

Diagramy czynności
Na podstawie
UML 2.0 Tutorial

http://sparxsystems.com.au/resources/uml2_tutorial/

Zofia Kruczkiewicz

Diagramy czynności

1. Diagramy czynności UML

http://sparxsystems.com.au/resources/uml2_tutorial/

2. Przykład diagramów czynności UML – modelowanie operacji

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

3. Przykład diagramów czynności UML – modelowanie przepływu czynności

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

Diagramy czynności

1. Diagramy czynności UML

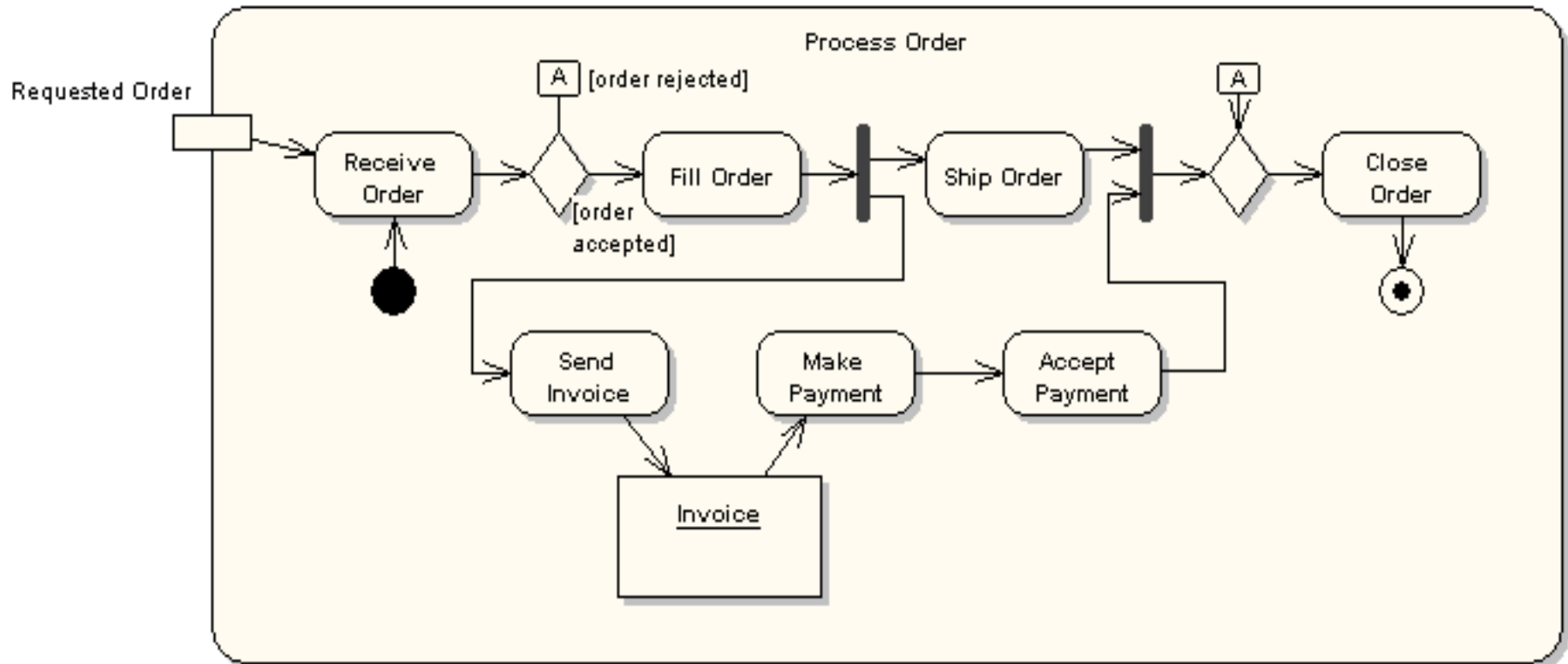
http://sparxsystems.com.au/resources/uml2_tutorial/

Diagramy czynności

Diagram czynności jest odmianą diagramu stanu i opisuje interakcje między obiektami.

- **jak** pobierane są operacje,
- **co** operacje wykonują (zmiana stanu obiektu),
- **kiedy** operacje są wykonywane (sekwencje czynności lub akcji)
- **gdzie** są wykonywane.

ad Activity (Example)



ad Activity

Activity

Czynność- zawiera specyfikację sparametryzowanych zachowań (akcje, przepływ sterowania)

ad Action

Perform
Action

Akcja – elementarny krok czynności

ad Conditions

```
«localPreCondition»  
{A drink is selected that the  
vending machine contains}
```



```
«localPostCondition»  
{The vending machine  
dispensed the drink selected}
```

ad Activity Edge



Ograniczenia akcji:

warunki przed akcją i
po akcji

Np.

Warunek przed: Wybór napoju
w automacie, jeśli istnieje

Stan: akcja wydania napoju
(Dispense drink)

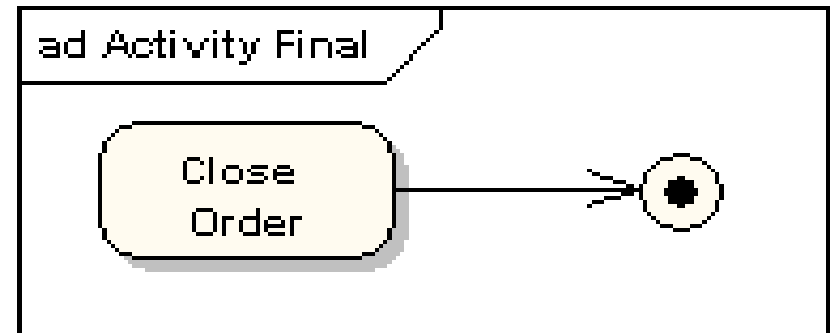
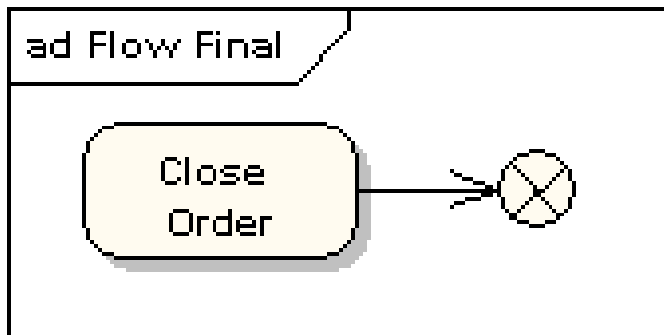
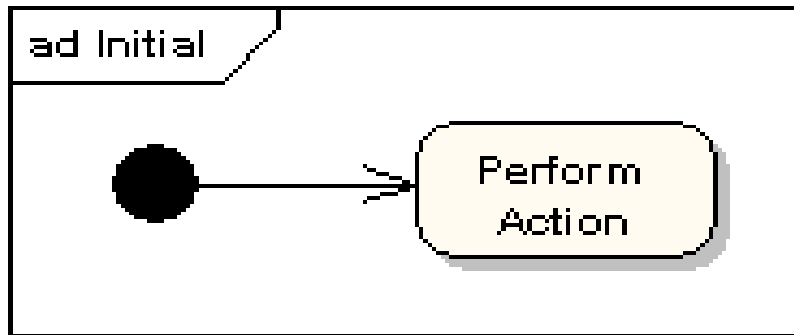
Warunek po: Maszyna wydała
wybrany napój

