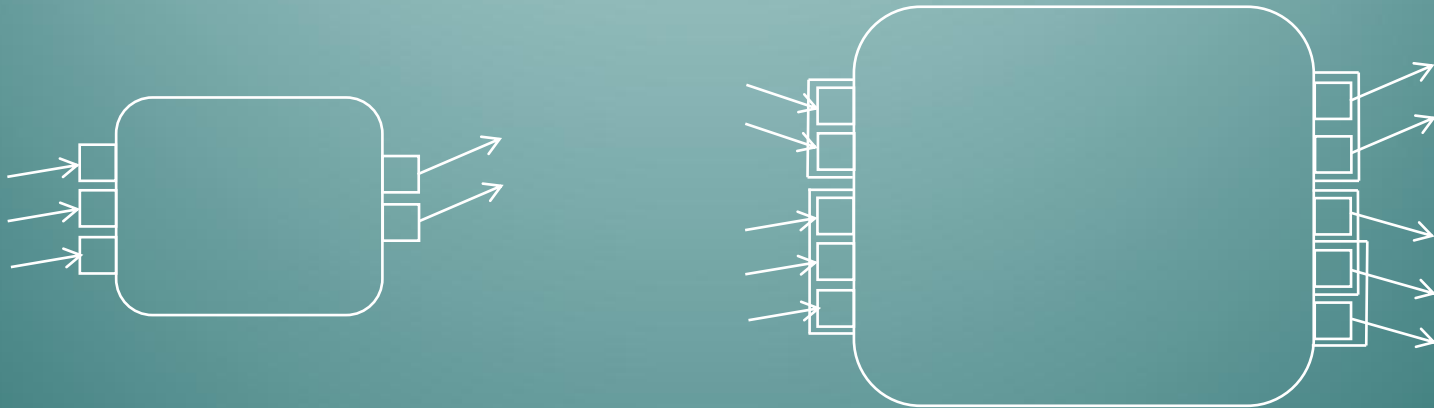
zamówione produkty będą ekspediowane w oparciu o priorytety zamówień; zamówienia o tym samym priorytecie mają być obsługiwane zgodnie z porządkiem FIFO



**transformacja** jest rodzajem selekcji; aktywność *Zamknij zamówienie* „produkuje” obiekty *Zamówienie*, podczas gdy aktywność *Wyślij informację do klienta* wymaga obiektów *Klient*; zostaną one uzyskane dzięki transformacji będącej efektem zapytania *Zamówienie.klient*

# Zbiory parametrów

**Zbiory parametrów** (ang. parameter sets): oznaczają alternatywne „podgrupy” danych we/wy dla operacji – oznaczane za pomocą obramowania wierzchołków; jeden wierzchołek typu obiekt może pojawić się w kilku podgrupach;



Domyślnie – wewnątrz grupy wierzchołków typu obiekt (stanowiących parametry we/wy dla operacji) wierzchołki są traktowane na zasadzie logicznej koniunkcji, tzn. można rozpocząć przetwarzanie specyfikowane przez operację dopiero wtedy, gdy wszystkie dane, specyfikowane przez wierzchołki wejściowe, zostały skompletowane (to samo dotyczy wierzchołków wyjściowych). Jeśli wykorzystano oznaczanie zbiorów parametrów, to przetwarzanie może rozpocząć się, gdy któraś z podgrup danych została skompletowana.

# Akcje związane z sygnałami; sygnał typu „czas”

Akcja związana z akceptacją sygnału (ang. accept event action)



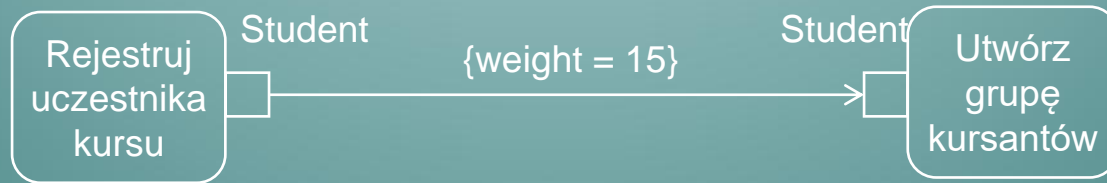
Akcja związana z wysyłaniem sygnału (ang. sent event action)



