

## **Instrukcja 8**

### **Laboratorium 11**

**Opracowanie diagramu stanów dla wybranej klasy,  
reprezentującego wpływ różnych przypadków użycia na  
zmiany stanów tej klasy, modelowanych za pomocą  
diagramów sekwencji**

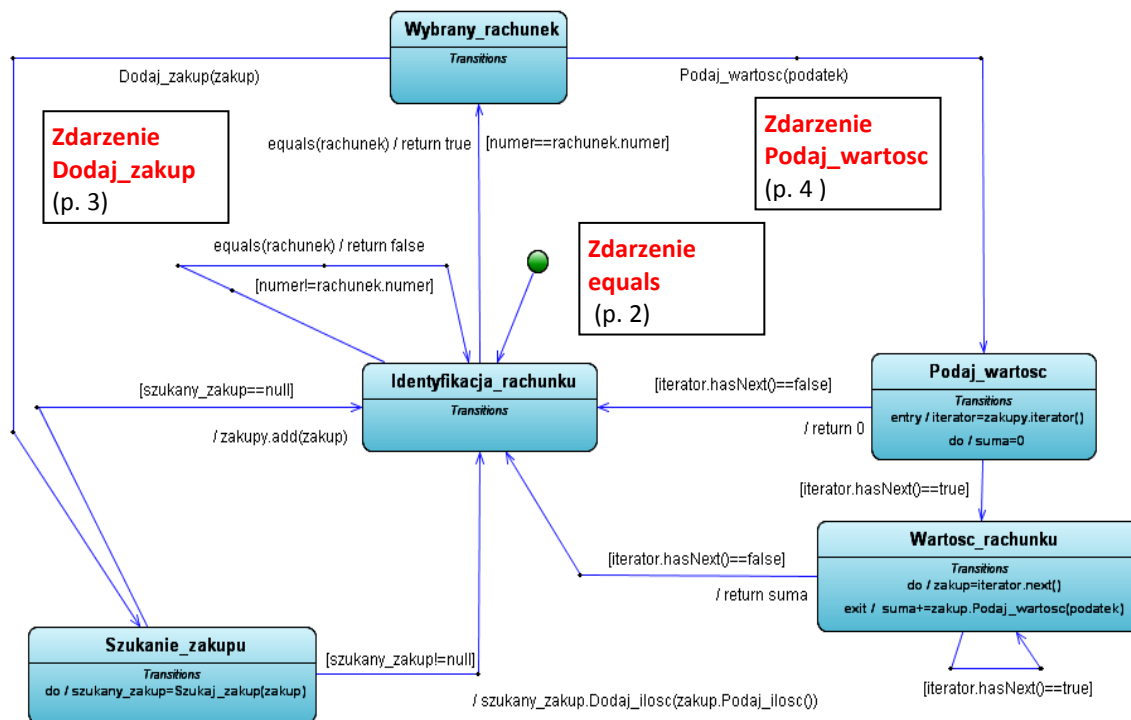
**Cel laboratorium:****Definiowanie diagramów stanów dla wybranych klas – kontynuacja tworzenia modelu projektowego ([wykład 1](#), [wykład 5-część 1](#); Dodatek 1 instrukcji)**

1. Należy wybrać klasę, która należy do modelu danych zdefiniowanego podczas lab7-10, zawierającą logikę biznesową, podobnie jak klasa **TRachunek** i wykonać dla wybranej klasy diagram stanów podobnie jak przedstawiono to dla klasy **TRachunek** ([wykład 5-część 1](#), Dodatek 1 instrukcji).
  2. Zgodnie z definicją diagramów stanów w języku UML ([wykład 5-część 1](#)), **zdarzeniem dla obiektu danej klasy jest wywołanie operacji tego obiektu przez inny obiekt, a akcjami są operacje użyte do definicji tej operacji**. Podczas laboratoriów 7-10 należało wykonać modele operacji poszczególnych klas, wykonane za pomocą diagramów sekwencji. Taki diagram sekwencji operacji wywołanej podczas zdarzenia zawiera operacje, które są akcjami. Są to: operacje wywołane od innych obiektów oraz operacje typu Message-to-Self. Dodatkowo, do akcji należy dołączyć różne wyrażenia należące do algorytmu operacji zdarzeniowej, ale niemodelowane za pomocą diagramów sekwencji np. wyrażenia matematyczne.
  3. Grupa dwuosobowa powinna wykonać dwa diagramy stanów dla dwóch wybranych klas
- Uwaga:** Należy rozwijać projekt wykonany przy realizacji instrukcji 2-7.

### Dodatek 1

#### Przykład tworzenia diagramu stanów wynikających ze scenariuszy przypadków użycia ( c.d. przykładu prezentowanego w Dodatku 1 w instrukcjach 2-7)

1. Diagram stanów obiektu klasy **TRachunek**. Modelowanie zdarzeń wywoływanych na tym obiekcie w różnych przypadkach użycia.