Przeływ sterowania:

Przejście z jednej akcji
do drugiej

np. wysłanie rachunku –Send
Payment) i akceptacja rachunku
(Accept Payment)

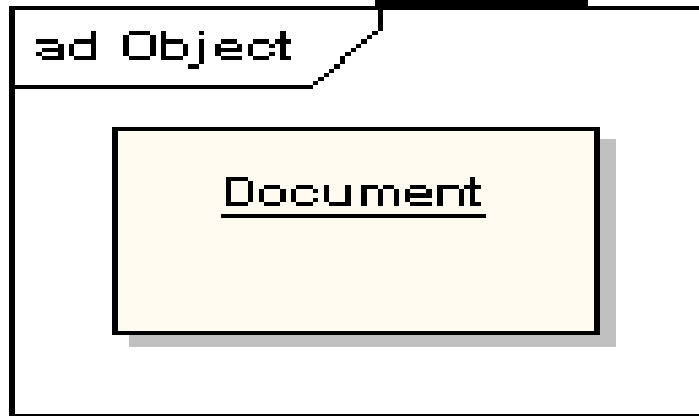
Wierzchołek początkowy (Initial) i końcowy (Final)



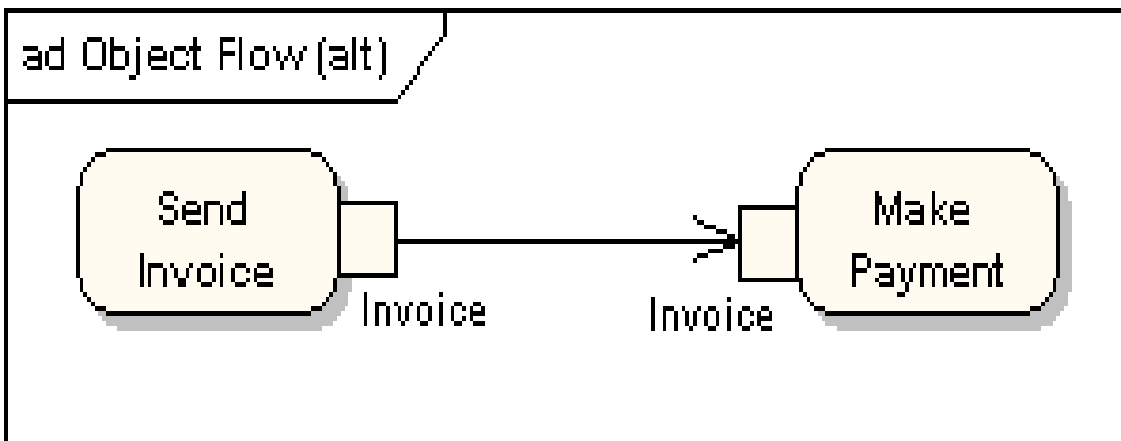
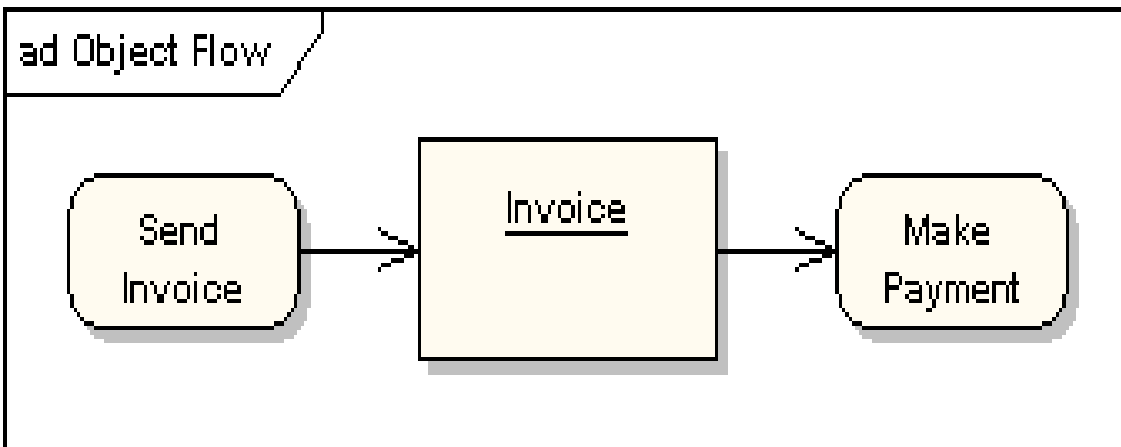
Wierzchołek końca sterowania koniec pojedynczego przebiegu sterowania

Wierzchołek końca wielu przebiegów sterowania wewnątrz czynności

Obiekt



Magazyn danych



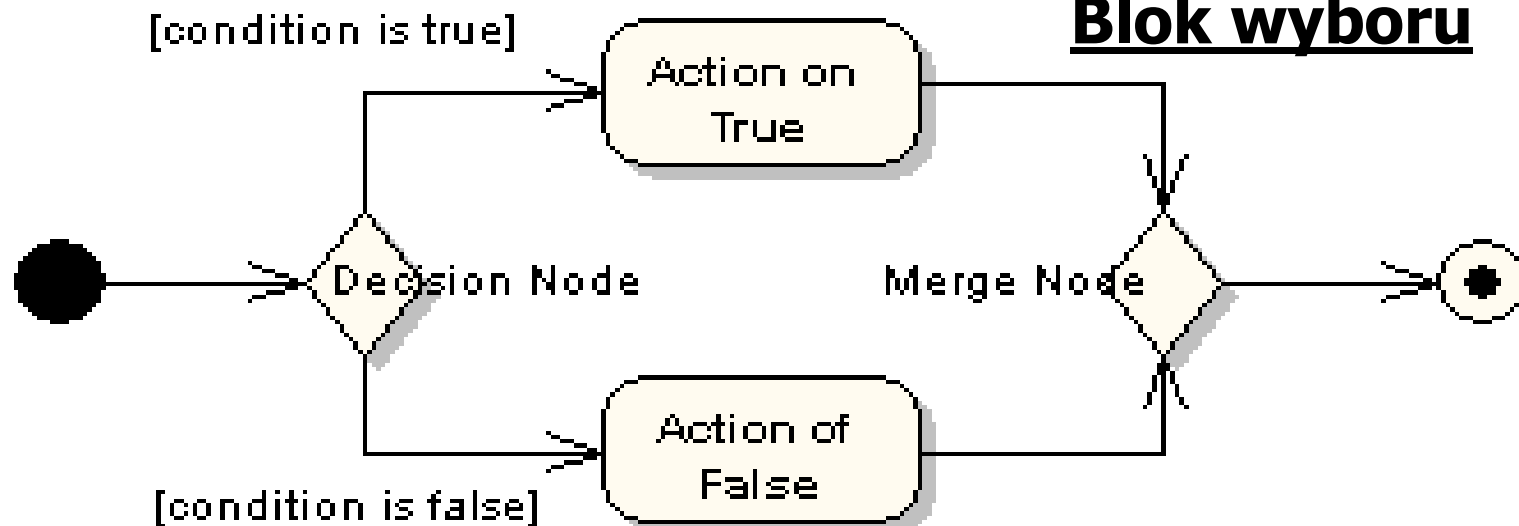
Przepływ obiektów

Wysłanie (Send Invoice) obiektu faktura (Invoice) w celu dokonania zapłaty (Make Payment)