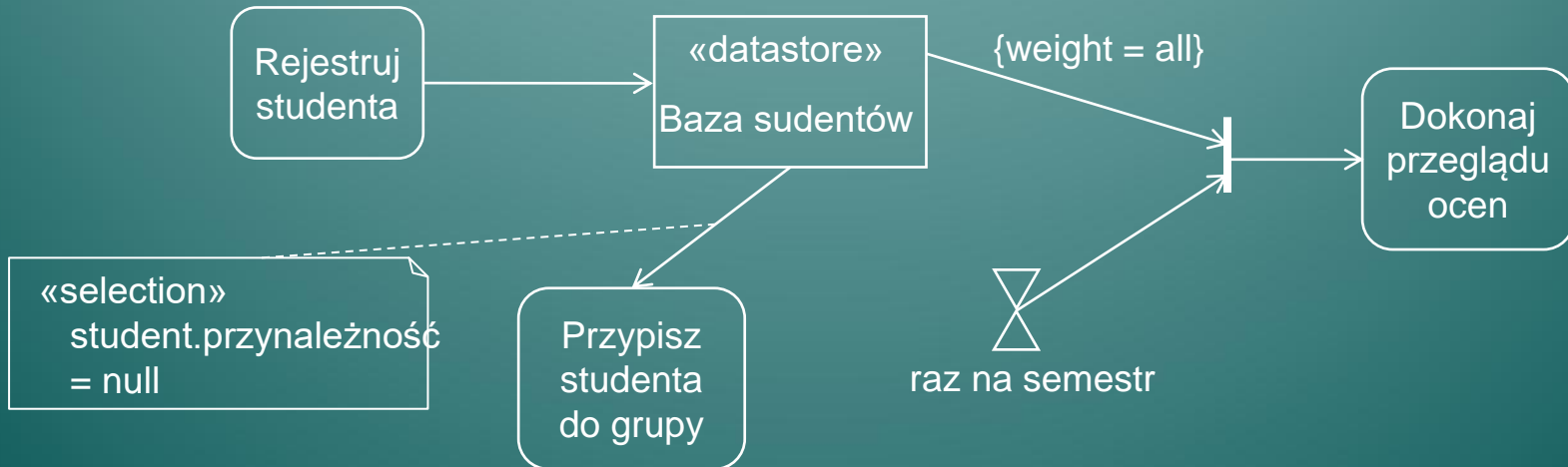
# Wagi

**Waga** (ang. weight): wskazuje minimalną liczbę obiektów, które muszą być przekazane z jednej aktywności do drugiej, aby ta druga mogła być uruchomiona. Wagę, będącą ograniczeniem, oznacza się w nawiasach klamrowych w pobliżu przepływu obiektu. Waga = 0 oznacza, że nie nałożono ograniczenia na liczbę przekazywanych obiektów.

## Przykład 1

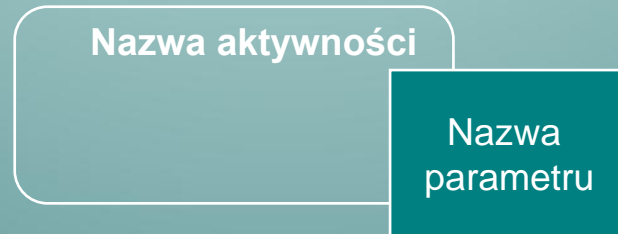


## Przykład 2

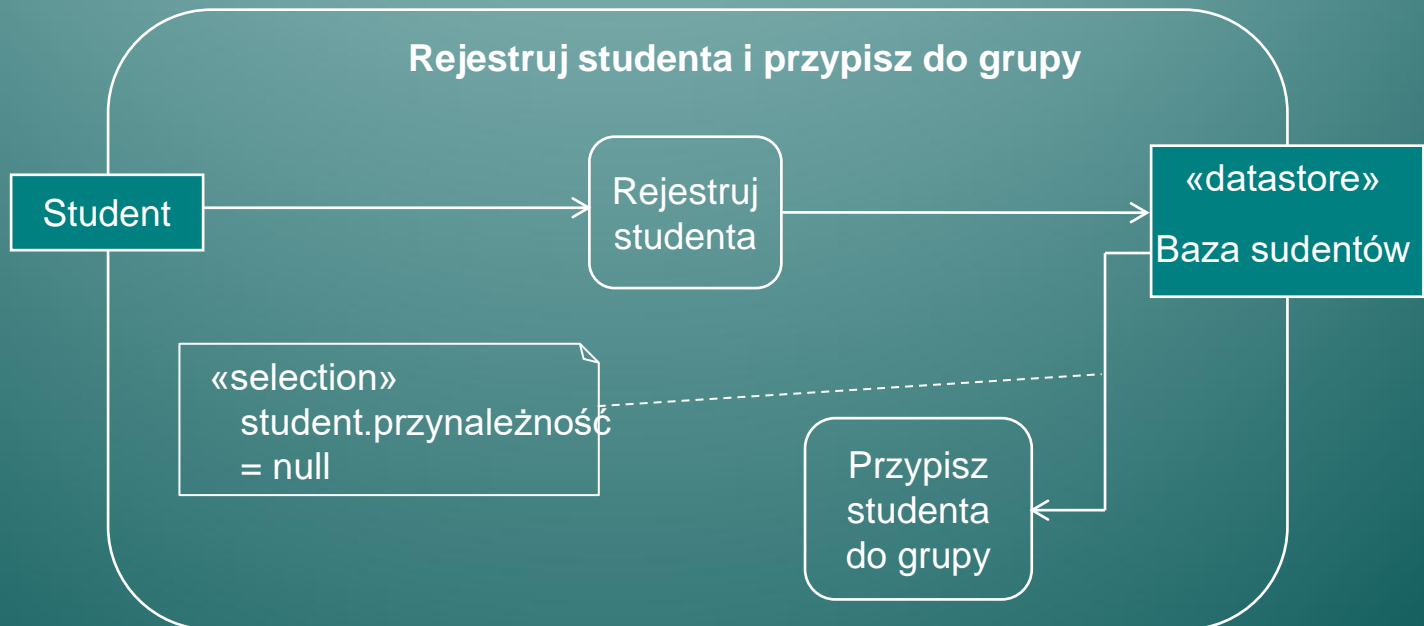


# Parametry złożonych aktywności

**Parametr aktywności:** oznacza parametr wejściowy lub parametr wyjściowy dla złożonych aktywności. Do danej aktywności można przypisać więcej niż jeden parametr.

