

Podstawy modelowania obiektowego

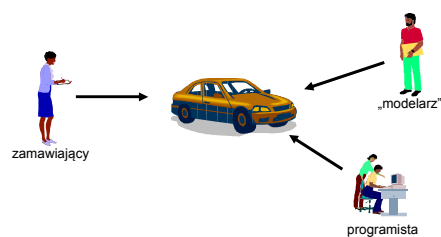
- Jak modelować świat obiektowo?
- Obiekty
- Klasy
- Diagramy języka UML

Modelowanie oprogramowania w języku UML, Michał Śmiełek

Obiekty jako wspólny język w projekcie

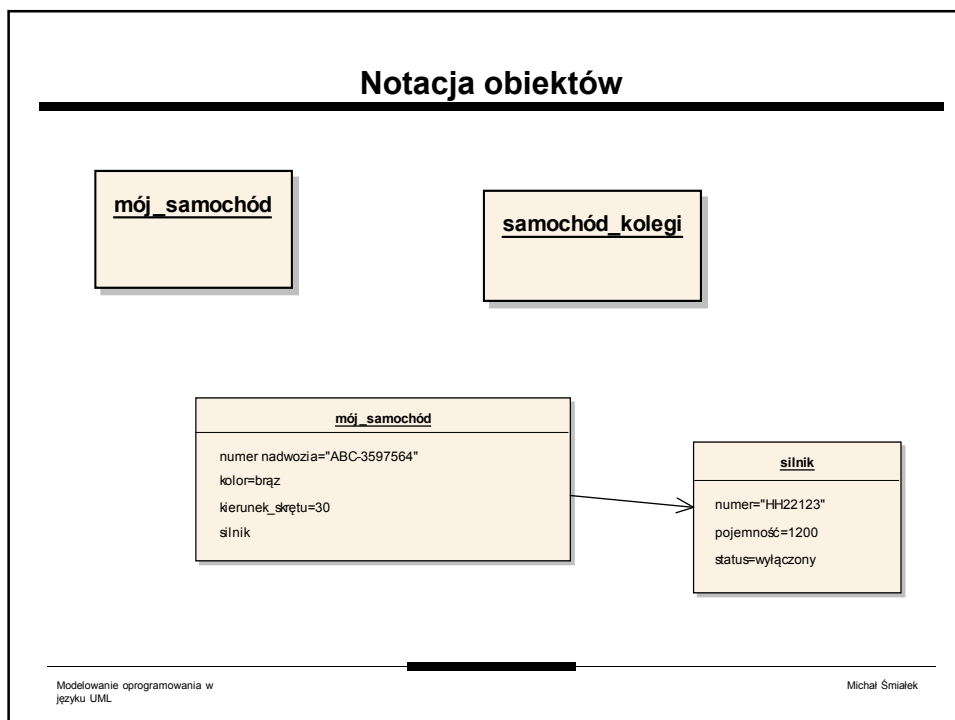
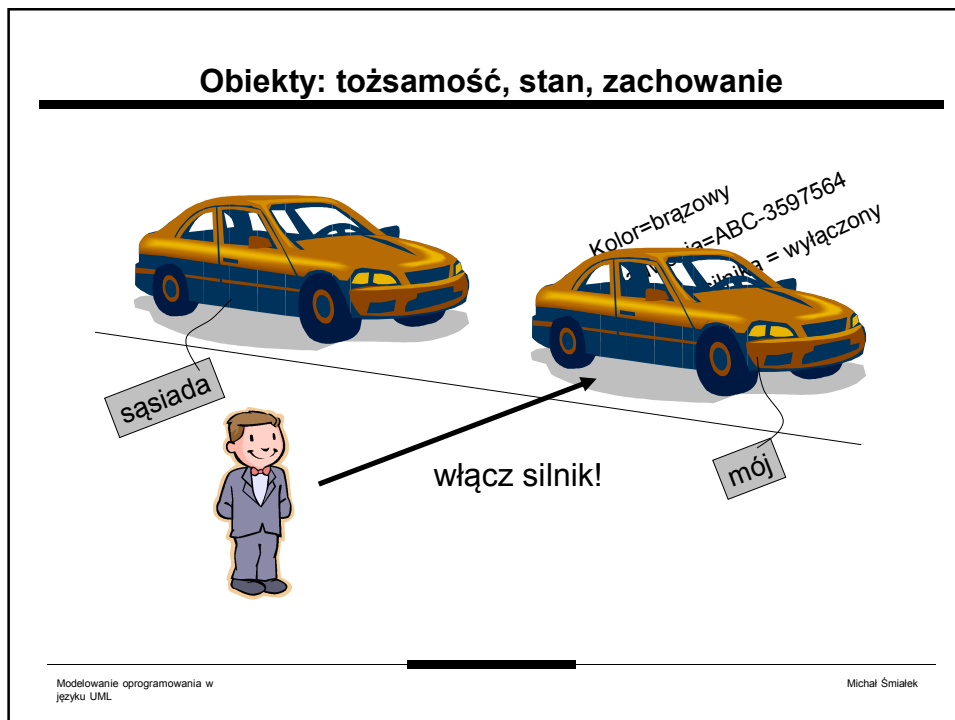
Modelowanie obiektowe polega na:

- znajdowaniu obiektów w naszym otoczeniu,
- opisywaniu struktury i dynamiki działania obiektów,
- klasyfikacji obiektów,
- opisywaniu struktury powiązań klas obiektów, oraz
- opisywaniu dynamiki współpracy obiektów podczas realizacji funkcjonalności systemu



Modelowanie oprogramowania w języku UML

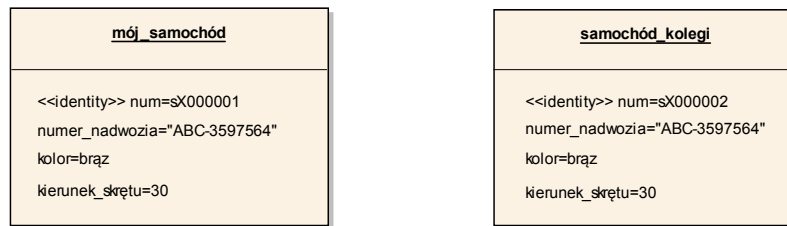
Michał Śmiełek



Rozróżnianie tożsamości obiektów

Stan obiektu to zbiór wartości (cech charakterystycznych) wszystkich jego właściwości. Stan obiektu zmienia się w czasie.