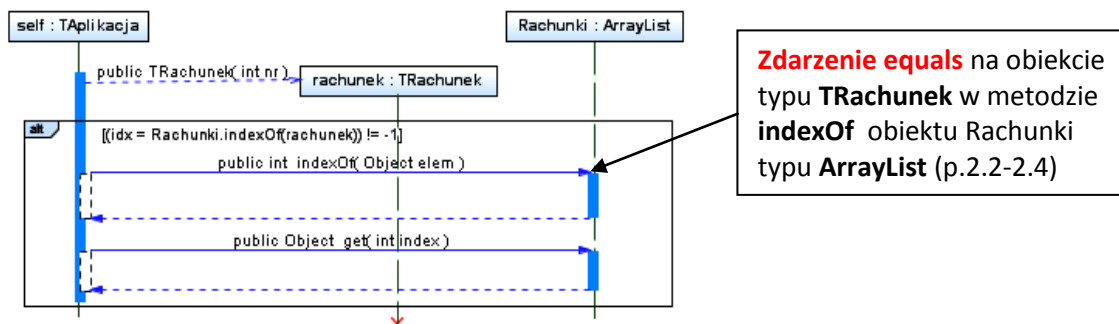


2. **Zdarzenie equals** jest wywoływane na obiekcie **TRachunek** przez obiekt Rachunki typu **ArrayList**. To zdarzenie zachodzi w momencie wywołania operacji **Szukaj\_rachunek** obiektu typu **TAplikacja** jako operacji typu *Message-to-Self*. W tej metodzie występuje wywołanie operacji **indexOf** obiektu Rachunki typu **ArrayList**. W metodzie **indexOf** tej klasy jest wywołana metoda **equals** obiektu typu **TRachunek**.

Operacja **Szukaj\_rachunek** jest wywoływana w następujących projektach przypadków użycia (instrukcje 6 i 7): **PU Wstawianie nowego rachunku**, **PU Wstawianie nowego zakupu**, **PU Obliczanie wartosci rachunku**.

Dalej pokazano diagram sekwencji operacji **Szukaj\_rachunek** i moment wykonania **zdarzenia equals** na obiekcie typu **TRachunek**.

2.1. Diagram sekwencji operacji **Szukaj\_rachunek**, która prowadzi do wywołanie zdarzenia **equals** na obiekcie typu **TRachunek**



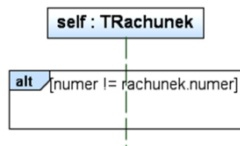
**Zdarzenie equals** na obiekcie typu **TRachunek** w metodzie **indexOf** obiektu Rachunki typu **ArrayList** (p.2.2-2.4)

2.2. Kod metody **indexOf**, wywołanej z obiektu Rachunki typu **ArrayList**

```
public int indexOf(Object o) {
    if (o == null) {
        for (int i = 0; i < size; i++)
            if (elementData[i]==null)
                return i;
    } else {
        for (int i = 0; i < size; i++)
            if (o.equals(elementData[i]))
                return i;
    }
    return -1;
}
```

**Zdarzenie equals** na obiekcie typu **TRachunek** w metodzie **indexOf** obiektu Rachunki typu **ArrayList**

2.3. Diagram sekwencji metody **equals** (zdarzenie) klasy **TRachunek**



**Dozór** zdarzenia **equals** na obiekcie typu **TRachunek**.  
**Akcją** zdarzenia **equals** na obiekcie typu **TRachunek** – zwrócenie wyniku **true** lub **false** jako wyniku dozoru.