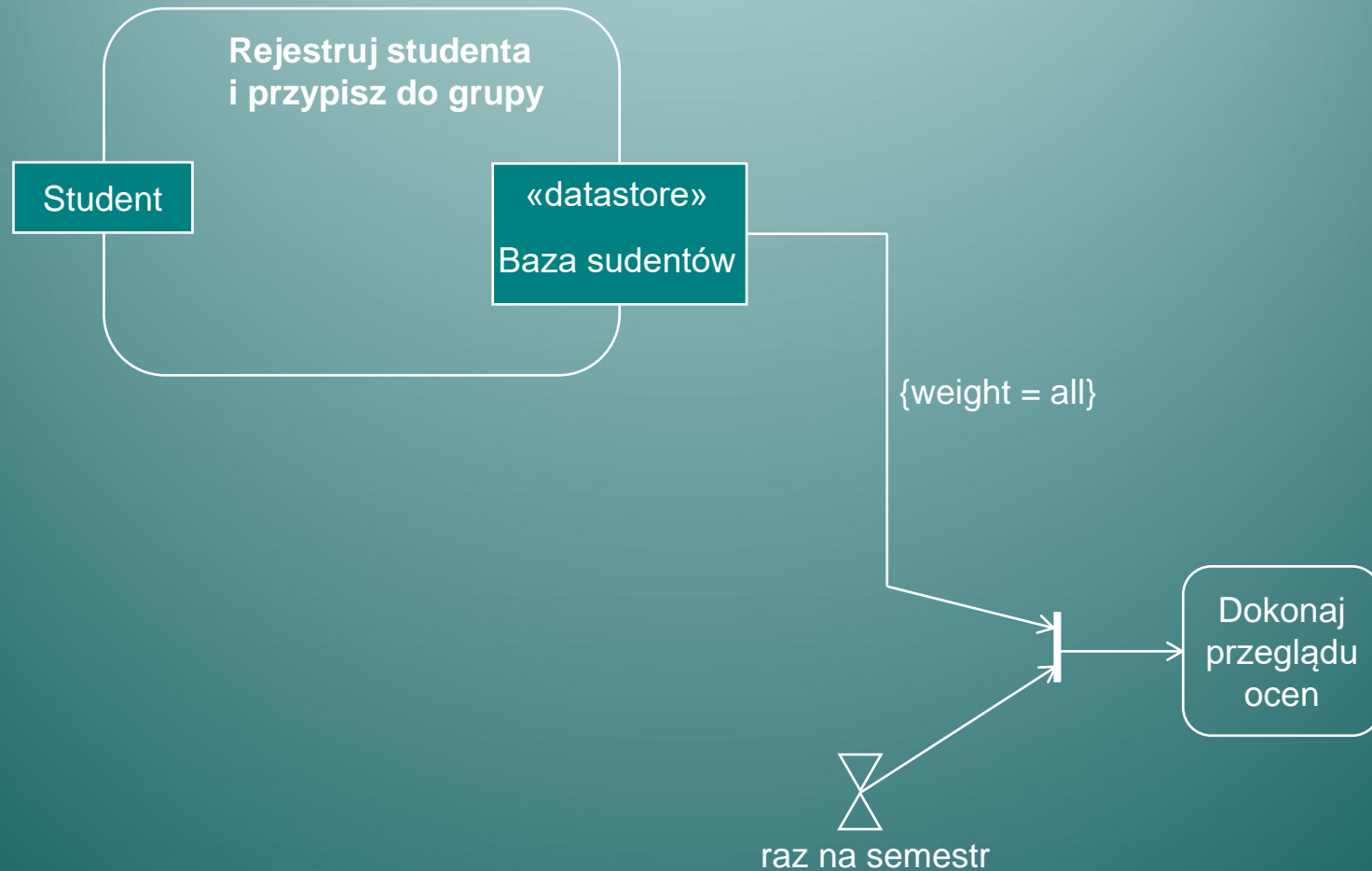


## Przykład



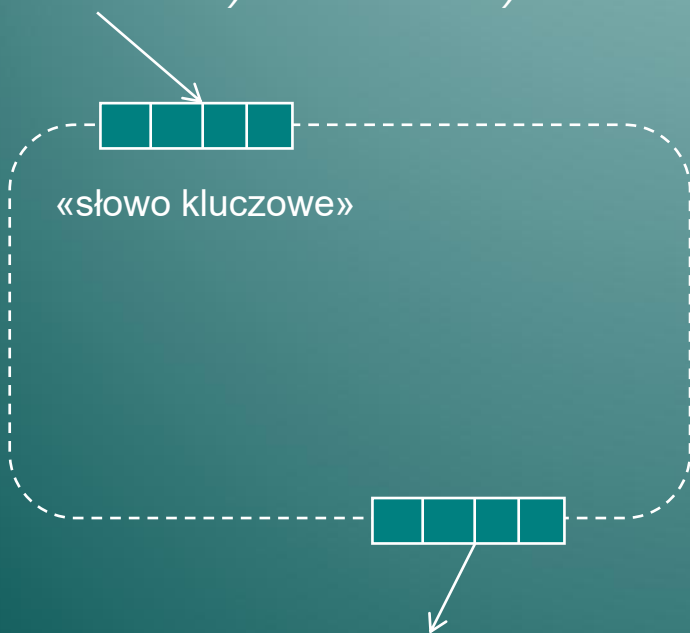


# Parametry złożonych aktywności; przykład



# Obszar rozszerzenia (1)

**Obszar rozszerzenia** (ang. *expansion region*): określa pewien zamknięty, zagnieżdżony fragment diagramu aktywności z wyspecyfikowanymi wejściami i wyjściami (w postaci wierzchołków rozszerzenia). Każde wejście jest kolekcją danych tego samego typu (co sugeruje notacja przyjęta dla wierzchołków – zbiór czterech