Informatyka

Wykład 4

Witold Dyrka
witold.dyrka@pwr.wroc.pl

2/04/2012

Uwaga!

Bardzo ważną częścią tego wykładu
jest zapis wideo „jak to zrobić?”
(przy użyciu programów: CamStudio, VideoDub, AVI ReComp)

Znajdziesz go na stronie:
http://www.youtube.com/playlist?list=PL3DFA2DF771AA1B1D

Wideo jest czytelne:
przy rozdzielczości HD
na pełnym ekranie

Program wykładów

0. Informatyka. Wprowadzenie do Matlaba (13.02.12)
1. Matlab dla programistów C/C++ (20.02.12)
2. Optymalizacja obliczeń. Grafika (05.03.12)
3. Złożone typy danych. Programowanie zorientowane obiektowo (OOP) (19.03.12)
4. **OOP część 2** **(02.04.12)**
5. Graficzny interfejs użytkownika (16.04.12)
6. Obliczenia numeryczne (30.04.12)
7. Kolokwium (14.05.12)

Dzisiejszy wykład w oparciu o...

- B. Mrozek, Z. Mrozek. MATLAB i Simulink. Poradnik użytkownika. Wydanie III. Helion 2010. Rozdział 5.
- MATLAB Product Documentation. Object-Oriented Programming. http://www.mathworks.com/help/techdoc/matlab_oop/ug_intropage.html
- Dave Foti. Inside MATLAB Objects in R2008a. Matlab Digest. http://www.mathworks.com/tagteam/50350_91586v00_Digest_OOP_FINAL.pdf
- Scott Gorlin. Advanced Matlab. OOPs, I wrote it again..., Advanced OOP. http://www.scottgorlin.com/category/teaching/

Dwa tygodnie temu...

- **Programowanie zorientowane obiektowo**
  - tworzenie klas w Matlabie
  - własności (zmienne) i metody (funkcje)
  - enkapsulacja