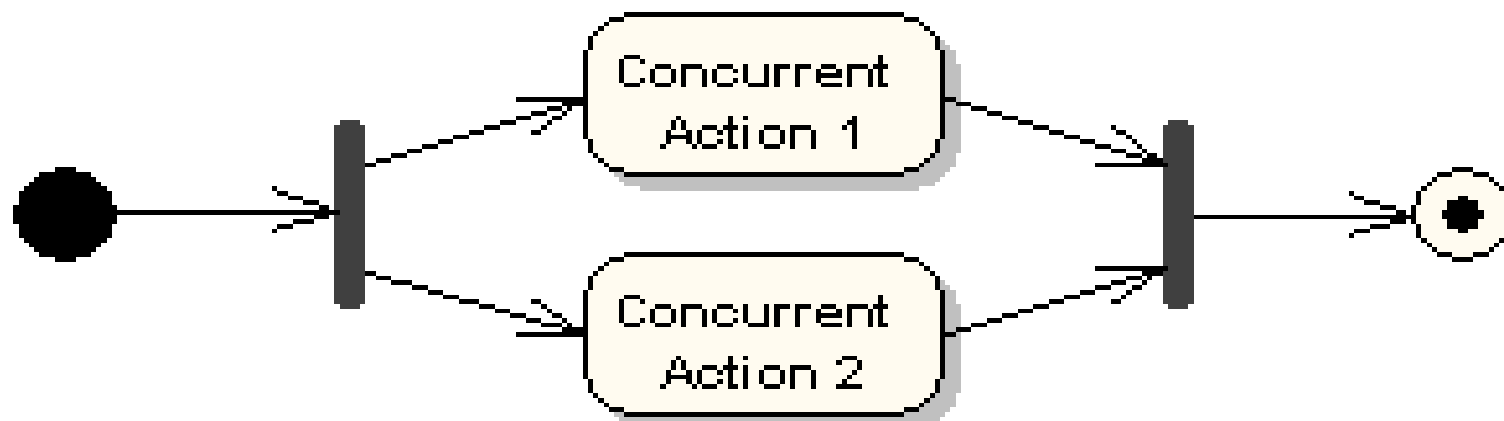
Przepływ obiektów

Równoważny diagram

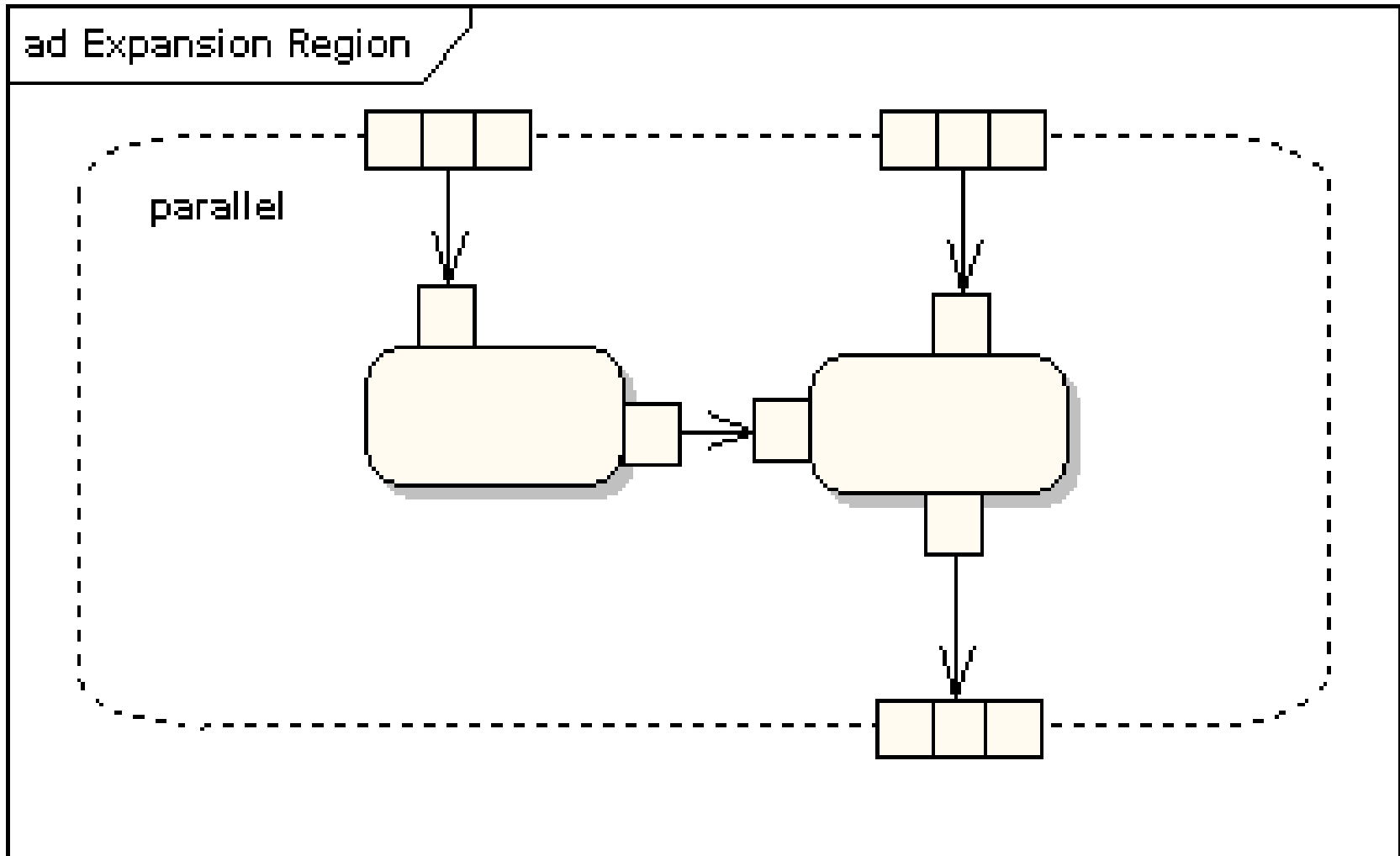
Blok wyboru



Bloki rozdzielania i łączenia

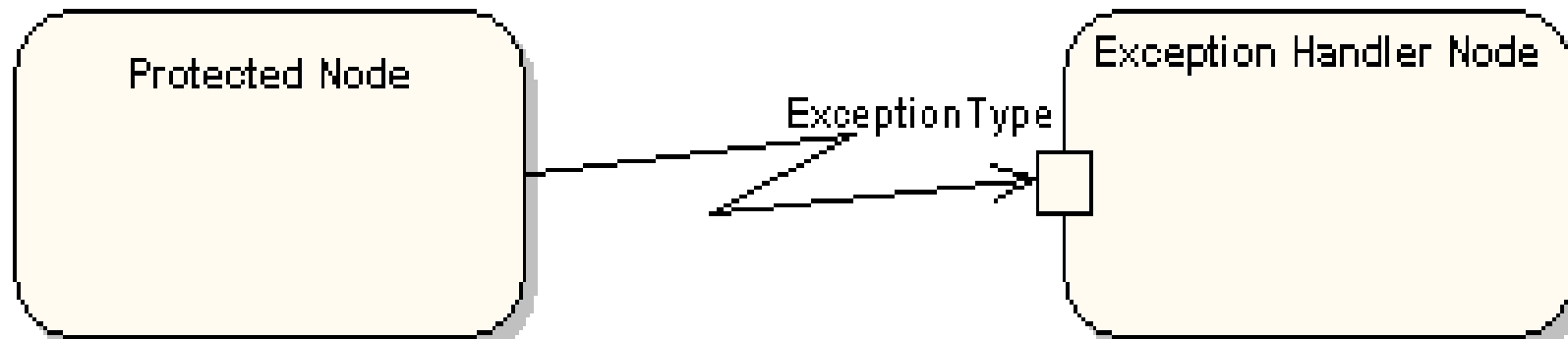


Powtarzanie czynności: iteracyjnie (iterative), równoległe (parallel) lub w postaci strumienia (stream)