segmentów). Operacje, zawarte wewnątrz obszaru rozszerzenia, są wykonywane raz dla każdego z wierzchołków wejściowych, zgodnie ze specyfikacją trybu przetwarzania (**słowo kluczowe**: *stream*, *parallel* (*concurrent*) lub *iterative*).



**stream:** do pojedynczej operacji jest przekazywany strumień danych

**parallel:** interakcje są wzajemnie niezależne

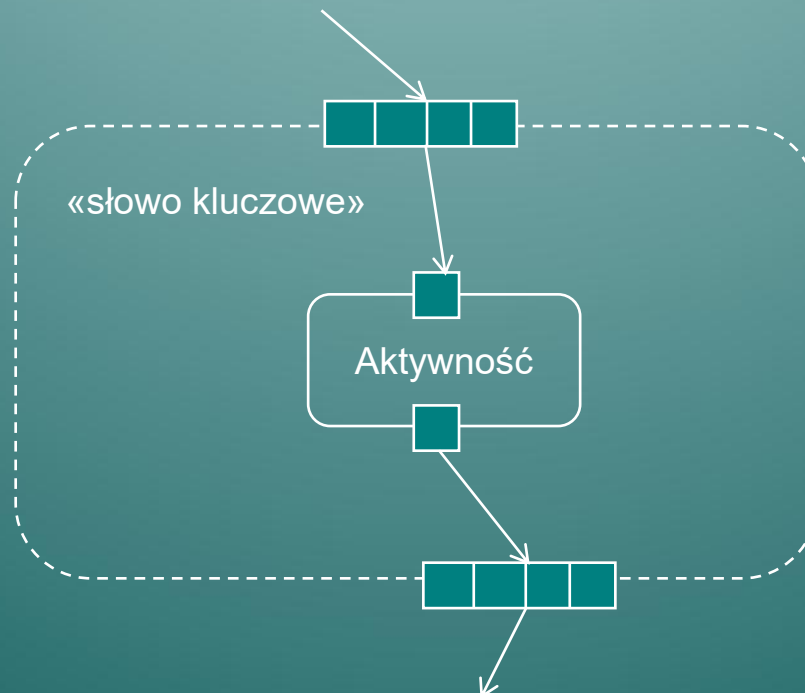
**iterative:** kolejność interakcji jest zgodna z porządkiem danych wejściowych



