

Modelowanie oprogramowania w języku UML

- Regulamin przedmiotu
- Literatura
- Wprowadzenie
- Treść przedmiotu

Modelowanie oprogramowania w języku UML, Michał Śmiełek

Informacje o przedmiocie

Strona wykładowcy: <http://www.iem.pw.edu.pl/~smialek/>

Zawiera:

- Aktualne ogłoszenia dla studentów (w zakładce „Dydaktyka”)
- Przykłady do książki „Zrozumieć UML 2.0” (w pliku .EAP)

Strona przedmiotu: <http://wikidyd.iem.pw.edu.pl/index.cgi/UML1>

Zawiera:

- Dokładny regulamin zaliczania
- Literaturę
- Konspekty kolejnych wykładów.
- Konspekty kolejnych ćwiczeń laboratoryjnych

Modelowanie oprogramowania w języku UML

Michał Śmiełek

Zasady zaliczenia wykładu

Wykład

- Ocena za egzamin: max. 55 pkt.
- Punkty dodatkowe za ćwiczenia na wykładzie.
- Konkurs na najlepsze notatki.
- Możliwość zwolnienia z egzaminu lub podwyższenia oceny dla osób, które wygrają konkurs!

Ćwiczenia na wykładzie

- 5 ćwiczeń wykonywanych na zakończenie wybranych wykładów (na razie nie wiadomo których... ☺)

Notatki

- Notatki wykonane na wykładzie, zeskanowane do formatu PDF.
- Podpis wykładowcy.
- Terminy oddania: co ok. 4 wykłady (będą podane tydzień wcześniej).
- Od kiedy? Od wykładu 3.

Zasady zaliczenia laboratorium

Laboratorium

- Ocena za laboratorium: max. 45 pkt.
- Ściśle powiązane z wykładem
- 4 małe ćwiczenia wprowadzające za 2 pkt
- 3 duże ćwiczenia (po 3 zajęcia) za 12-13 pkt.
- Obecność obowiązkowa – ocena wystawiana bezpośrednio na zajęciach

Więcej informacji na zajęciach laboratoryjnych.

Zasady wyliczania oceny końcowej

```
if (w >= 28 and l >= 23 and w+l >= 51) {
  suma = w+l+c+n;
  case suma:
    suma < 61
      ocena = 3
      break;
    suma < 71
      ocena = 3,5
      break;
    suma < 81
      ocena = 4
      break;
    suma < 91
      ocena = 4,5
      break;
    suma <= 100
      ocena = 5
      break;
  }
}
```

w - punkty za wykład
l - punkty za laboratorium
c - punkty za ćwiczenia (ekstra)
n - punkty za notatki (ekstra)

Literatura

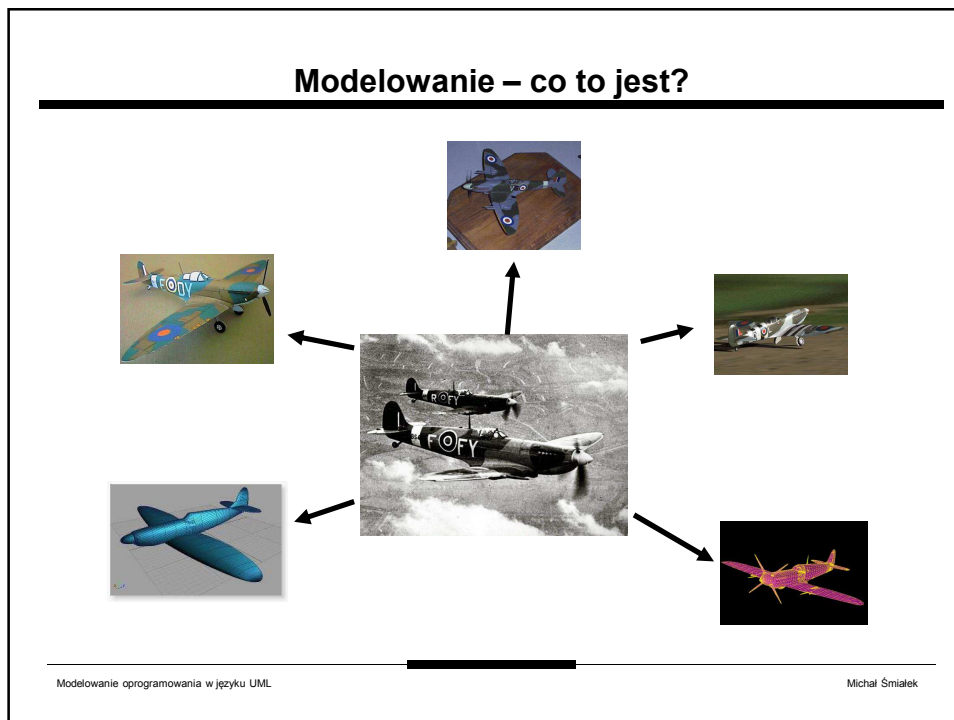
Podstawowy podręcznik:

- Michał Śmiałek, „Zrozumieć UML 2.0”, Helion, 2005

Książki dodatkowe:

- R. S. Pressman, „Praktyczne podejście do inżynierii oprogramowania”, Wydawnictwa Naukowo-Techniczne, 2004
- I. Sommerville, „Inżynieria oprogramowania”, Wydawnictwa Naukowo-Techniczne, 2003
- M. Fowler, „UML 2.0 w kropelce”, LTP, 2005
- T. Pender, „UML Bible”, Wiley, 2003

- Inne książki na temat języka UML...



Dlaczego modelowanie oprogramowania?

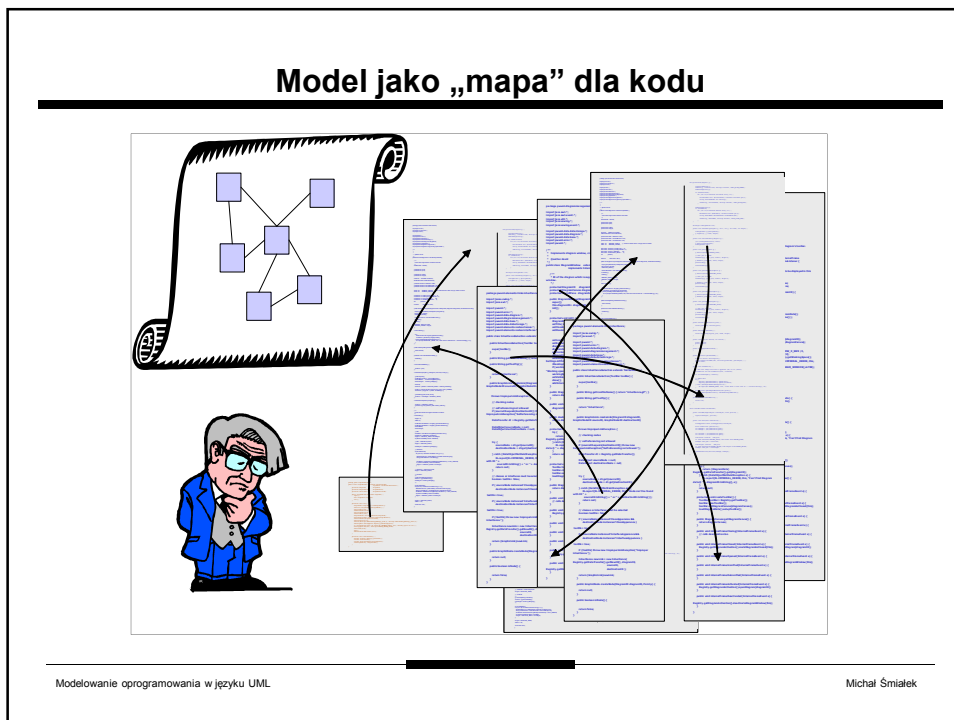
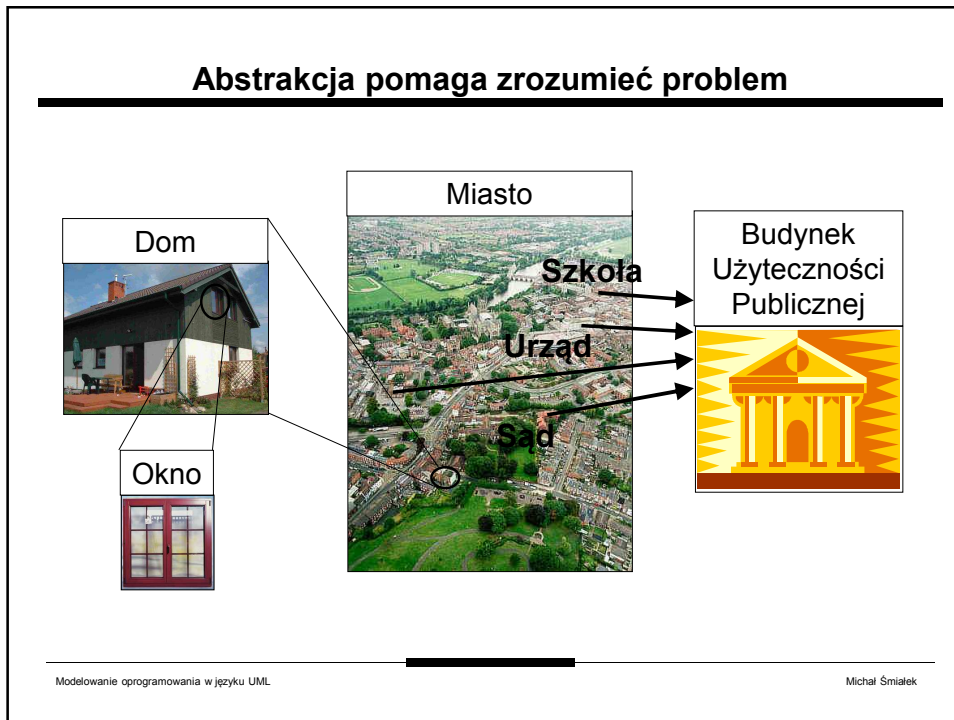
Gerald M. Weinberg, w swojej książce zatytułowanej *Zrozumieć Profesjonalnego Programistę* napisał tak:

„(...) programowanie komputerów jest zdecydowanie najtrudniejszym zadaniem intelektualnym, jakie ludzie próbowali kiedykolwiek wykonywać. Kiedykolwiek.”

Zatem:

- **Potrzebujemy metod na pokonanie tej złożoności!**
- **Modelowanie pokonuje złożoność poprzez dodanie opisu umożliwiającego zastosowanie zasady abstrakcji.**

Modelowanie oprogramowania w języku UML Michał Śmiałek



Po co to wszystko?

Aby wytarzać systemy:

- Na czas
- W budżecie
- Spełniające **rzeczywiste** potrzeby klienta

Modelowanie oprogramowania w języku UML Michał Śmiątek

Problemy, problemy...

Niezadowoleni zamawiający: „nie o taki system nam chodziło”, „żaden z naszych urzędników nie będzie w stanie zrozumieć jak to działa”.

Niezadowoleni wykonawcy: „to właściwie, jaki system oni od nas chcą?”, „przecież tak ciężko się napracowaliśmy”.

Kłótnie o zakres systemu: „żądacie od nas systemu dwa razy większego niż było zapisane w umowie!”, „przecież te funkcje miały być uwzględnione dopiero w następnej wersji”.

Chaotyczna obsługa zmian wymagań: „koledzy, dodajcie nam tu jeszcze taką małą tabelkę na środku, to was prawie nic nie będzie kosztowało”, „myśleliśmy, że ta zmiana nie będzie miała dla was istotnego znaczenia”.

Niewyspani programiści: „poprosimy pizzę pod drzwi o północy i dużo Jolt Coli”, „dajcie nam dziesięciu nowych programistów”.

Stres pod koniec projektu: „ten system działa jak żółw!”, „przecież tu brakuje połowy funkcjonalności, która jest zapisana w par. 7 pkt. 223 umowy!”.