ad Exception Handler

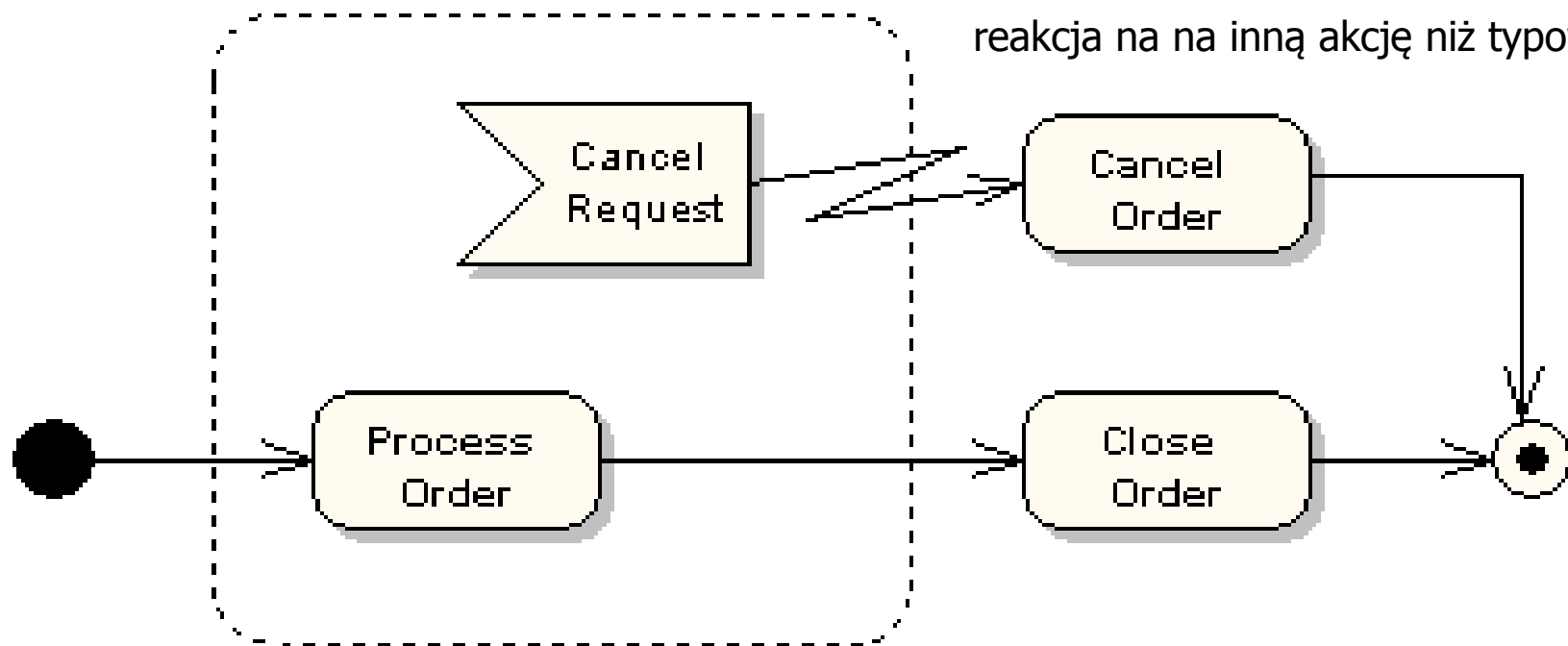
Obsługa wyjątków reakcja na błąd



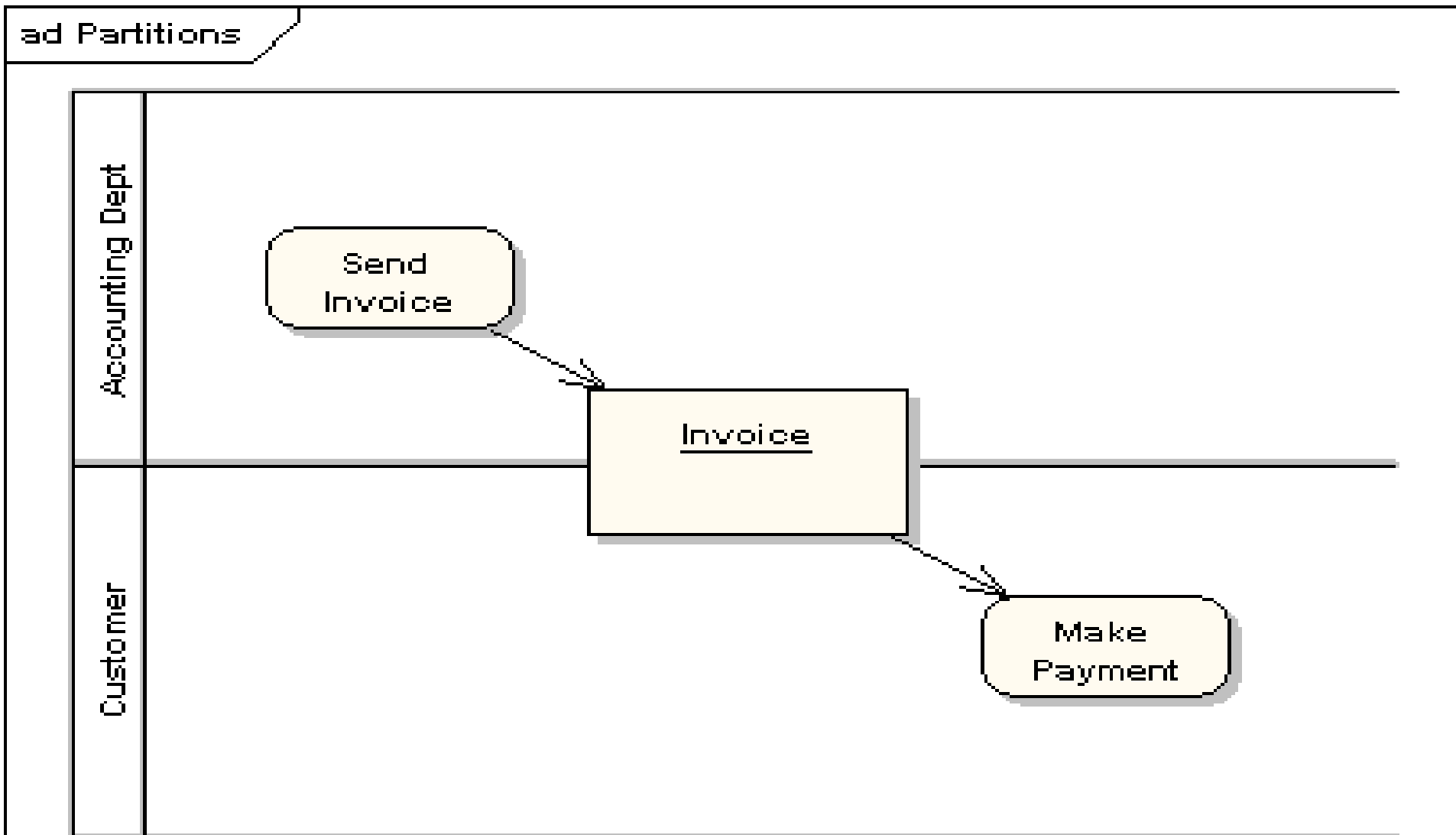
ad Interruptible Activity Region

Obsługa przerwania

reakcja na na inną akcję niż typowa



Partycje(tory) – np. podział czynności wykonywanych na obiekcie Faktura (Invoice) przez dwa różne obiekty: Wydział Finansowy (Accounting Department) i klienta (Customer).