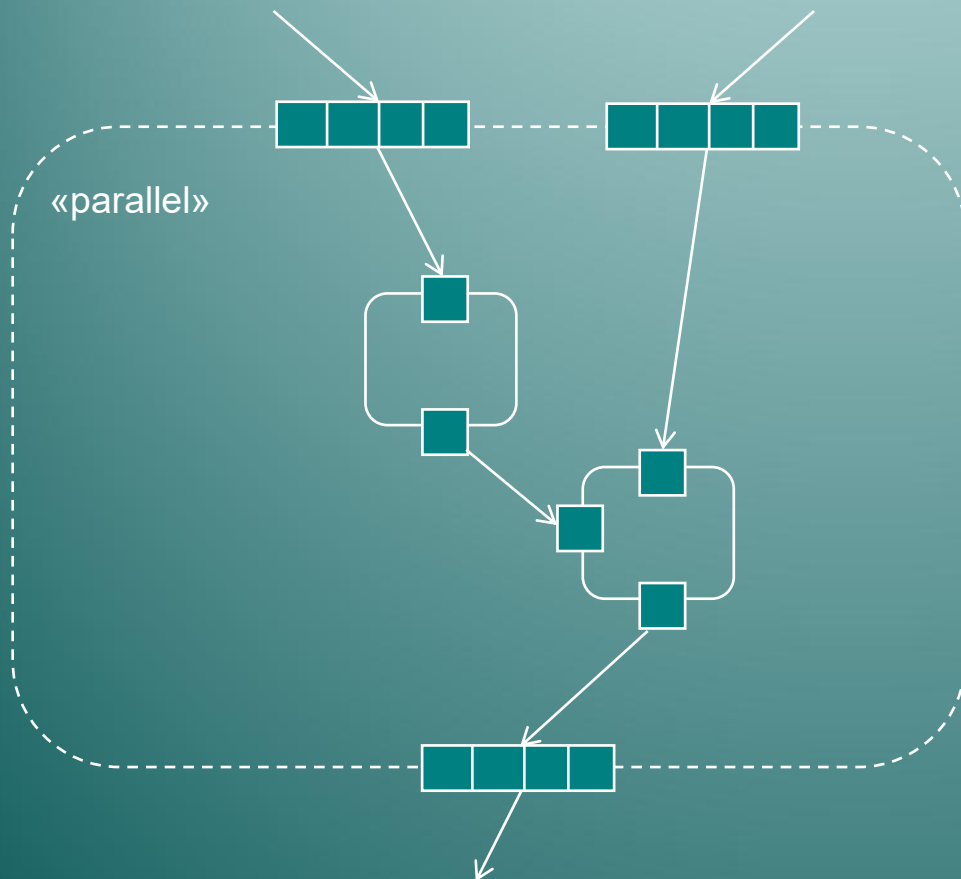
alternatywna notacja dla strzałek umożliwiających odróżnienie wierzchołków wejściowych od wyjściowych

# Obszar rozszerzenia (2)

- ✓ W przypadku ogólnym, liczba wierzchołków wejściowych obszaru rozszerzenia nie musi być równa liczbie wierzchołków wyjściowych.
- ✓ Jeśli obszar rozszerzenia posiada wierzchołki wyjściowe, typy korespondujących wierzchołków we-wy muszą być zgodne.
- ✓ „Na zewnątrz” obszaru, wierzchołki wejściowe i wyjściowe są traktowane jako kolekcje elementów, zaś „w jego wnętrzu” jako pojedyncze elementy.



# Obszar rozszerzenia; przykład

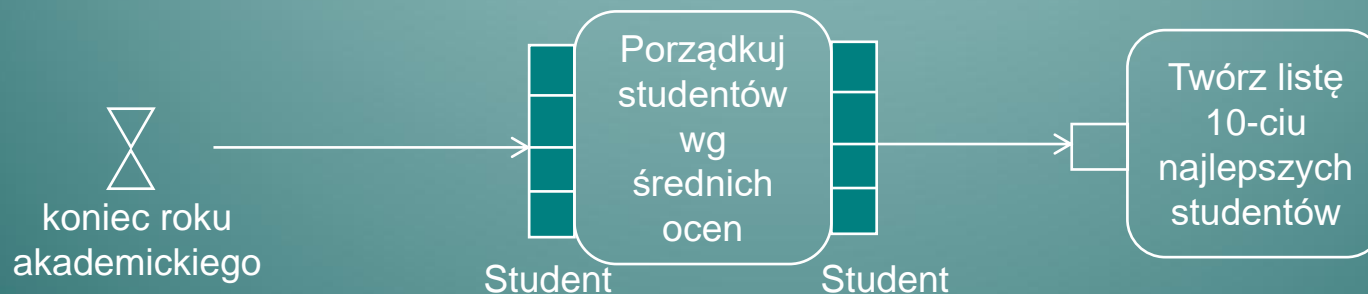


## Obszar rozszerzenia z dwoma wejściami i jednym wyjściem:

- ✓ przetwarzanie nie zostanie rozpoczęte dopóki obie wejściowe kolekcje nie będą dostępne (tryb parallel);
- ✓ obie kolekcje wejściowe muszą mieć tyle samo elementów;
- ✓ W danym momencie, przetwarzana jest para elementów, po jednym elemencie z każdej kolekcji;
- ✓ wyniki operacji na parach elementów są formowane w kolekcję wyjściową;
- ✓ kolekcja wyjściowa ma taki sam rozmiar, jak kolekcje wejściowe;
- ✓ kolekcja wyjściowa będzie dostępna „na zewnątrz” obszaru, kiedy zostanie zakończone przetwarzanie dla wszystkich elementów obu kolekcji wejściowych.