2.4. Kod metody **equals** klasy **TRachunek**

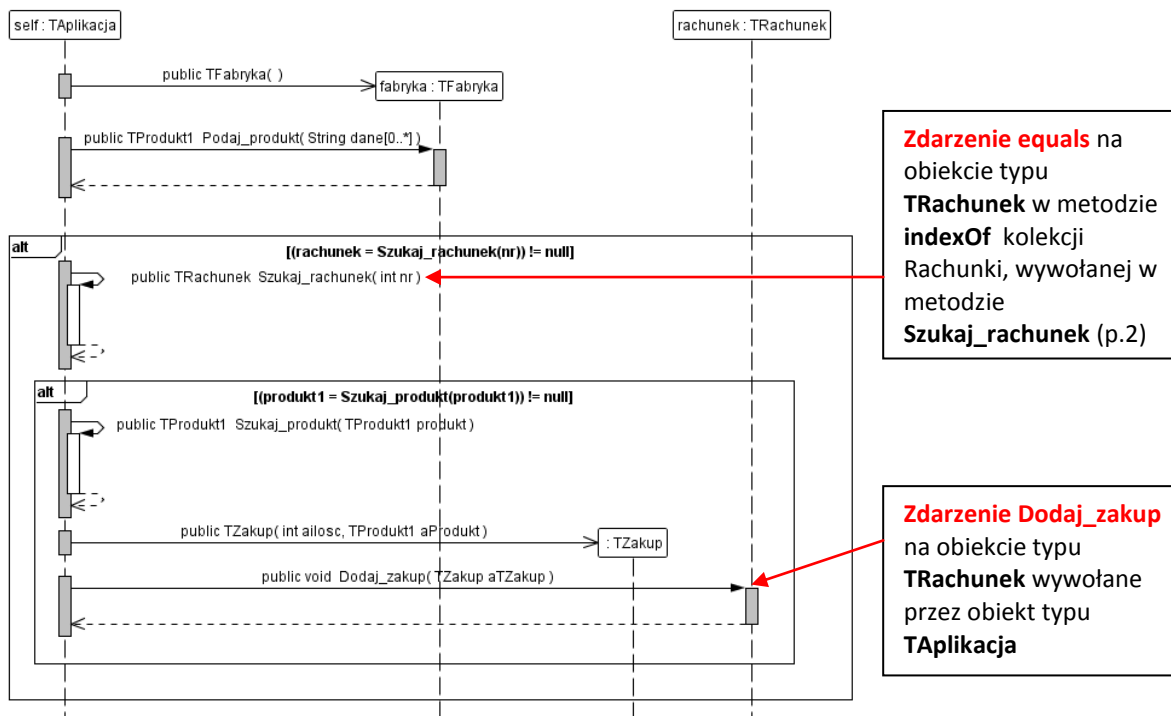
```
public boolean equals (Object aTRachunek)
{
    TRachunek rachunek= (TRachunek)aTRachunek;
    boolean bStatus = true;
    if ( numer!= rachunek.numer )
        bStatus = false;
    return bStatus;
}
```

**Dozór** zdarzenia **equals** na obiekcie typu **TRachunek**

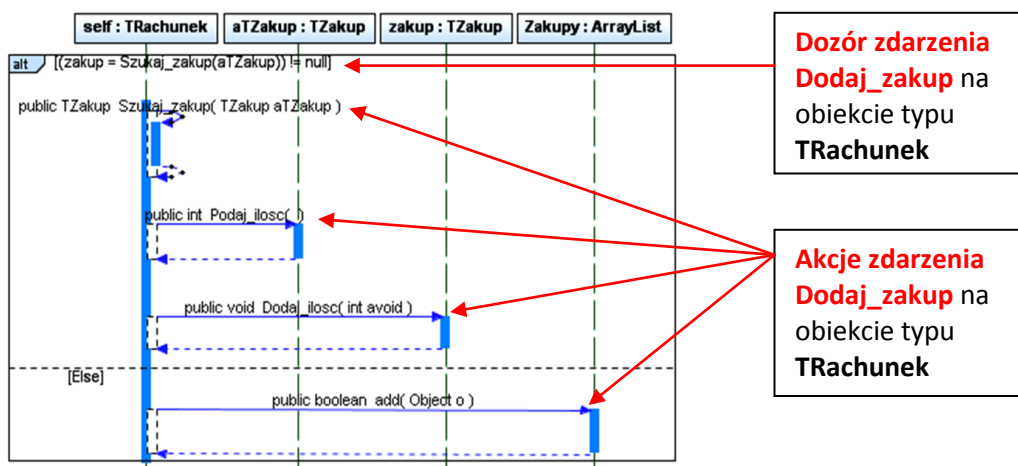
**Akcja** zdarzenia **equals** na obiekcie typu **TRachunek** – zwrócenie wyniku dozoru **true** lub **false**. Pominięto domyślne akcje typu przypisanie wartości do zmiennej oraz konwersję typu danych.

3. **Zdarzenia equals** i **Dodaj\_zakup** (PU Wstawianie nowego zakupu )

3.1. Diagram sekwencji operacji **Wstaw\_zakup** z klasy **TAplikacja** prezentujący momenty wywołania **zdarzenia equals** na obiekcie typu **TRachunek** przez obiekt typu **Rachunki** (p.2) i **zdarzenia Dodaj\_zakup** na obiekcie typu **TRachunek** przez obiekt typu **TAplikacja**.