Diagramy czynności

1. Diagramy czynności UML

http://sparxsystems.com.au/resources/uml2_tutorial/

2. Przykład diagramów czynności UML – modelowanie operacji

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

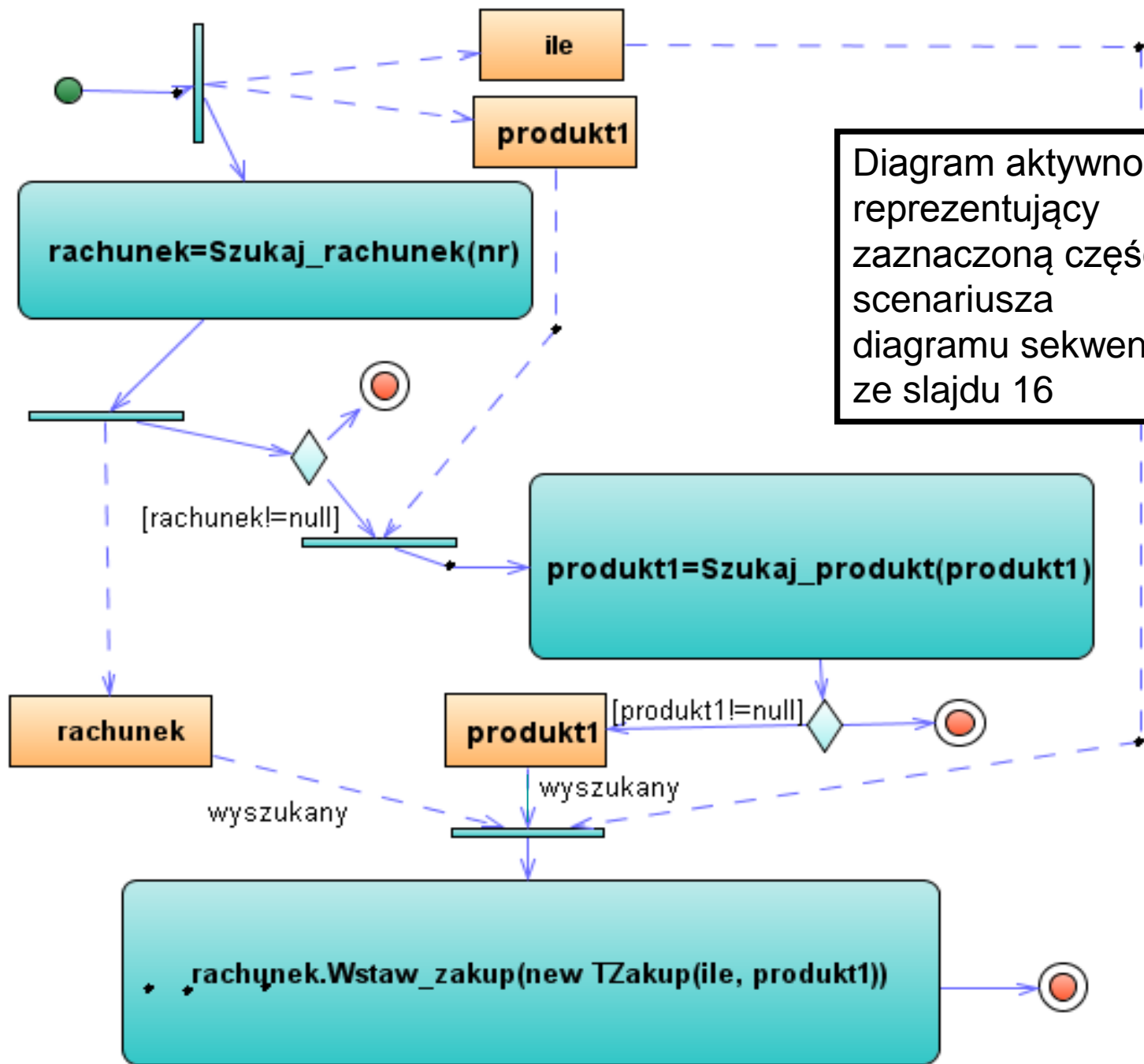


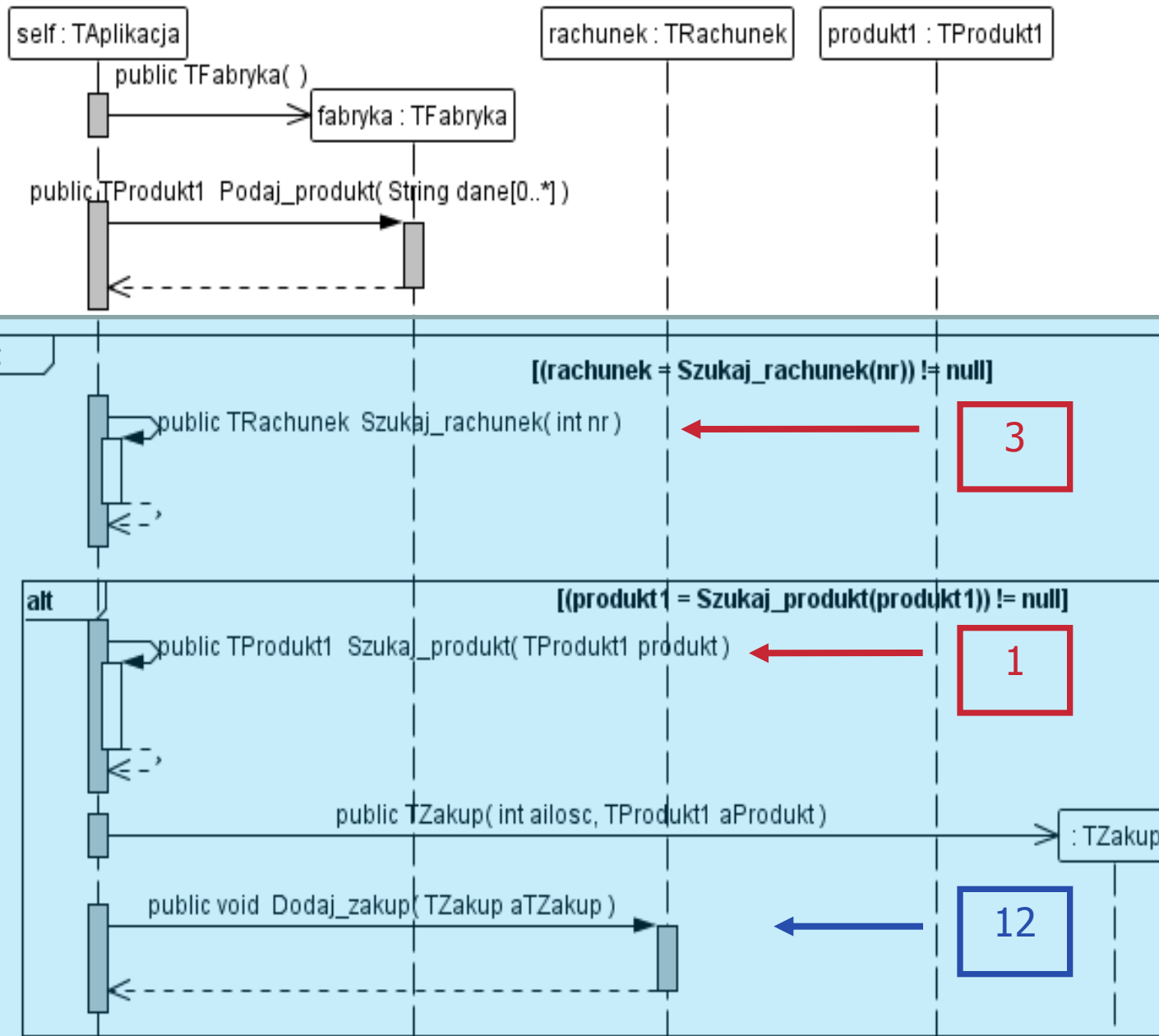
Diagram aktywności reprezentujący zaznaczoną część scenariusza diagramu sekwencji ze slajdu 16