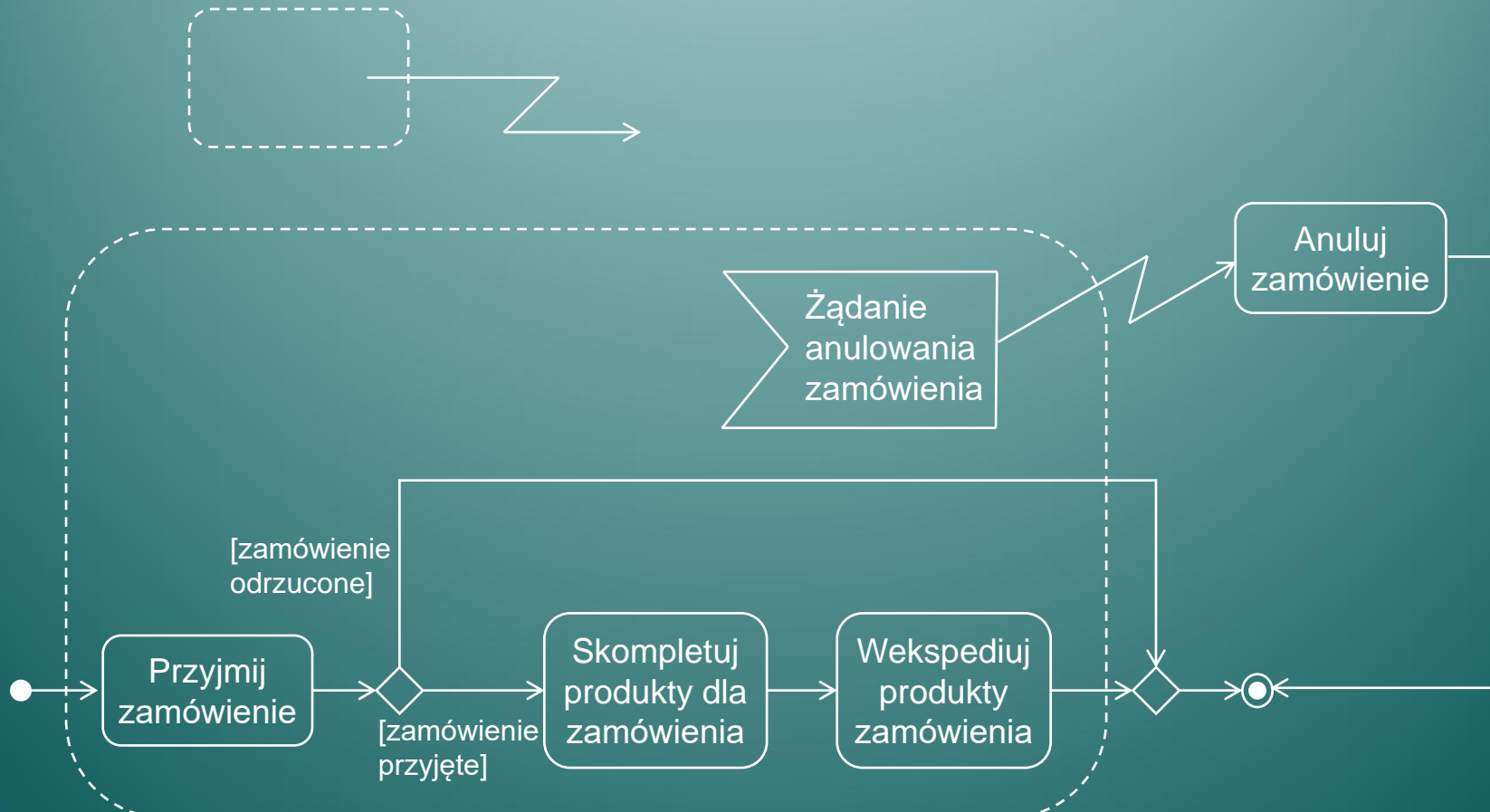
# Uproszczona notacja dla obszarów rozszerzenia

Uproszczona notacja dla obszarów rozszerzenia: wykorzystywana jest, dla regionów rozszerzeń z tylko jedną operacją; operacja musi być wykonana na wszystkich elementach kolekcji wejściowej.



# Obszar przerwania

**Obszar przerwania:** określa zamknięty, zagnieżdżony fragment diagramu aktywności, wewnątrz którego w wyniku wystąpienia określonego zdarzenia, możliwe jest przerwanie „normalnego” przetwarzania.

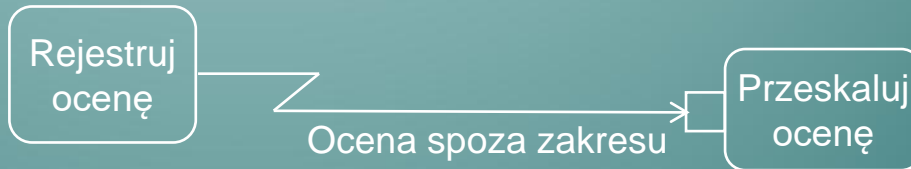


# Obsługa wyjątków

## Notacja podstawowa



## Przykład



## Notacja alternatywna



**Uwaga:** Zarówno *wierzchołek chroniony*, jak i *wierzchołek z obsługą wyjątku* muszą być na tym samym poziomie zagnieżdżenia.