Tożsamość obiektu wyróżnia obiekt wśród innych obiektów jako osobną jednostkę. Tożsamość jest wyróżnioną cechą obiektu, która pozostaje niezmienna przez cały czas życia tego obiektu.



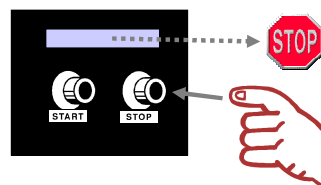
Zachowanie obiektów

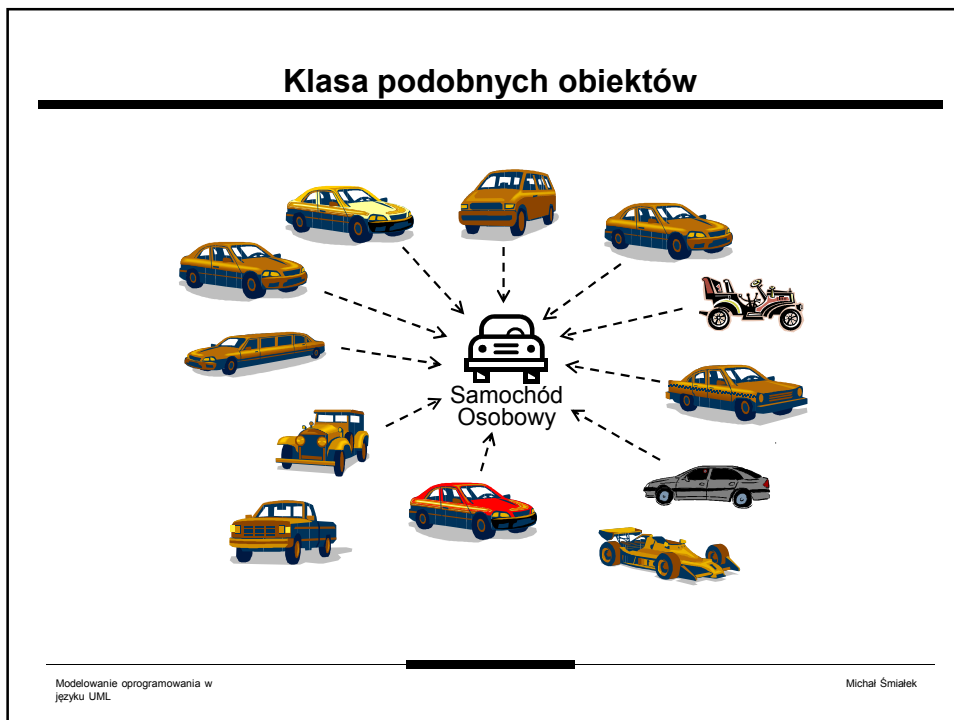
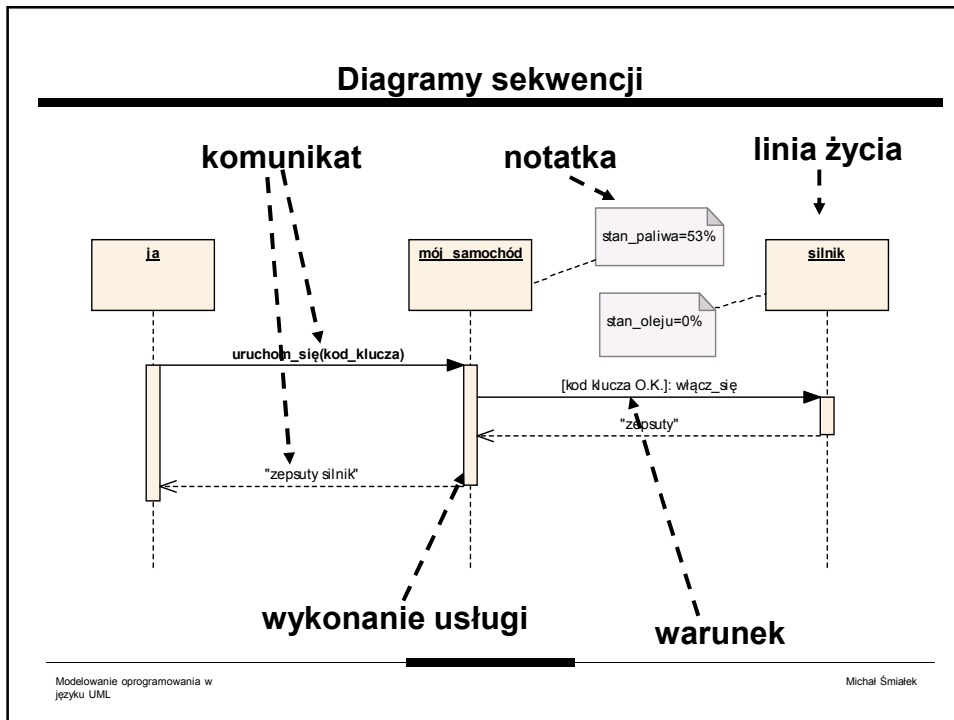
Zachowanie obiektu to zbiór usług, które obiekt potrafi wykonywać na rzecz innych obiektów.

Obiekty mogą prosić inne obiekty o wykonanie odpowiednich usług. Obiekt reaguje na taką prośbę, jeżeli usługa jest w zbiorze obsługiwanych przez niego usług. Prośby obiektów o wykonanie usług będziemy nazywali **komunikatami**.