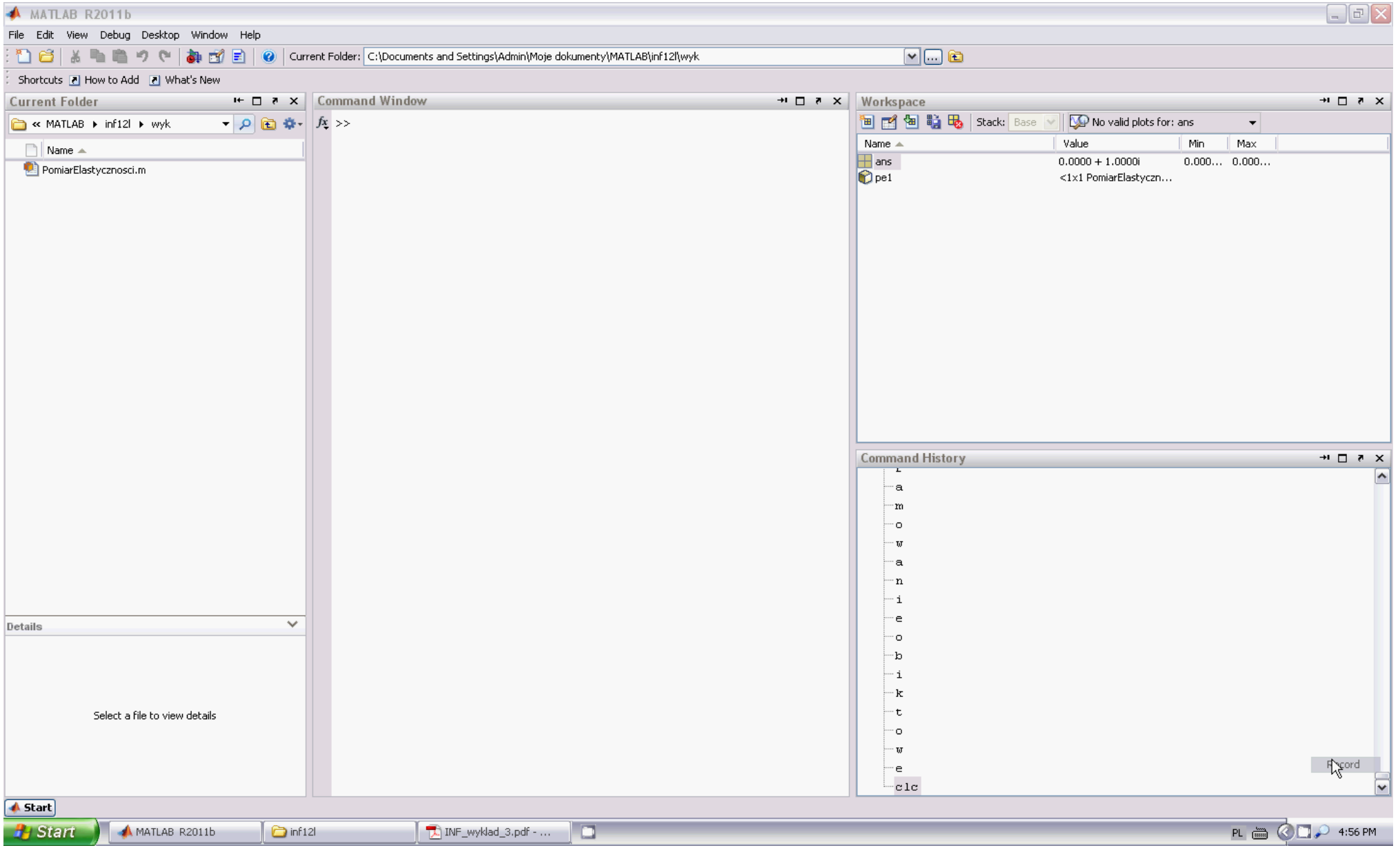
Program na dziś

- **Programowanie zorientowane obiektowo** (część 2)
  - weryfikacja poprawności obiektu
  - dziedziczenie
  - polimorfizm
  - klasy „referencyjne”

Tak to wygląda teraz

```
classdef PomiarElastycznosci
    %POMIARELASTYCZNOSCI przechowuje pomiary elastycznosci ...

    properties
      Material
      NumerProbki
      Odksztalcenie
      Naprezenie
    end % properties

    properties (Dependent = true, SetAccess = private)
       Modul
    end % Dependent + set-private properties

    methods
      function pe = PomiarElastycznosci(material,numprob,naprez,odkszt) % ...

      function obj = set.Material(obj,material)
         if ~(strcmpi(material,'aluminum') ||...
            strcmpi(material,'stal nierdzewna') ||...
            strcmpi(material,'stal niestopowa'))
            error('Akceptowany material to aluminium...')
         end
      obj.Material = material;
      end % set.Material

      function modul = get.Modul(obj)
        ind = find(obj.Odksztalcenie > 0); % Tylko niezerowe odksztalcenie
        modul = mean(obj.Naprezenie(ind)./obj.Odksztalcenie(ind));
      end % get.Modul

    end % methods

end
```

Problem weryfikacji liczby pomiarów



Problem weryfikacji liczby pomiarów (2)

- **konstruktor** sprawdza czy nie podano więcej wartości odkształcenia niż naprężenia – bardzo dobrze

  ale

- **metoda `set`**, np. dla `Odksztalcenie`, nie sprawdza długości wektora `Naprezenie`, bo nie ma gwarancji kolejności w jakiej Matlab ładuje obiekt – wielka szkoda

- Oczywiście, nie godzimy się na tolerowanie błędnego stanu obiektu:

```
>> pe.Odksztalcenie = [.12 .20 .31];
>> pe.Naprezenie = [2e4 4e4];
>> pe.Modul
Index exceeds matrix dimensions.
Error in PomiarElastycznosci/get.Modul (line 44)
     modul = mean(obj.Naprezenie(ind)./obj.Odksztalcenie(ind));
```

  ale

- nie chcemy też „prywatyzować” `Odksztalcenie` i `Naprezenie`, żeby nie rezygnować z wygodnego operowania na wektorach

Weryfikacja liczby pomiarów – rozwiązanie

- Ukrywamy prawdziwe wektory pomiarów

```
properties (Access = private)
  Odksztalcenie1
  Naprezenie1
end % private properties
```

- Dajemy do nich dostęp przez własności zależne

```
properties (Dependent = true)
  Odksztalcenie
  Naprezenie
end % dependent properties
```