Brak powtarzalności procesu – wyniki zawsze są trudne do przewidzenia: „to o ile tym razem przekroczymy budżet?”, „właściwie to jak będziemy tym razem pisać te wymagania?”.

Godzenie się na „marsz ku klęsce”: „musicie ten system zrobić dwa razy szybciej niż konkurencja; O.K., nie ma sprawy”.

Modelowanie oprogramowania w języku UML Michał Śmiątek

Lekarstwo? Metodyka!

Patrz: wykład z Podstaw inżynierii oprogramowania (2 semestr)

Metodyka to:

- Notacja - język modelowania, przy pomocy którego zapisuje się produkty procesu tworzenia oprogramowania (modele) – **UML!**
- Techniki - zbiór sposobów postępowania w celu stworzenia konkretnych elementów modeli (zapisanych przy pomocy notacji).
- Proces techniczny - uporządkowanie wytwarzania oprogramowania poprzez wyodrębnienie atomowych zadań do wykonania (przy użyciu technik) i określenie kolejności i zależności między zadaniami.

Co więcej:

- Narzędzia CASE (będziemy je używać na laboratorium)
- Integracja modelowania i programowania (generowanie kodu – będzie na laboratorium)

Co dalej? (1)

Podstawy modelowania obiektowego

- Obiekty, klasy, interakcje między obiektami, przegląd rodzajów diagramów

Model przypadków użycia

- Notowanie wymagań dla systemu oprogramowania

Model czynności – scenariusze

- Notowanie szczegółów dla przypadków użycia
- Scenariusze tekstowe (rozszerzenie UML)
- Scenariusze w formie diagramów

Model klas

- Obiekty – UML jest obiektowy!
- Klasy na poziomie wymagań (model dziedziny problemu)
- Klasy na poziomie projektu (mapa kodu)

Co dalej? (2)

Model interakcji

- Model działania systemu jako współdziałania (interakcji) między obiektami
- Diagramy sekwencji – komunikaty, warunki, pętle (kod w formie diagramu!)

Model komponentów i model montażu

- Znacznie ułatwiają pokazanie struktury bardziej złożonych systemów (kilkadziesiąt klas i więcej)
- Na przykład: jak pokazać system składający się z tysiąca klas w Javie??!!

Modelowanie w procesie wytwarzania oprogramowania

- Umiejscowienie języka UML w procesie technicznym
- Nawiązanie do przedmiotu „Podstawy inżynierii oprogramowania” z 2 semestru

Modelowanie oprogramowania w języku UML

Michał Śmiełek

Schemat wykładu

Krótkie wprowadzenie do notacji

- Przedstawienie głównych elementów języka UML i ich semantyki (znaczenia)

Aktywna demonstracja

- Ćwiczenie wykonywane wspólnie ze studentami
- Symulacje działania systemu „na żywo”
- Projektowanie systemu „na żywo”
- Generacja kodu „na żywo”
- itp...

Ćwiczenie

- Po niektórych wykładach
- Krótkie zadanie do wykonania na podstawie bieżącego wykładu (max. 15 min)

Modelowanie oprogramowania w języku UML

Michał Śmiełek

Ćwiczenie (1)

Proszę przeanalizować kod i spróbować narysować wyjaśnienie, jak ten kod działa.

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

UWAGA: kod ten nie jest pisany w jakimś konkretnym języku, ale przypomina Javę lub C#!

```
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 Śmiełek

Ćwiczenie (2)

Wskazówki:

- Proszę w jakiś sposób oznaczyć klasy i ich składniki (atrybuty i operacje)
- Proszę w jakiś sposób oznaczyć kolejność (sekwencję) wywołań dla trzech obiektów (w, c, d) podczas realizacji funkcji „main” oraz podczas i po wywołaniu funkcji „on_click”.

Najważniejsze:

- Rysunki powinny być „mapą” dla kodu, tak, aby łatwiej go można było zrozumieć i się po nim poruszać.

Rozwiązanie w języku UML? Za tydzień! Na razie proszę wymyślić dowolny sposób rysowania „obrazków”.

Modelowanie oprogramowania w języku UML

Michał Śmiełek