Po wykonaniu usługi może nastąpić:

- Zmiana stanu obiektu
- Dostarczenie wyniku przetwarzania

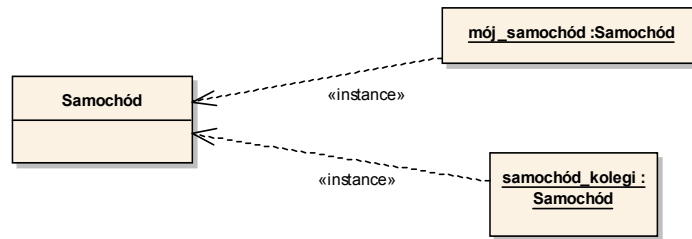




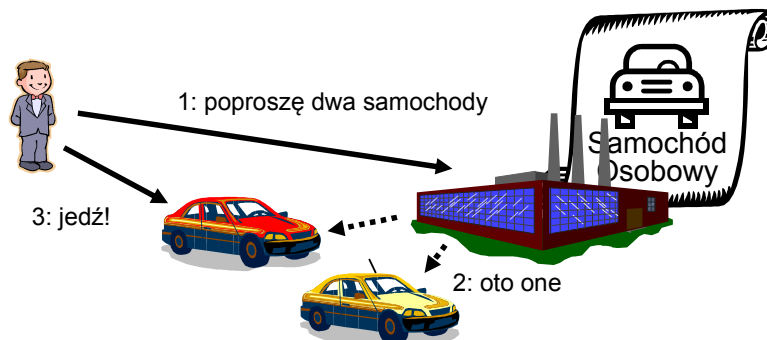
Klasa jako typ dla obiektów

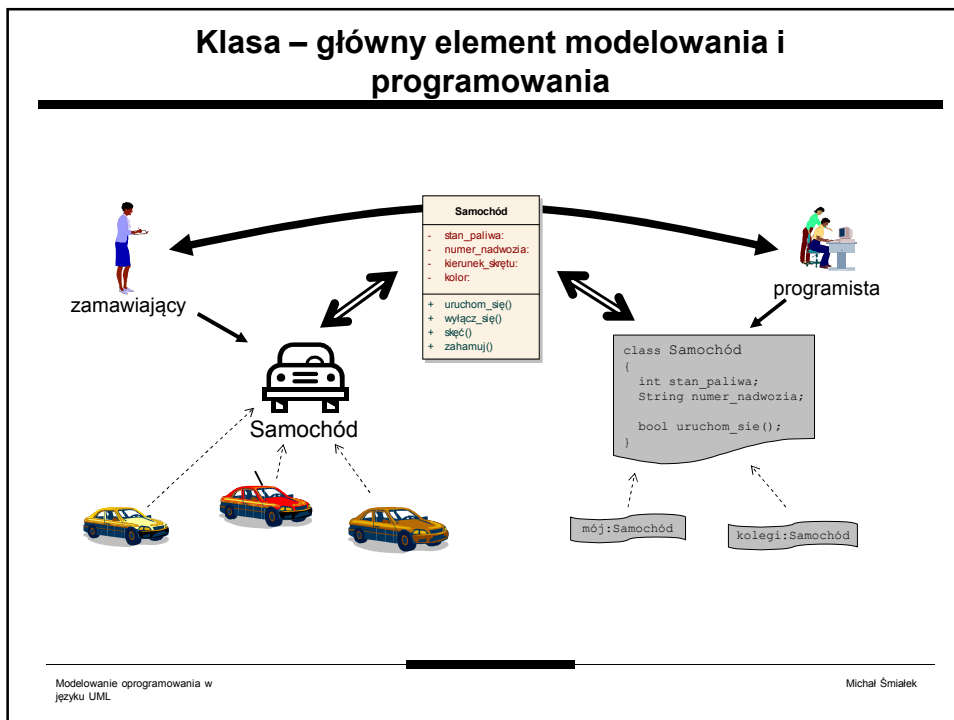
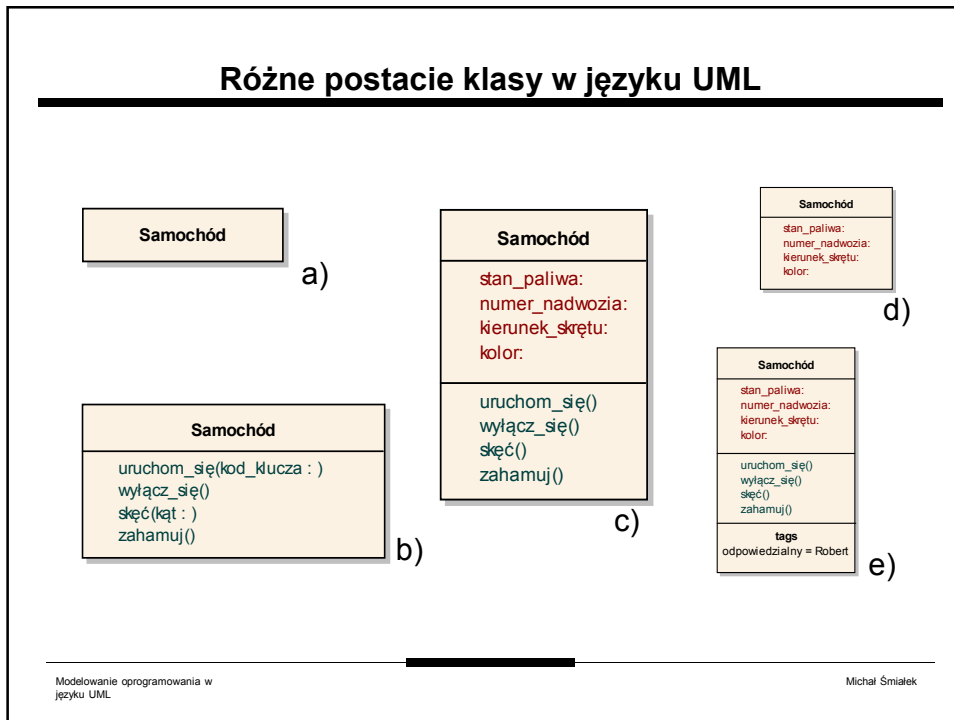
Klasa jest opisem grupy obiektów o jednakowym zestawie właściwości i sposobie zachowania. Opis klasy stanowi pewnego rodzaju wzornik dla tworzenia obiektów tej klasy (tzw. instancji klasy). Ten wzornik zawiera:

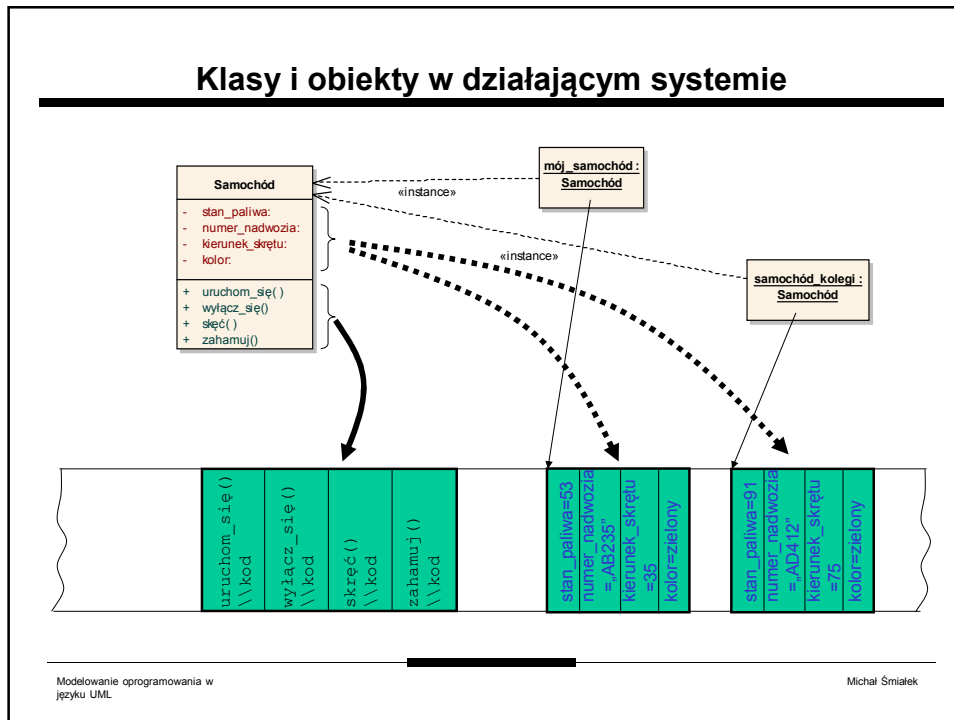
- nazwę klasy,
- zestaw właściwości jednakowych dla wszystkich obiektów klasy,
- zestaw usług obsługiwanych przez wszystkie obiekty klasy.



Klasa jako „fabryka” obiektów







Ćwiczenie

Proszę narysować diagram klas oraz diagramy sekwencji (funkcje „main” i „on_click” dla poniższego kodu.

```

class wnd : event {
    cntrl c;
    wnd(cntrl cn) {
        c = cn;
    }
    void on_click() {
        c.run();
    }
    void show(int n) {
        printf(n);
    }
}
        
```

```

class cntrl {
    wnd w; dom d;
    cntrl(dom dm) {
        d = dm;
        d.init();
    }
    void set_wnd(wnd wn) {
        w = wn;
    }
    void run() {
        num = d.get_next();
        w.show(num);
    }
    void show(int n) {
        printf(n);
    }
}
        
```

```

class dom {
    int number;
    void init() {
        number = 0;
    }
    int get_next() {
        number = number + 1;
        return number;
    }
}

main() {
    dom d = new dom();
    cntrl c = new cntrl(d);
    wnd w = new wnd(c);
    UI.set_event(wnd);
}
        
```