- Kiedy użytkownik zmienia `Odksztalcenie`,
  - metoda `set.Odksztalcenie` zmienia prywatne `Odksztalcenie1`
    - ale tylko jeśli wektory mają prawidłowe długości

Weryfikacja liczby pomiarów – rozw. (2)

MATLAB R2011b

File Edit Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« MATLAB ▸ inf12l ▸ wyk

Name

PomiarElastycznosci.m

Command Window

```
>> help PomiarElastycznosci
 PomiarElastycznosci przechowuje pomiary elastycznosci i liczy modu
  Konstruktor: pe = PomiarElastycznosci(material,numprob,naprez,odk
   material - nazwa materialu
   numprob  - numer badanej probki
   naprez   - wektor naprezen
   odkszt   - wektor odksztalcen

>> pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12 0.
>> pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12 0.
Error using PomiarElastycznosci (line 25)
Liczba pomiarow odksztalcenia nie moze byc mniejsza niz liczba
pomiarow naprezenia

>> pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12 0.
>> pe1.Odksztalcenie = [0.12 0.20 0.31];
>> pe1.Modul
Index exceeds matrix dimensions.

Error in PomiarElastycznosci/get.Modul (line 44)
        modul =
        mean(obj.Naprezenie(ind)./obj.Odksztalcenie(ind));

>>
```

Workspace

Stack: Base | No valid plots for: ans

Name	Value	Min	Max
ans	0.0000 + 1.0000i	0.000...	0.000...
pe1	<1x1 PomiarElastyczn...		

Command History

```
l
e
o
b
i
k
t
o
w
e
clc
help PomiarElastycznosci
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1.Odksztalcenie = [0.12 0.20 0.31];
pe1.Modul
```

Record

PomiarElastycznosci.m (MATLAB Class)

przechowuje pomiary elastycznosci i liczy modul

- Material
- NumerProbki
- Odksztalcenie
- Naprezenie
- Modul
- PomiarElastycznosci(material, numprob, naprez, od...
- set.Material(obj, material)
- get.Modul(obj)

Start

Start | MATLAB R2011b | inf12l | INF_wyklad_3.pdf - ... | PL | 4:59 PM

Weryfikacja liczby pomiarów – testowanie

MATLAB R2011b

File Edit View Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« MATLAB ▸ inf12l ▸ wyk

Name

PomiarElast.m

PomiarElastycznosci.m

PomiarElastycznosci.m (MATLAB Class)

przechowuje pomiary elastycznosci i liczy modul

- Material
- NumerProbki
- Odksztalcenie
- Naprezenie
- Modul
- PomiarElastycznosci(material, numprob, naprez, od...
- set.Material(obj, material)
- get.Modul(obj)

Command Window

```
>> help PomiarElastycznosci
 PomiarElastycznosci przechowuje pomiary elastycznosci i liczy modu
  Konstruktor: pe = PomiarElastycznosci(material,numprob,naprez,odk
   material - nazwa materialu
   numprob  - numer badanej probki
   naprez   - wektor naprezen
   odkszt   - wektor odksztalcen

>> pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12 0.
>> pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12 0.
Error using PomiarElastycznosci (line 25)
Liczba pomiarow odksztalcenia nie moze byc mniejsza niz liczba
pomiarow naprezenia

>> pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12 0.
>> pe1.Odksztalcenie = [0.12 0.20 0.31];
>> pe1.Modul
Index exceeds matrix dimensions.

Error in PomiarElastycznosci/get.Modul (line 44)
        modul =
        mean(obj.Naprezenie(ind)./obj.Odksztalcenie(ind));

fx >>
```

Workspace

Stack: Base | No valid plots for: ans

Name	Value	Min	Max
ans	0.0000 + 1.0000i	0.000...	0.000...
pe1	<1x1 PomiarElastyczn...		