# Partycje (1)

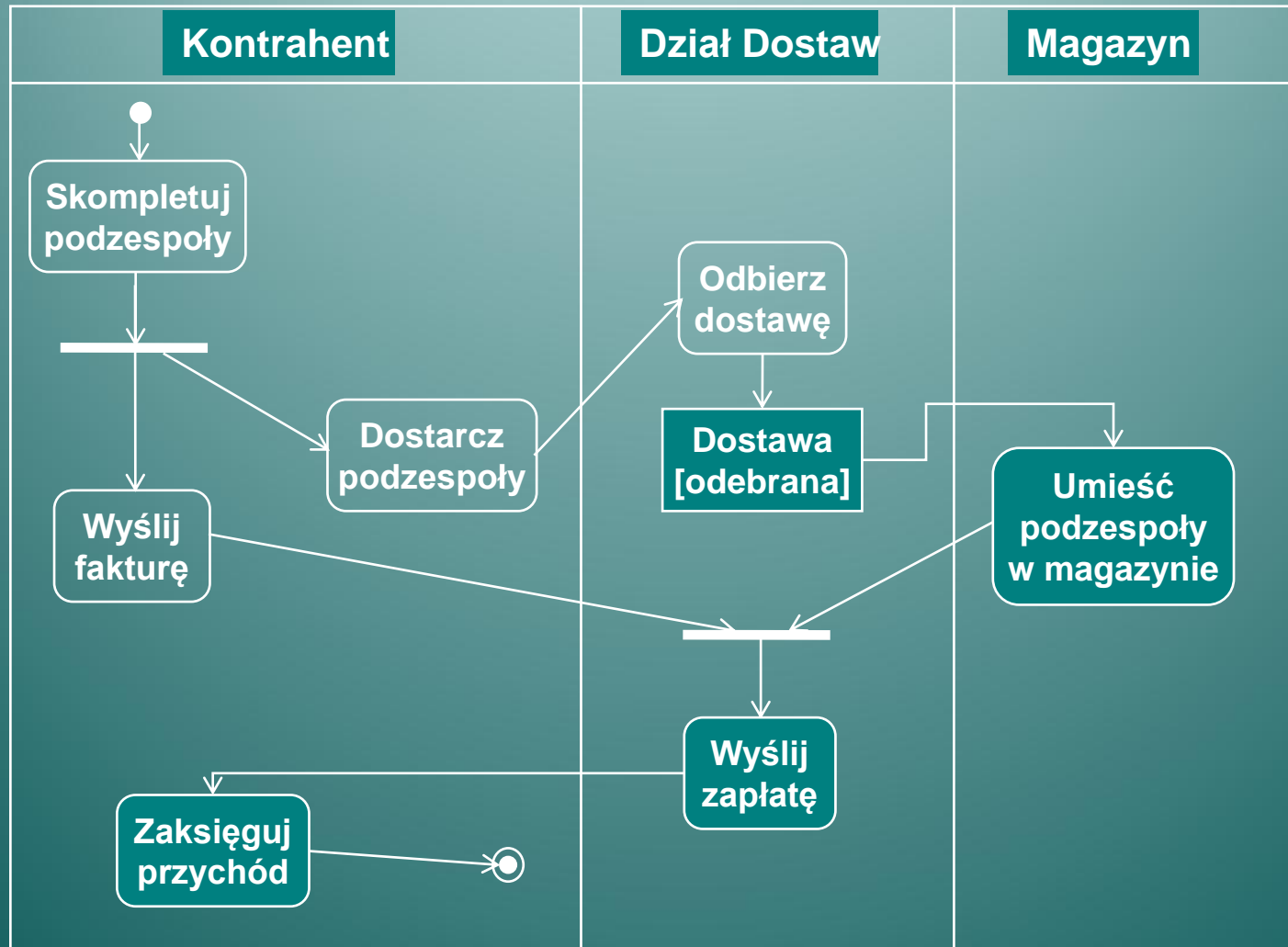
**Partycja** (ang. activity partitions): kategoria modelowania, wykorzystywana do grupowania elementów diagramów aktywności. Bazę dla grupowania stanowi podział odpowiedzialności za realizację operacji zawartych w wyróżnionej grupie.

Diagramy aktywności, w wersji podstawowej, specyfikują przepływy operacji bez określania, kto jest odpowiedzialny za ich realizację, np. którzy ludzie czy które komórki organizacyjne (z perspektywy pojęciowej) czy też które klasy (z perspektywy projektowej). Można opisywać każdą aktywność podając osobę, komórkę organizacyjną, organizację czy klasę odpowiedzialną za jej wykonanie, ale być może wygodniejszym sposobem przenoszenia informacji tego rodzaju jest grupowanie aktywności odpowiednio do odpowiedzialności i umieszczanie ich w oddzielnych partycjach rozdzielonych pionowymi (i/lub poziomymi) liniami. Nazwy partycji mogą odpowiadać nazwom osób, komórek organizacyjnych, organizacji czy klas odpowiedzialnych za wykonanie aktywności. Partycje mogą być dalej dzielone na *podpartycje* (ang. activity subpartition).