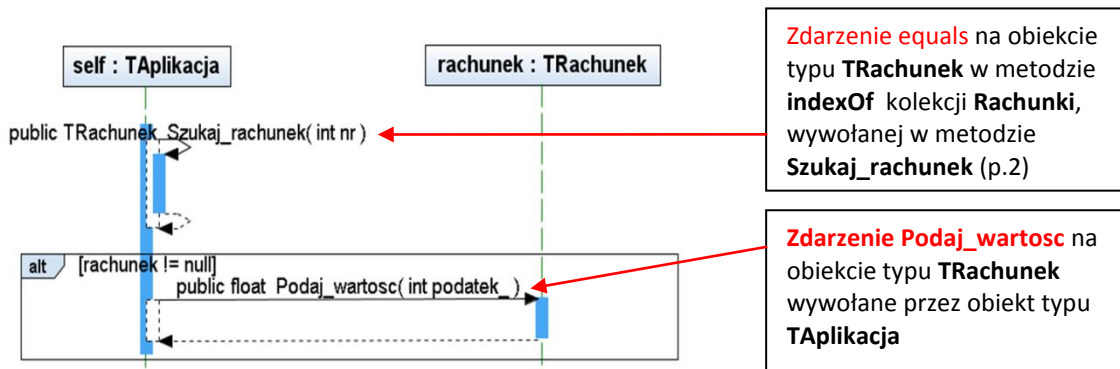


3.2. Diagram sekwencji operacji **Dodaj\_zakup** klasy **TRachunek** jako **zdarzenie** wywołane na obiekcie **TRachunek** przez obiekt typu **TAplikacja** – operacje służące do definicji tej operacji są **akcjami zdarzenia Dodaj\_zakup**.

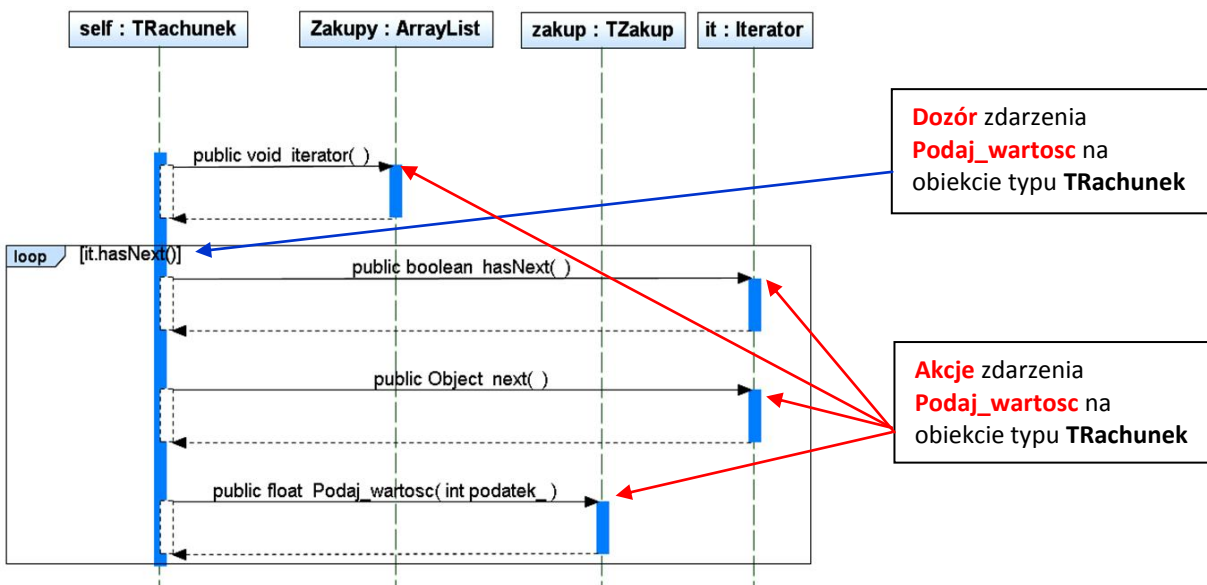


4. **Zdarzenia equals i Podaj\_wartosc** (PU Obliczanie wartości rachunku )

4.1. Diagram sekwencji operacji **Podaj\_wartosc** z klasy **TAplikacja** prezentujący momenty wywołania **zdarzenia equals** na obiekcie typu **TRachunek** przez obiekt typu **Rachunki** (p.2) i **zdarzenia Podaj\_wartosc** na obiekcie typu **TRachunek** przez obiekt typu **TAplikacja**.



4.2. Diagram sekwencji operacji **Podaj\_wartosc** klasy **TRachunek** jako **zdarzenie** wywołane na obiekcie **TRachunek** przez obiekt typu **TAplikacja** – operacje służące do definicji tej operacji są **akcjami zdarzenia Podaj\_wartosc**.



## **Dodatek 2.**

Sposób tworzenia diagramów stanów w środowisku Visual Paradigm wg [Drawing state machine diagrams](http://www.visual-paradigm.com/support/documents/vpumluserguide/94/2579/6714_drawingstate.html).

([http://www.visual-paradigm.com/support/documents/vpumluserguide/94/2579/6714\\_drawingstate.html](http://www.visual-paradigm.com/support/documents/vpumluserguide/94/2579/6714_drawingstate.html))