Command History

```
l
e
o
b
i
k
t
o
w
e
clc
help PomiarElastycznosci
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1.Odksztalcenie = [0.12 0.20 0.31];
pe1.Modul
```

Record

Start | MATLAB R2011b | Editor - C:\Document... | inf12l | INF_wyklad_3.pdf - ... | PL | 5:04 PM

Podział klasy

- Klasa `PomiarElast` zrobiła się całkiem skomplikowana
  - 4 rodzaje własności
  - 9 metod
- Co więcej, klasa łączy
  - funkcje związane z modułem sprężystości `set.Material,get.Modul`
  - z ogólną obsługą pomiarów `set/get.Odksztalcenie` `set/get.Naprezenie`
- Czas ją podzielić !

```
classdef PomiarElast
    %POMIARELAST przechowuje pomiary elastycznosci i liczy modul%.

    properties... % properties

    properties (Dependent = true)... % dependent properties

    properties (Access = private)... % private properties

    properties (Dependent = true, SetAccess = private)... % Depenc

    methods
      function pe = PomiarElast(material,numprob,naprez,odkszt)

      function obj = set.Material(obj,material)... % set.Material

      function obj = set.Odksztalcenie(obj, odkszt)...

      function odkszt = get.Odksztalcenie(obj)...

      function obj = set.Naprezenie(obj, naprez)...

      function naprez = get.Naprezenie(obj)...

      function modul = get.Modul(obj)... % get.Modul

      function disp(obj)... % disp

      function plot(obj,varargin)... % plot

    end % methods

end
```

Klasa `Pomiar`

- własności % czyli zmienne
  - `Nazwa` -nazwa badania (np. nazwa materialu)
  - `NumerPom` -numer badania (np. numer badanej próbki)
  - `Zadane` -wektor wartości niezależnych (zadanych)
  - `Zmierzone` -wektor wartości zależnych (zmierzonych)
- metody % czyli funkcje
  - `Pomiar` -konstruktor
  - `set.Zadane` -weryfikacja wektora zadanych
  - `get.Zadane` -odczyt wektora zadanych
  - `set.Zmierzone` -weryfikacja wektora zmierzonych
  - `get.Zmierzone` -odczyt wektora zmierzonych
  - `plot` -rysowanie wykresu

Klasa `Pomiar` – implementacja

MATLAB R2011b

File Edit View Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« MATLAB ▸ inf12l ▸ wyk

Name

- PomiarElast.m
- PomiarElastycznosci.m

PomiarElast.m (MATLAB Class)

przechowuje pomiary elastycznosci i liczy modul

- Material
- NumerProbki
- Odksztalcenie
- Naprezenie
- Odksztalcenie1
- Naprezenie1
- Modul
- PomiarElast(material, numprob, naprez, odkszt)

Command Window

```
This is a Classroom License for instructional use only.
Research and commercial use is prohibited.
>>
```

Workspace

Stack: Base | Select data to plot

Name	Value	Min	Max

Command History

```
k
t
o
w
e
clc
help PomiarElastycznosci
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1.Odksztalcenie = [0.12 0.20 0.31];
pe1.Modul
pe2 = PomiarElast('stal nierdzewna',1,[2e4 4e4],[0.12 0.20]);
pe2.Odksztalcenie = [0.12 0.20 0.31];
pe2.Odksztalcenie
pe2.Odksztalcenie1
%-- 3/29/2012 3:08 PM --%
```

Record

Start Ready

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | PL | 3:19 PM

Klasa `Pomiar` – testowanie

MATLAB R2011b

File Edit Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« MATLAB ▸ inf12l ▸ wyk

Name

- Pomiar.asv
- Pomiar.m
- PomiarElast.m
- PomiarElastycznosci.m

PomiarElast.m (MATLAB Class)

przechowuje pomiary elastycznosci i liczy modul

- Material
- NumerProbki
- Odksztalcenie
- Naprezenie
- Odksztalcenie1
- Naprezenie1
- Modul
- PomiarElast(material, numprob, naprez, odkszt)