Modelowanie oprogramowania w języku UML Michał Śmiłek

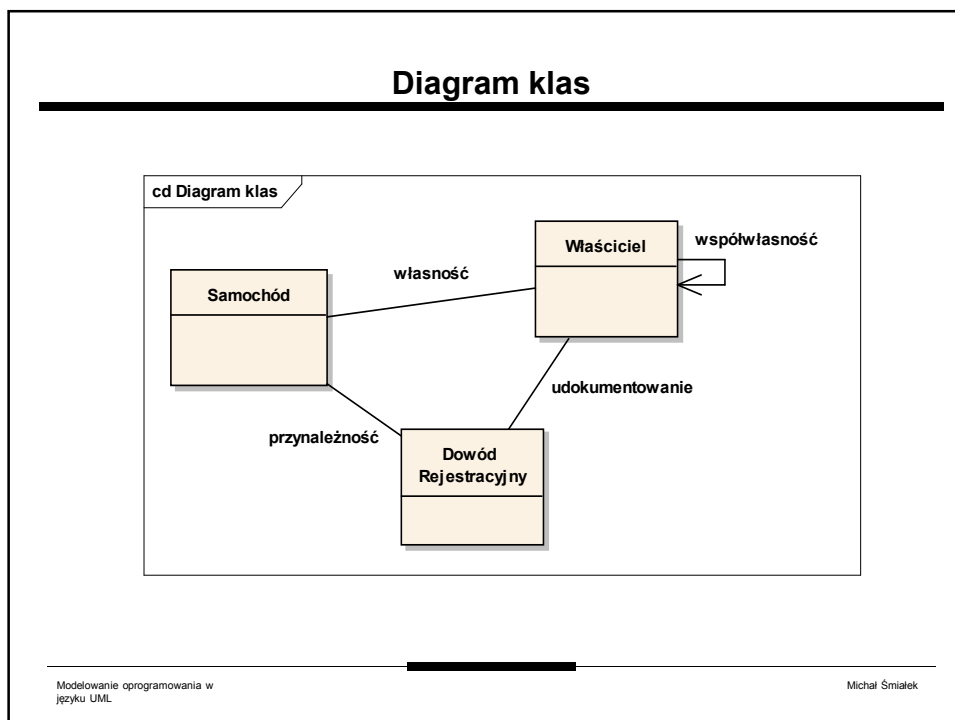
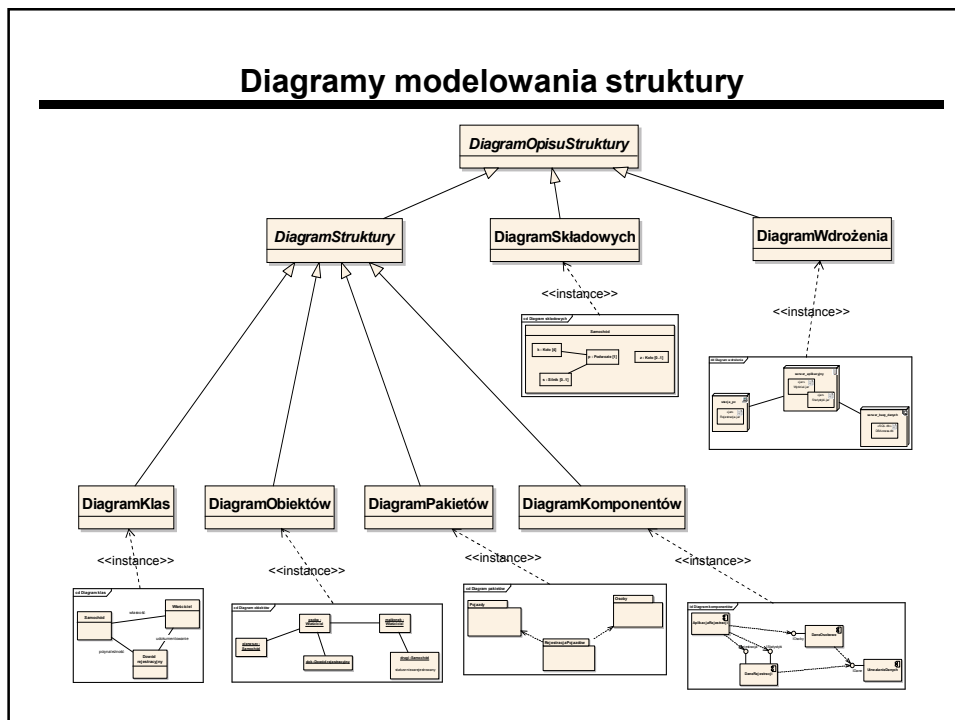
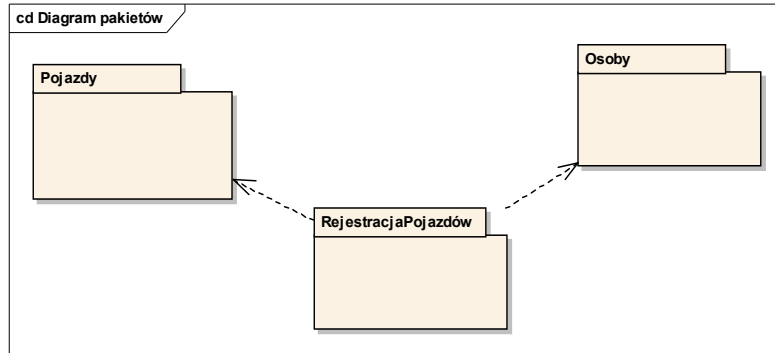