Kod źródłowy projektowanej metody za pomocą diagramu aktywności

```
public void Wstaw_zakup (int nr, int ile, TProdukt1 produkt1)
{
    TRachunek rachunek;
    if ((rachunek=Szukaj_rachunek(nr))!=null )
        if ((produkt1=Szukaj_produkt(produkt1))!=null )
            rachunek.Wstaw_zakup(new TZakup(ile,produkt1));
}
```

(5) Wstawianie nowego zakupu

(void TApplikacja::Wstaw_zakup (int nr, int ailosc, String dane[]))




```
//TAplikacja
```

```
public void Wstaw_zakup (int nr, int ile, String dane[])  
{  
    TRachunek rachunek;  
    TFabryka fabryka = new TFabryka();  
    TProdukt1 produkt1 = fabryka.Podaj_produkt(dane);  
    if ((rachunek=Szukaj_rachunek(nr)) != null)  
        if ((produkt1=Szukaj_produkt(produkt1)) != null)  
            rachunek.Dodaj_zakup(new TZakup(ile, produkt1));  
}
```

Wytyczne dla tworzenia diagramów operacji

1. Należy uwzględnić abstrakcje uczestniczące w tej operacji. Są to parametry przekazywane do metody, atrybuty klasy metody oraz klas obiektów związanych z metodą.
2. Należy zidentyfikować warunki wstępne stanu początkowego i końcowego modelowanej operacji. Należy znaleźć wszystkie wartości, które nie mogą zmienić się podczas realizacji operacji.
3. Przechodząc od stanu początkowego do końcowego należy modelować kolejne stany czynności lub stany akcji.
4. Należy w razie potrzeby korzystać z rozgałęzień do modelowania ścieżek warunkowych i iteracji.
5. Jeśli operacja należy do klasy aktywnej (i tylko wtedy), należy wtedy korzystać z rozwidleń i scaleń, żeby określić równoległe przepływy sterowania.

Diagramy czynności

1. Diagramy czynności UML

http://sparxsystems.com.au/resources/uml2_tutorial/

2. Przykład diagramów czynności UML – modelowanie operacji

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

3. Przykład diagramów czynności UML – modelowanie przepływu czynności i obiektów

(wg Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika)