Command Window

```
>>
```

Workspace

Stack: Base | Select data to plot

Name	Value	Min	Max

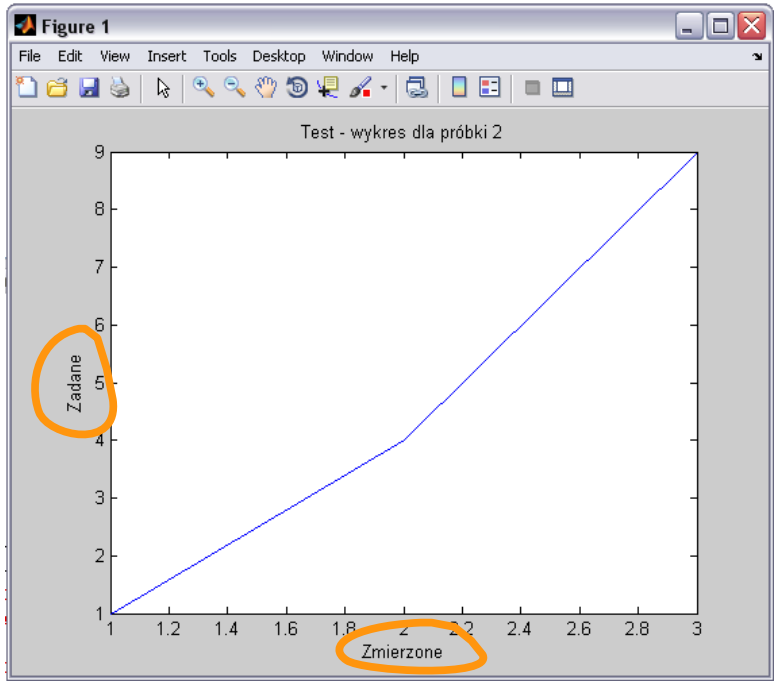
Command History

```
k
t
o
w
e
clc
help PomiarElastycznosci
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1.Odksztalcenie = [0.12 0.20 0.31];
pe1.Modul
pe2 = PomiarElast('stal nierdzewna',1,[2e4 4e4],[0.12 0.20]);
pe2.Odksztalcenie = [0.12 0.20 0.31];
pe2.Odksztalcenie
pe2.Odksztalcenie1
%-- 3/29/2012 3:08 PM --%
```

Record

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | Editor | PL | 3:31 PM

Ulepszanie wykresu



- Chyba warto by móc określać rodzaj i jednostki pomiarów?
  - dodajmy własności przechowujące te informacje

Własności niemodyfikowalne

- własności klasy `Pomiar`
  - `Nazwa` –nazwa badania (np. nazwa materialu)
  - `NumerPom` -numer badania (np. numer badanej próbki)
  - **`ZmNiezal` -nazwa i jednostki zmiennej niezależnej (zadanej, np. 'Naprezenie [psi]')**
  - `Zadane` -wektor wartości niezależnych (zadanych)
  - **`ZmZal` -nazwa i jednostki zmiennej zależnej (zmierzonej, np. 'Odksztalcenie')**
  - `Zmierzone` -wektor wartości zależnych (zmierzonych)

- Rozsądne założenie:
  - rodzaj i jednostki zmiennych są ustawiane w konstruktorze
    - nie można ich później modyfikować – są <u>niemodyfikowalne</u> (ang. *immutable*)

Klasa `Pomiar` – zmienne niemodyfikowalne

MATLAB R2011b

Figure 1

Test - wykres dla próbki 2

Zmierzone

Zadane

```
>> p = Pomia
Error using
Liczba wart
wartosci za

>> p = Pomia
>> p.Zmierz
Error using
Liczba wart
wartosci za

>> p.Zadane
Error using
Liczba wart
wartosci za

>> disp(p)
  Pomiar

  Properties
        Naz
     NumerP
   Zmierzone: [1 4 9]
      Zadane: [1 2 3]

  Methods

>> plot(p)
Undefined function 'numstr' for input arguments of type 'char'.

Error in Pomiar/plot (line 67)
        title([numstr(obj.Nazwa), ' - wykres dla próbki',...

>> plot(p)
>> plot(p)
>>
```

```
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1.Odksztalcenie = [0.12 0.20 0.31];
pe1.Modul
pe2 = PomiarElast('stal nierdzewna',1,[2e4 4e4],[0.12 0.20]);
pe2.Odksztalcenie = [0.12 0.20 0.31];
pe2.Odksztalcenie
pe2.Odksztalcenie1
%-- 3/29/2012 3:08 PM --%
p = Pomiar('Test',2,[1 2 3],[1 4 9 10]);
p = Pomiar('Test',2,[1 2 3],[1 4 9]);
p.Zmierzone = [1 4 9 10];
p.Zadane = [1 4];
disp(p)
plot(p)
```

Pomiar.m (MATLAB Class)

do zmiany!!