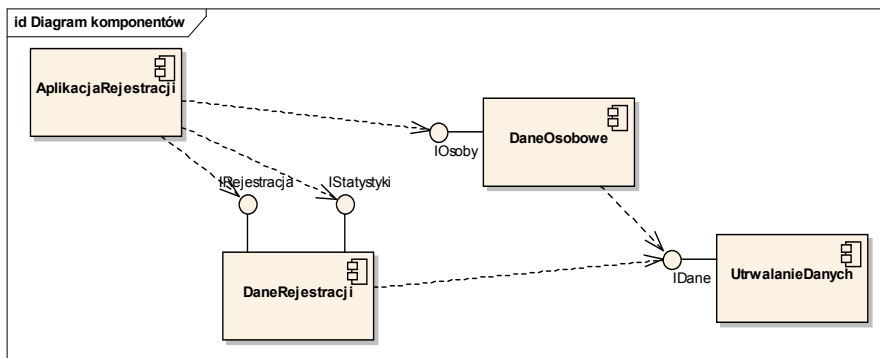
Diagram pakietów



Modelowanie oprogramowania w języku UML

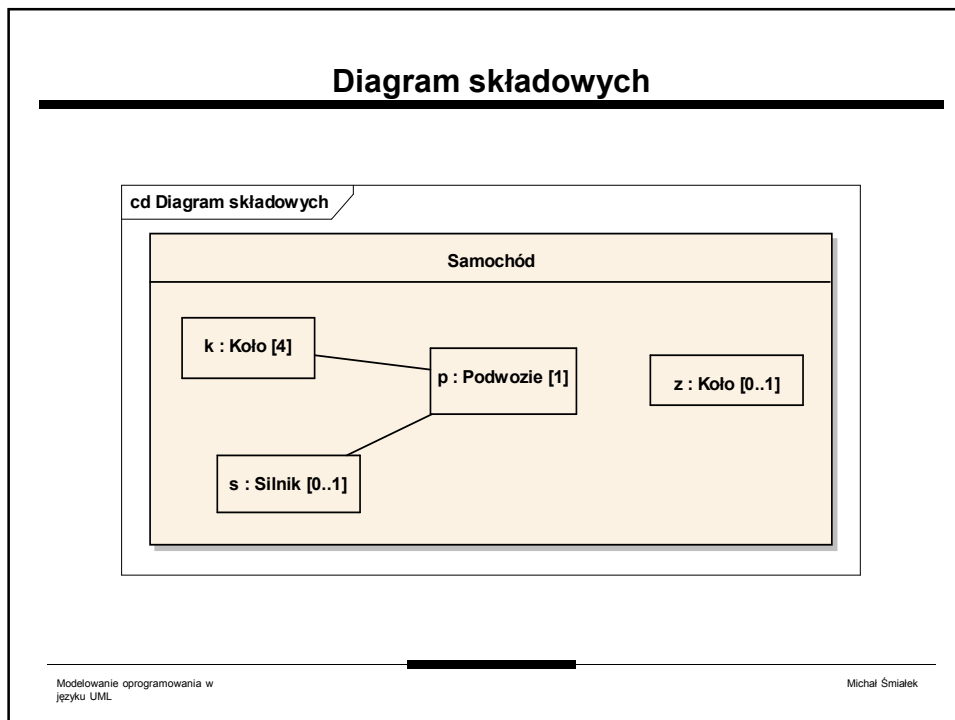
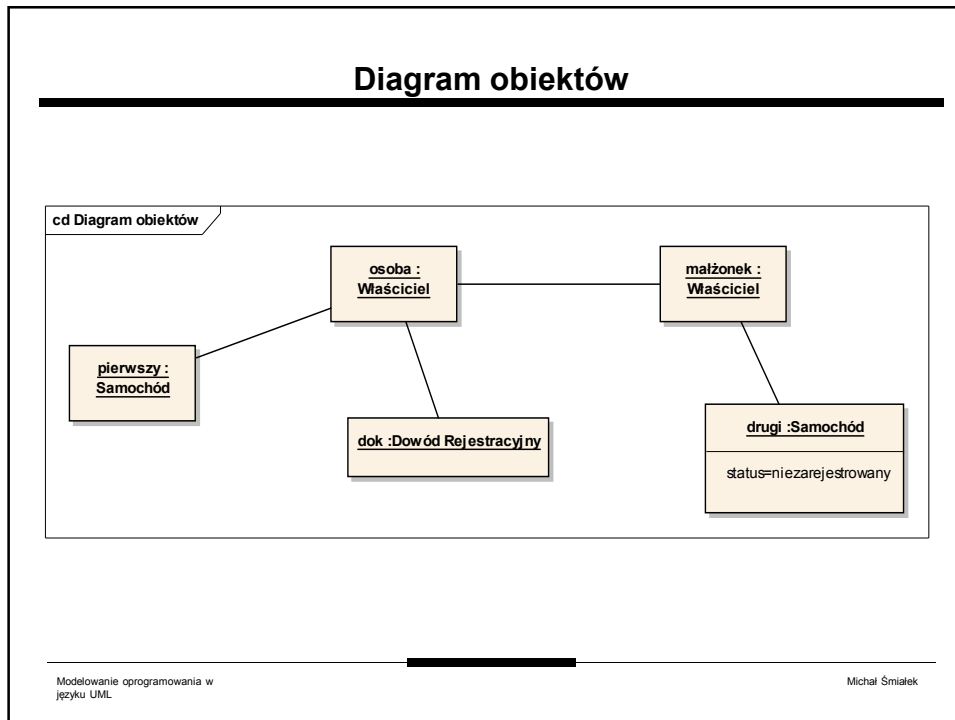
Michał Śmiałek