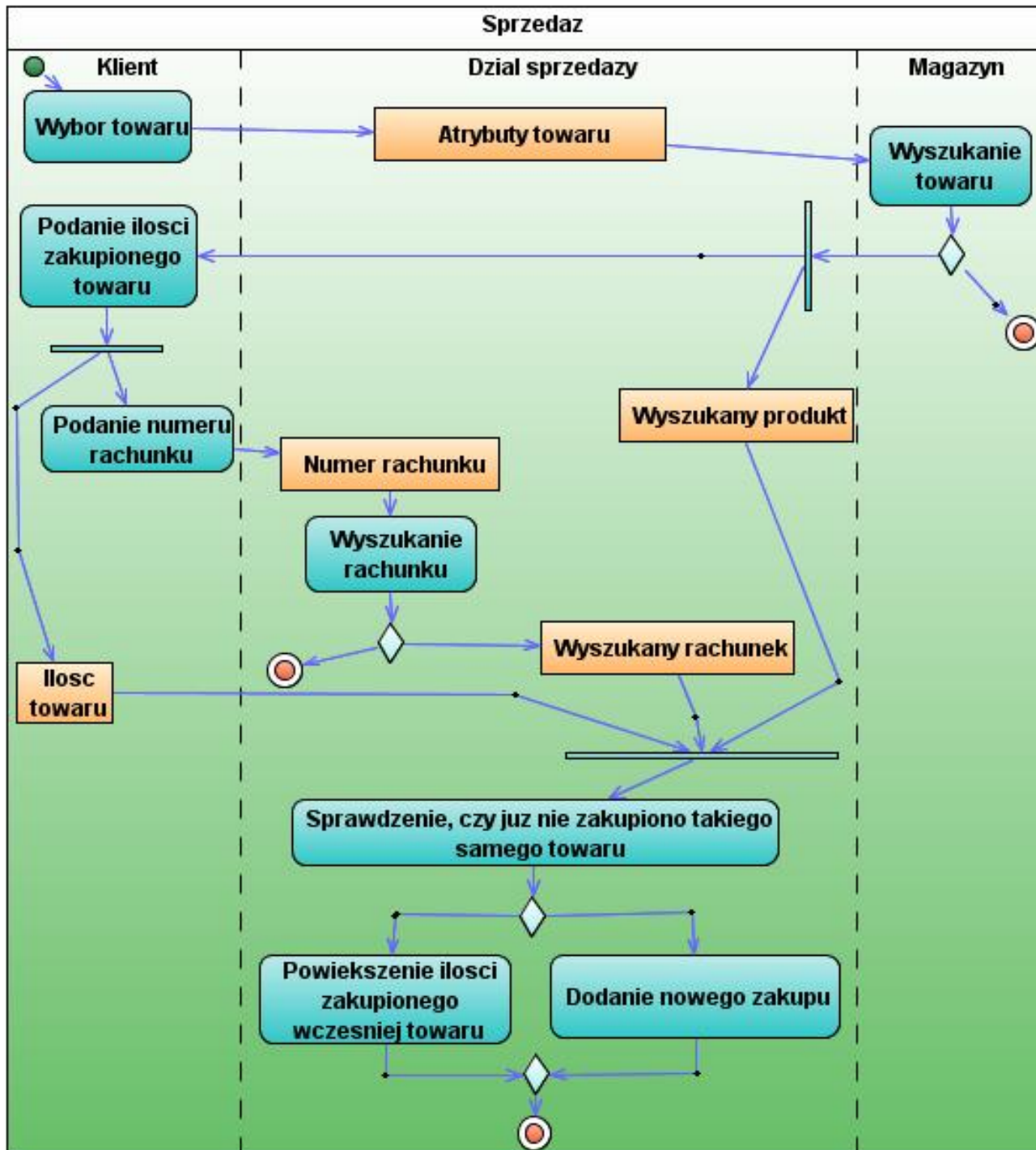
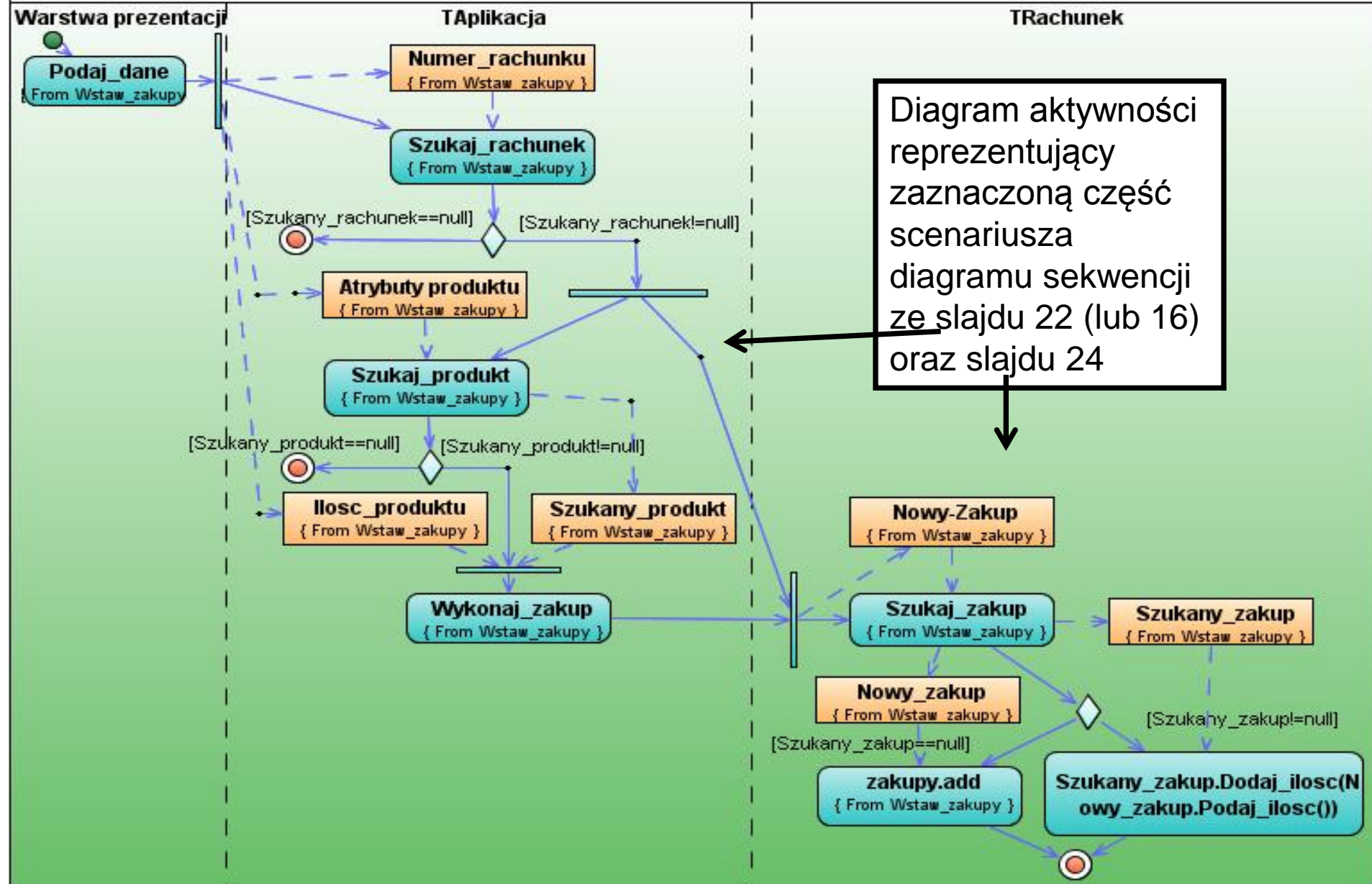


Diagram czynności jako **model biznesowy** systemu sprzedaży towarów

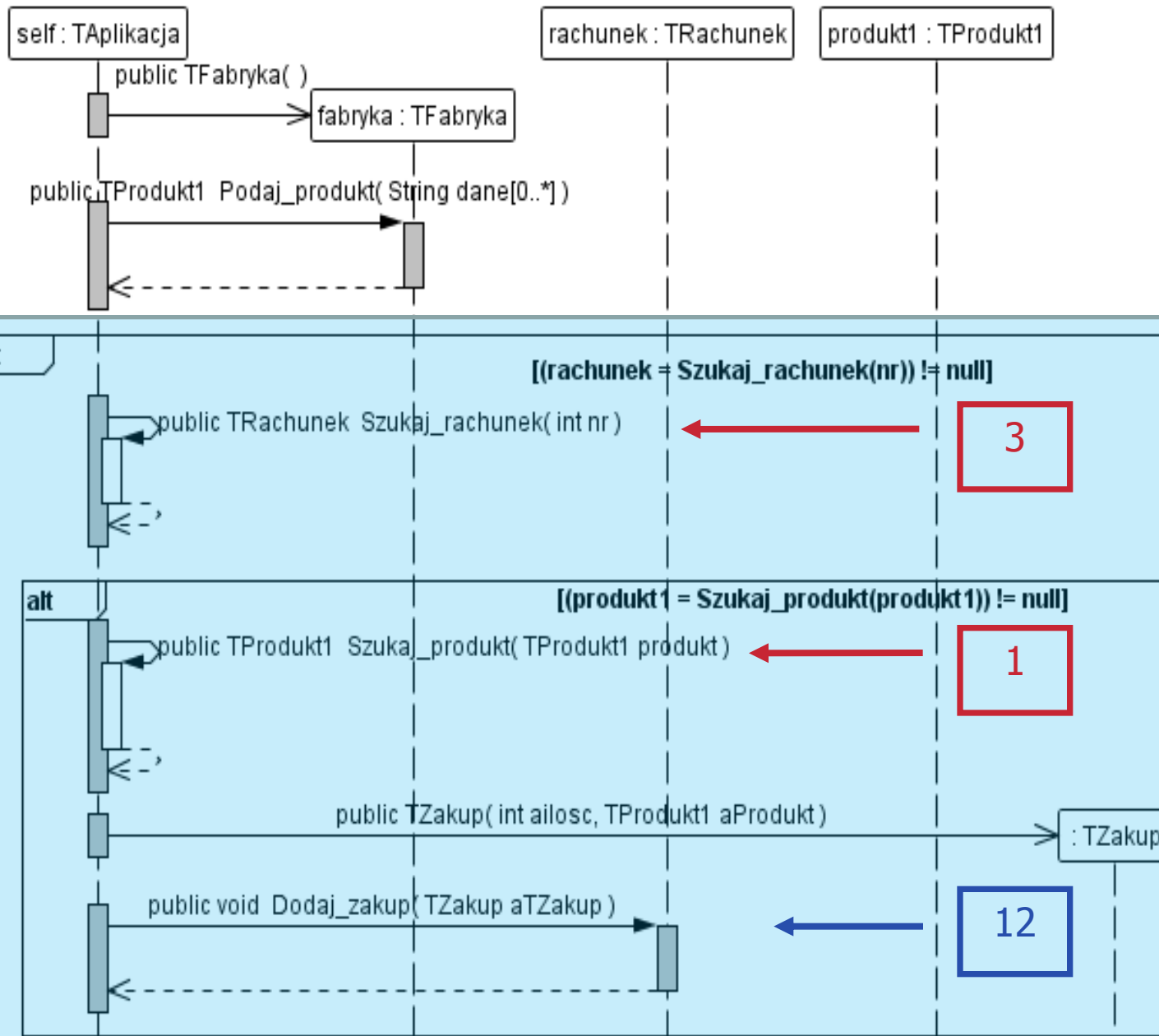
Diagram czynności przypadku użycia *Wstawianie nowego zakupu* (model przypadku użycia w warstwie biznesowej)