Nazwa
NumerPom
Zmierzone
Zadane
Zmierzone1
Zadane1
Pomiar(nazwa, numpom, zadane, zmierzone)

Klasa `Pomiar` – testowanie i wykańczanie

MATLAB R2011b

File Edit Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« MATLAB ▸ inf12l ▸ wyk

Name

- Pomiar.asv
- Pomiar.m
- PomiarElast.m
- PomiarElastycznosci.m

Pomiar.m (MATLAB Class)

do zmiany!!

- Nazwa
- NumerPom
- ZmNiezal
- ZmZal
- Zadane
- Zmierzone
- Zmierzone1
- Zadane1

Command Window

```
>>
```

Workspace

Stack: Base | Select data to plot

Name	Value	Min	Max

Command History

```
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1 = PomiarElastycznosci('stal nierdzewna',1,[2e4 4e4],[0.12
pe1.Odksztalcenie = [0.12 0.20 0.31];
pe1.Modul
pe2 = PomiarElast('stal nierdzewna',1,[2e4 4e4],[0.12 0.20]);
pe2.Odksztalcenie = [0.12 0.20 0.31];
pe2.Odksztalcenie
pe2.Odksztalcenie1
%-- 3/29/2012 3:08 PM --%
p = Pomiar('Test',2,[1 2 3],[1 4 9 10]);
p = Pomiar('Test',2,[1 2 3],[1 4 9]);
p.Zmierzone = [1 4 9 10];
p.Zadane = [1 4];
disp(p)
plot(p)
clc
```

Record

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | Editor | Bez tytułu - Notatnik | PL | 3:48 PM

Czego nauczyliśmy się pisząc klasę `Pomiar`?

- Jak abstrahować klasę ogólną (`Pomiar`)
  - ze szczegółowej (`PomiarElast`)
- Jak utworzyć własności niemodyfikowalne

```
properties (SetAccess = immutable)
```

- Nigdy dość testowania i pielęgnacji !

Kolejny krok – dziedziczenie

- Czas wykorzystać klasę `Pomiar`
  - do napisania nowej wersji klasy `PomiarElast`
    - aby nazwy się nie myliły, nową wersję nazwiemy `PomiarSprez`

- Poznamy mechanizm **dziedziczenia**
  - `PomiarSprez` dziedziczy własności i metody `Pomiar`-u
    - oraz dodaje swoje