Diagram komponentów



Modelowanie oprogramowania w języku UML

Michał Śmiałek



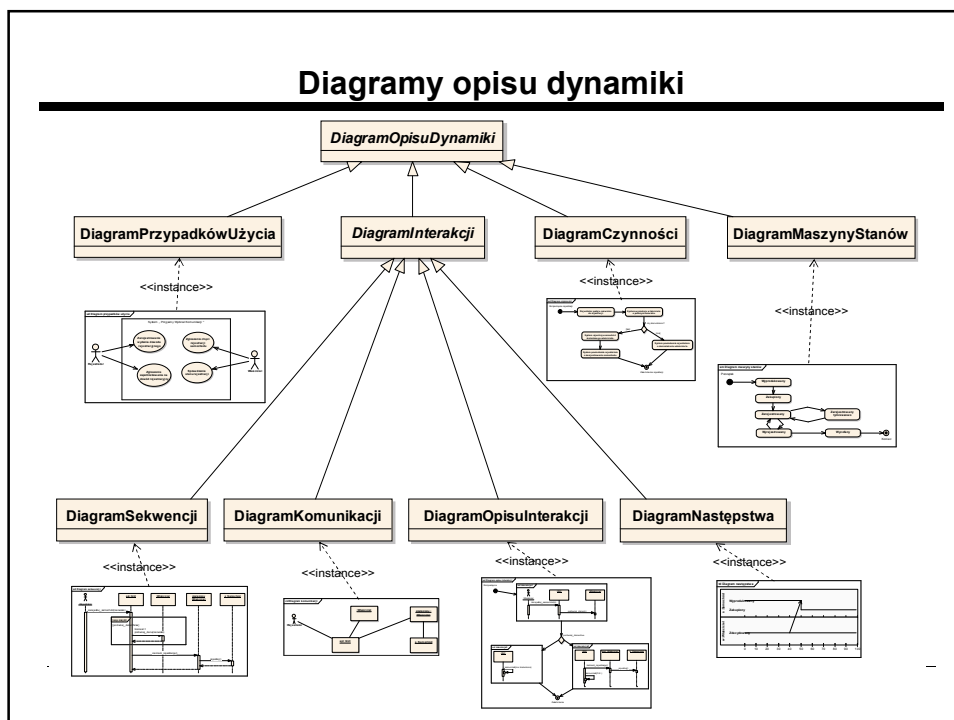
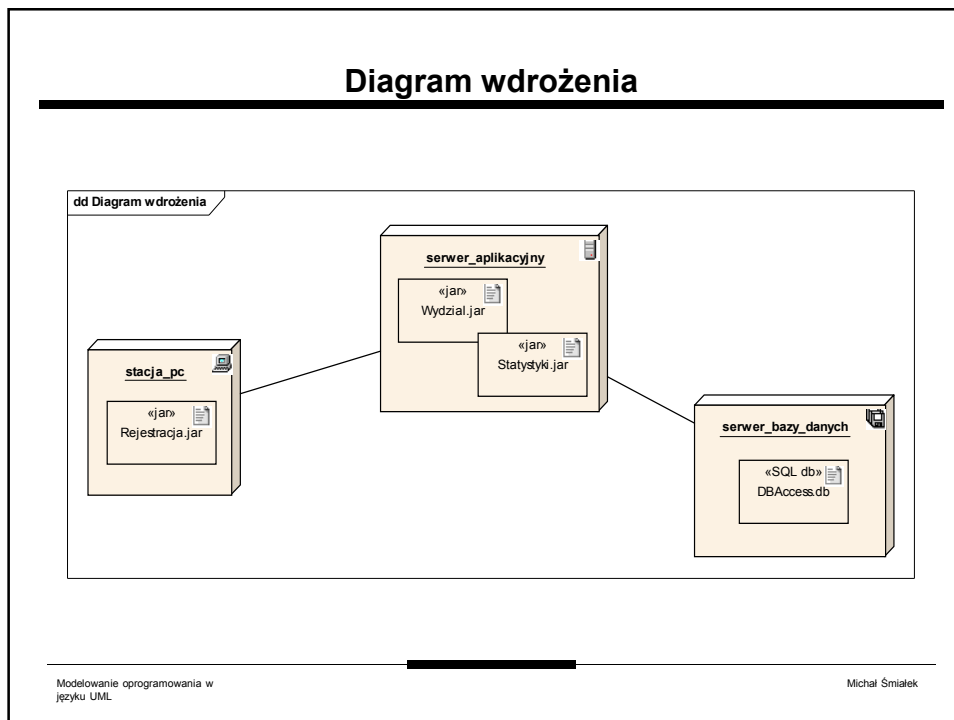
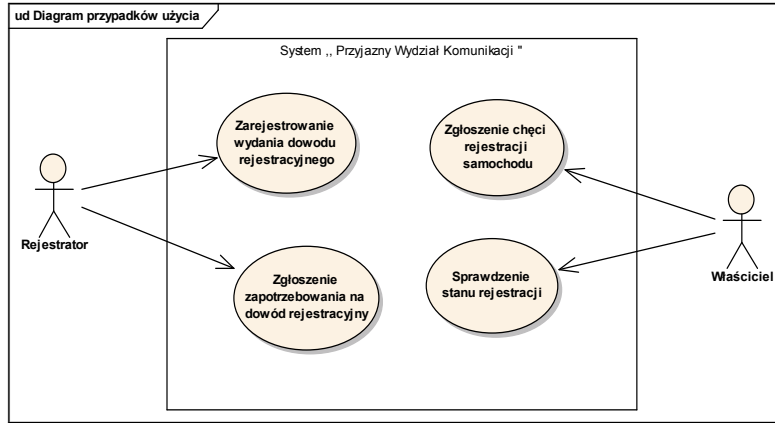