Dodanie_zakupu



(5) Wstawianie nowego zakupu

(void TApplikacja::Wstaw_zakup (int nr, int ailosc, String dane[]))



```
//TAplikacja
```

```
public void Wstaw_zakup (int nr, int ile, String dane[])  
{  
    TRachunek rachunek;  
    TFabryka fabryka = new TFabryka();  
    TProdukt1 produkt1 = fabryka.Podaj_produkt(dane);  
    if ((rachunek=Szukaj_rachunek(nr)) != null)  
        if ((produkt1=Szukaj_produkt(produkt1)) != null)  
            rachunek.Dodaj_zakup(new TZakup(ile, produkt1));  
}
```

(12) void TRachunek::Dodaj zakup(TZakup aTZakup)

self : TRachunek

aTZakup : TZakup

zakup : TZakup

Zakupy : ArrayList

alt [(zakup = Szukaj_zakup(aTZakup)) != null]

public TZakup Szukaj_zakup(TZakup aTZakup)

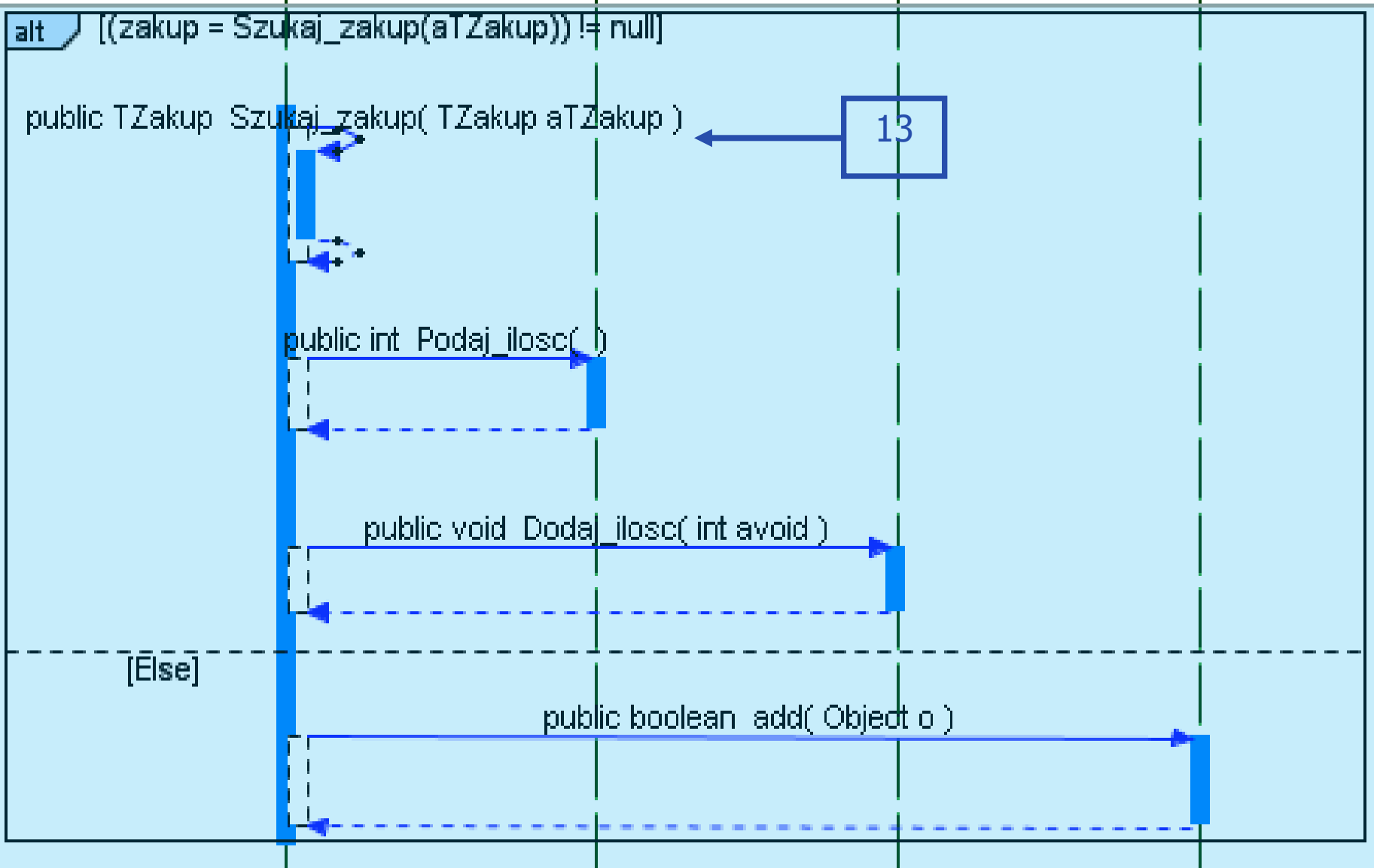
13

public int Podaj_ilosc()

public void Dodaj_ilosc(int avoid)

[Else]

public boolean add(Object o)




```
//TRachunek
```

```
private ArrayList<TZakup> Zakupy =  
    new ArrayList<TZakup>();
```

```
public void Dodaj_zakup (TZakup aTZakup)
```

```
{  
    TZakup zakup;  
    if ((zakup = Szukaj_zakup(aTZakup)) != null)  
        zakup.Dodaj_ilosc(aTZakup.Podaj_ilosc());  
    else  
        Zakupy.add(aTZakup);  
}
```

Wytyczne dla tworzenia diagramów czynności

1. Należy ustalić najważniejsze czynności - nie można przedstawić na jednym diagramie wszystkich czynności
2. Należy wybrać obiekty przedsiębiorstwa, które są zobowiązane do realizacji bardziej ogólnego przepływu. Mogą to być elementy rzeczywiste, istniejące w słownictwie systemu, lub elementy abstrakcyjne. W obu przypadkach należy utworzyć tor dla każdego wybranego obiektu.
3. Należy zidentyfikować stan początkowy i końcowy modelowanego przepływu.
4. Przechodząc od stanu początkowego do końcowego należy modelować kolejne stany czynności lub stany akcji.
5. W przypadku złożonych akcji lub często występujących zbiorów akcji należy je połączyć w stany czynności. Z każdym stanem skojarz oddzielny diagram czynności, który przedstawia zebrane nim akcje.
6. Należy zobrazować przepływy czynności między stanami akcji i stanami czynności. Pierwsze powinny być brane pod uwagę przepływy sekwencyjne, potem rozgałęzienia, na końcu rozwidlenia i scalenia.
7. Jeśli w modelowanym przepływie czynności biorą udział istotne obiekty, należy je umieścić na diagramie. Należy uwzględniać zmieniające się atrybuty i stany tych obiektów, jeśli jest to konieczne do zrozumienia przepływu tych obiektów.