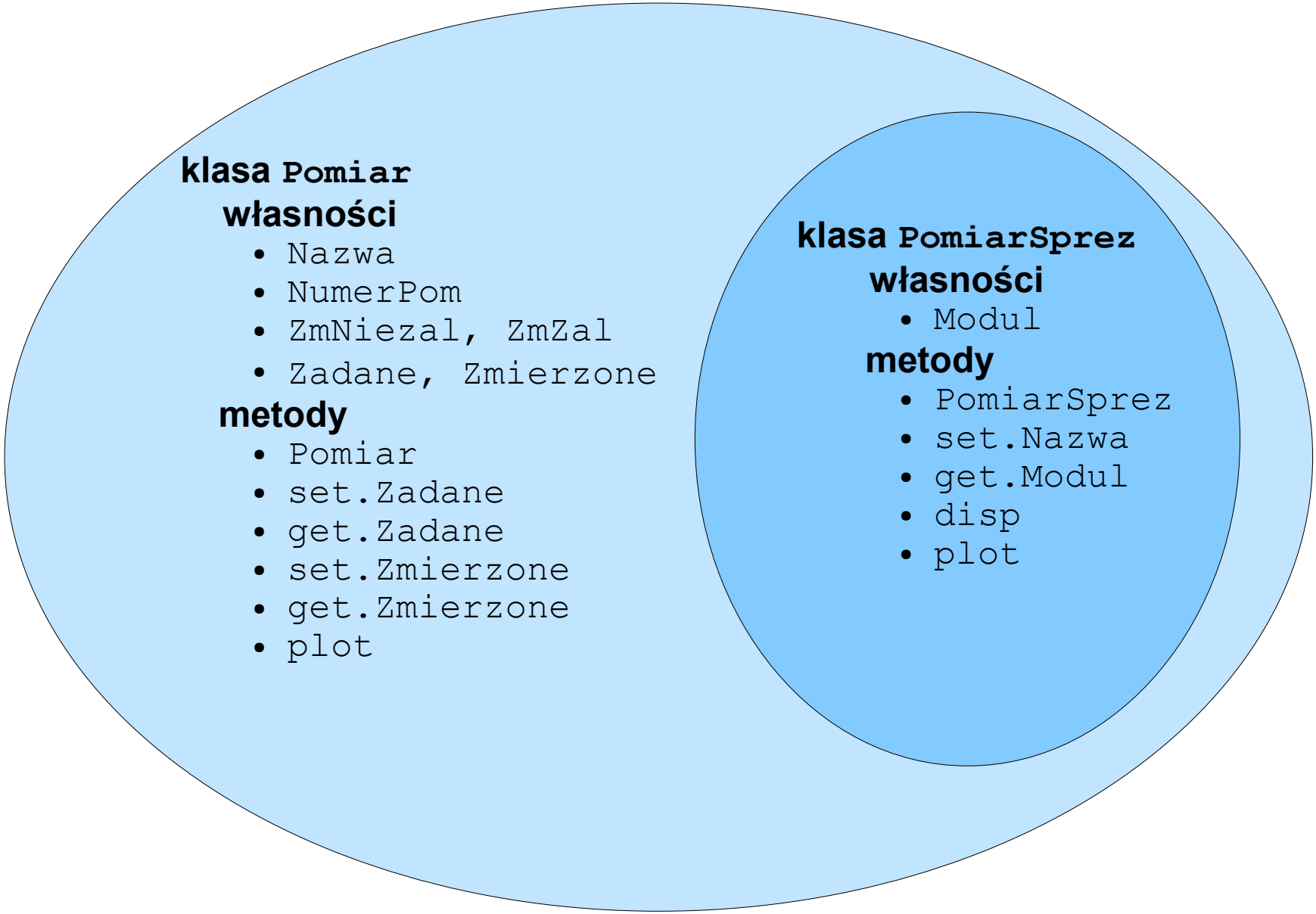
  - w Matlabie zapisujemy to tak:

```
PomiarSprez < Pomiar
```

Klasa `PomiarSprez` *jest pochodna (ang.* subclass*)*
wobec klasy bazowej (ang. superclass*)* `Pomiar`

http://www.mathworks.com/help/techdoc/matlab_oop/brglmxp.html

klasa `PomiarSprez` – projekt



Implementacja klasy pochodnej `PomiarSprez`

MATLAB R2011b

File Edit View Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« Moje dokumenty ▸ MATLAB ▸ inf12l ▸ wyk

Name

- Pomiar.m
- PomiarElast.m
- PomiarElastycznosci.m

Pomiar.m (MATLAB Class)

przechowuje pomiary

- Nazwa
- NumerPom
- ZmNiezal
- ZmZal
- Zadane
- Zmierzone
- Zmierzone1
- Zadane1

Command Window

```
>>
```

Workspace

Stack: Select data to plot

Name	Value

Command History

```
p = Pomiar('Test',2,[1 2 3],[1
p = Pomiar('Test',2,[1 2 3],[1
p.Zmierzone = [1 4 9 10];
p.Zadane = [1 4];
disp(p)
plot(p)
clc
p = Pomiar('Test',2,[1 2 3],[1
p = Pomiar('Test',2,'jedn 1 [m]
plot(p)
p.Zmierzone = [1 4]
plot(p)
help Pomiar
clear p
help Pomiar
%-- 3/30/2012 9:59 AM --%
clc
```

Record

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | Editor | PL | 12:12 PM

Implementacja klasy pochodnej

- Klasa pochodna **nie musi** zawierać <u>własności</u> i <u>metod</u>
  - które są zdefiniowane w klasie bazowej

    np. własności: `Nazwa, Zadane, Zmierzone`
    metody: `set.Zadane, get.Zmierzone`
  - chyba, że chce zmodyfikować działanie <u>metod</u>

    np. metoda `plot`

- Konstruktor klasy <u>pochodnej</u> inicjuje własności klasy <u>bazowej</u>
  - wywołując jej konstruktora, np.

```
pe = pe@Pomiar(material, numprob, 'Naprezenie [psi]',...
     naprez, 'Odksztalcenie', odkszt);
```

http://www.mathworks.com/help/techdoc/matlab_oop/brenzg6.html

Implementacja klasy `PomiarSprez` – testowanie

MATLAB R2011b

File Edit Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« Moje dokumenty ▸ MATLAB ▸ inf12l ▸ wyk