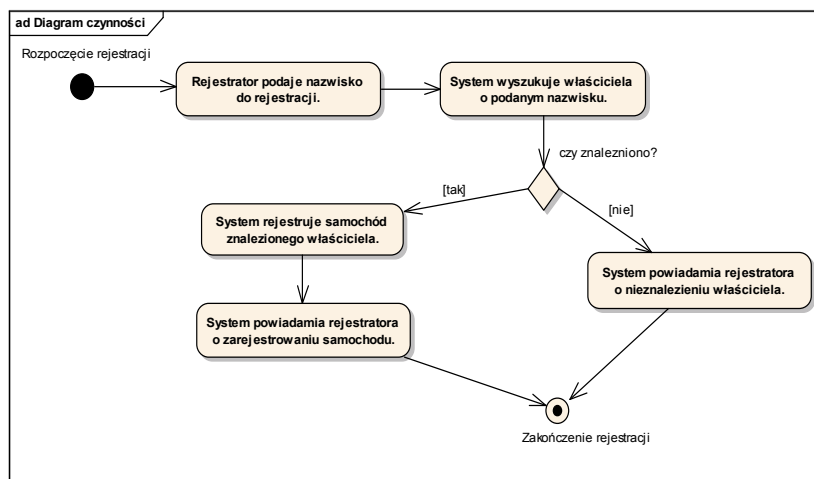
Diagram przypadków użycia



Modelowanie oprogramowania w języku UML

Michał Śmiełek

Diagram czynności



Modelowanie oprogramowania w języku UML

Michał Śmiełek

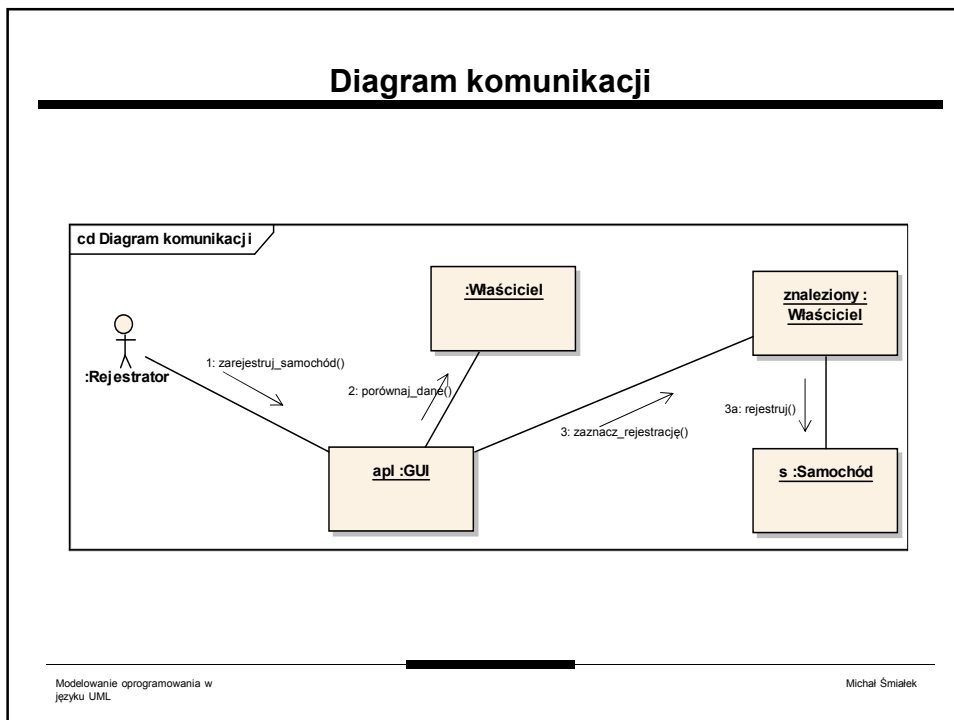
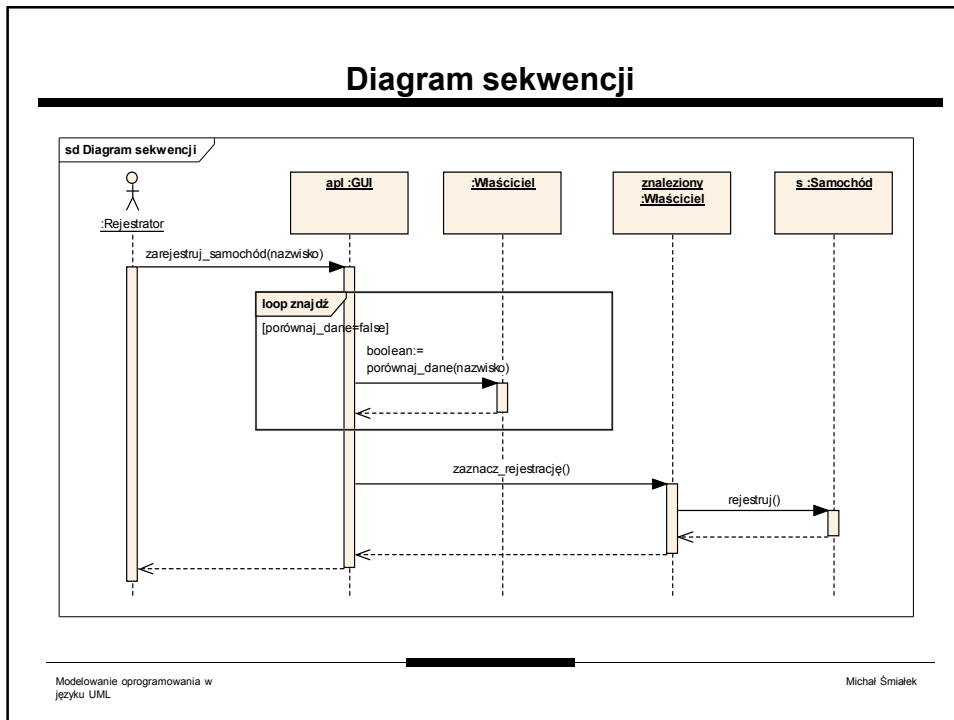
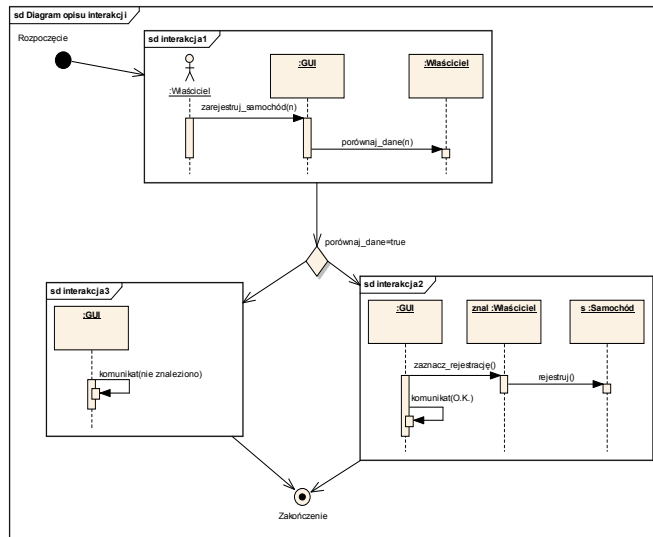


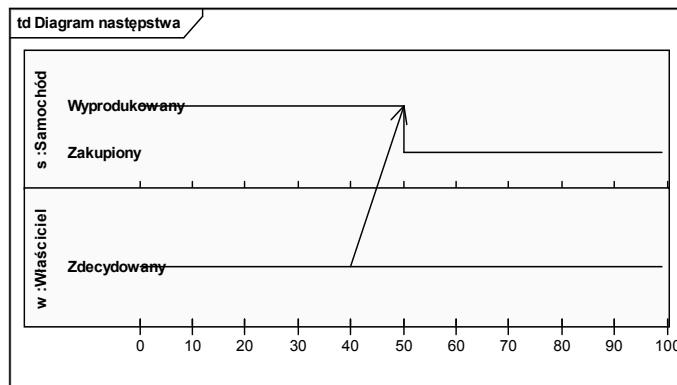
Diagram opisu interakcji



Modelowanie oprogramowania w języku UML

Michał Śmiełek

Diagram następstwa



Modelowanie oprogramowania w języku UML

Michał Śmiełek