# Partycje (2)

ad Realizacja dostawy podzespołów



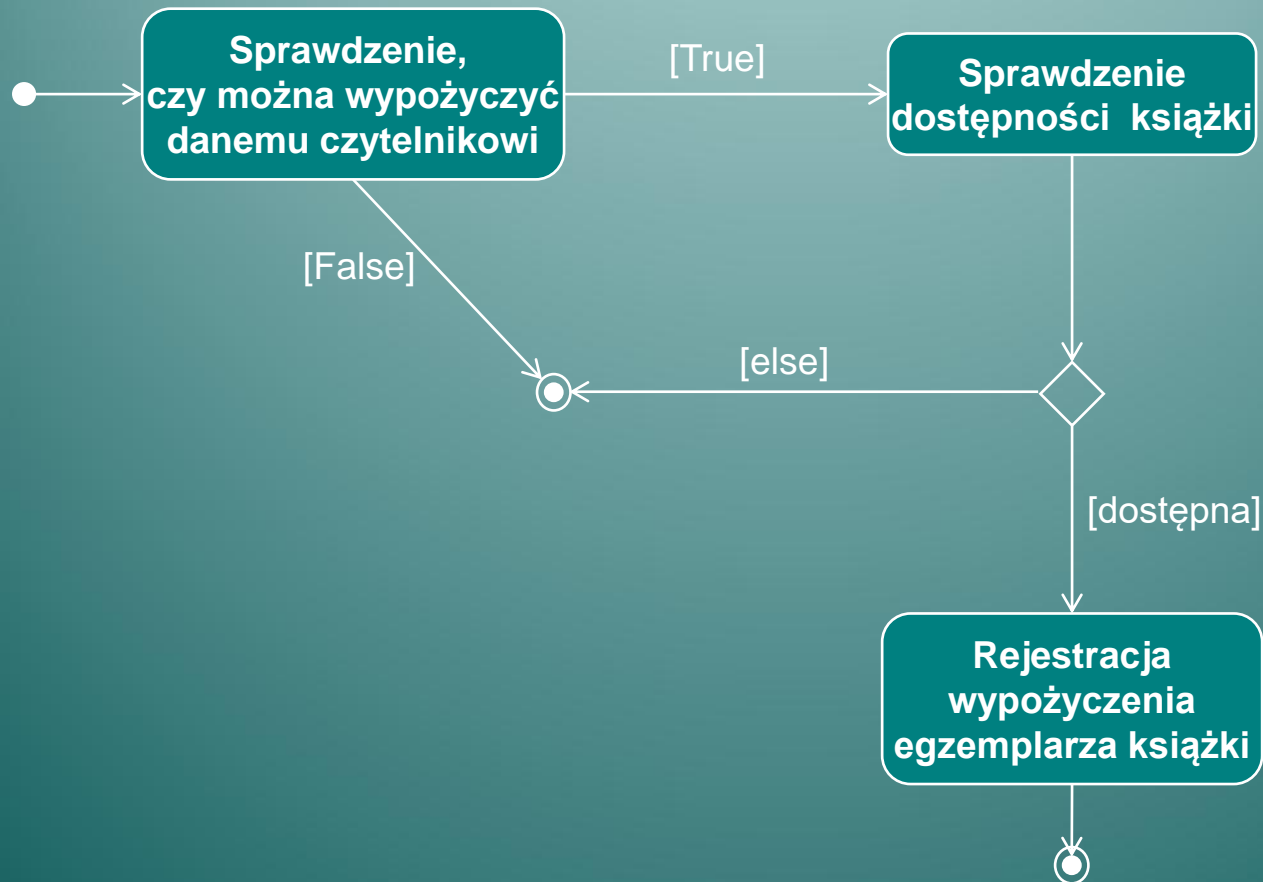
# Przykład dla wymagań z biblioteką (1)

## Uproszczony scenariusz dla przypadku użycia: wypożyczenie egzemplarza książki

- ✓ Sprawdzenie, czy można wypożyczyć danemu czytelnikowi o ile można, to:
  - ✓ Sprawdzenie, czy książka jest dostępna (czy jest wolny egzemplarz) o ile jest dostępny egzemplarz, to:
    - ✓ Rejestracja wypożyczenia



# Przykład dla wymagań z biblioteką (2)



# Podsumowanie diagramów aktywności

## Kiedy używać diagramów aktywności:

- Do analizowania przypadków użycia – gdy interesują nas bardziej operacje niezbędne do realizacji danego przypadku (czy też wzajemne zależności między tymi operacjami), a nie to, kto jest odpowiedzialny za ich przeprowadzenie. Przepisanie operacji do obiektów może być wykonywane na etapie późniejszym, z wykorzystaniem diagramów interakcji.
- Do zrozumienia interakcji zachodzących między przypadkami użycia (ważne zastosowanie).
- Do modelowania przetwarzania wielowątkowego.

## Kiedy nie używać diagramów aktywności:

- Do pokazywania współpracy między obiektami w trakcie realizacji przypadku użycia – do tego bardziej nadają się diagramy interakcji.
- Do pokazywania zachowań obiektów w trakcie ich życia, w tym celu powinno się wykorzystywać diagramy stanów.