Name

- Pomiar.m
- PomiarElast.m
- PomiarElastycznosci.m
- PomiarSprez.asv
- PomiarSprez.m

PomiarElast.m (MATLAB Class)

przechowuje pomiary elastycznosci i liczy modul

- Material
- NumerProbki
- Odksztalcenie
- Naprezenie
- Odksztalcenie1
- Naprezenie1
- Modul
- PomiarElast(material, numprob, naprez, odkszt)

Command Window

```
>>
```

Workspace

Stack: Select data to plot

Name	Value

Command History

```
p = Pomiar('Test',2,[1 2 3],[1
p = Pomiar('Test',2,[1 2 3],[1
p.Zmierzone = [1 4 9 10];
p.Zadane = [1 4];
disp(p)
plot(p)
clc
p = Pomiar('Test',2,[1 2 3],[1
p = Pomiar('Test',2,'jedn 1 [m]
plot(p)
p.Zmierzone = [1 4]
plot(p)
help Pomiar
clear p
help Pomiar
%-- 3/30/2012 9:59 AM --%
clc
```

Record

Start

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | Editor | SWF Producer | 3_1-dziedziczenie.avi... | PL | 12:21 PM

Kolejny problem

- Własność `Nazwa` należy do klasy bazowej `Pomiar`
  - tylko klasa `Pomiar` może kontrolować jej poprawność
    - poprzez metodę `set.Nazwa`
- Nie ma ogólnej zasady poprawności nazwy pomiaru
  - tylko klasa `PomiarSprez` wie, która `Nazwa` jest poprawna
    - ale nie może posiadać metody `set.Nazwa`
- Jakiś pomysł ?

Kolejny problem (2)

- Chcemy by metoda `set.Nazwa` w klasie bazowej `Pomiar`
  - wywołała metodę sprawdzającą nazwę,
    - która jest zdefiniowana w klasie pochodnej, np. tak:

```
function obj = set.Nazwa(obj,nazwa)
    if ~sprawdzNazwe(obj,nazwa)
        error('Niewlasciwa nazwa eksperymentu!');
    end
    obj.Nazwa = nazwa;
end
```

http://www.mathworks.com/help/techdoc/matlab_oop/brenyev-1.html

Kolejny problem (2)

- Chcemy by metoda `set.Nazwa` w klasie bazowej `Pomiar`
  - wywołała metodę sprawdzającą nazwę,
    - która jest zdefiniowana w klasie pochodnej, np. tak:

```
function obj = set.Nazwa(obj,nazwa)
    if ~sprawdzNazwe(obj,nazwa)
        error('Niewlasciwa nazwa eksperymentu!');
    end
    obj.Nazwa = nazwa;
end
```

- Mam dobrą wiadomość:
  - właśnie tak można to zrobić !

http://www.mathworks.com/help/techdoc/matlab_oop/brenyev-1.html

Do dzieła !

Editor

File Edit Text Go Cell Tools Debug Desktop Window Help

C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk\PomiarSprez.m

```
classdef PomiarSprez < Pomiar
    %POMIARSPREZ przechowuje pomiary elastycznosci i liczy modul
    % Konstruktor: pe = PomiarSprez(material,numprob,naprez,odkszt)
    %  material - nazwa materialu
    %  numprob  - numer badanej probki
    %  naprez   - wektor naprezen
    %  odkszt   - wektor odksztalcen

    properties (Dependent = true, SetAccess = private) ... % Dependent + set-private pro

    methods
      function pe = PomiarSprez(material,numprob,naprez,odkszt)          ... % PomiarSpr

%       function obj = set.Nazwa(obj,material)
%          if ~(strcmpi(material,'aluminum') ||...
%             strcmpi(material,'stal nierdzewna') ||...
%             strcmpi(material,'stal niestopowa'))
%             error('Akceptowany material to aluminium, stal nierdzewna, stal niestopow
%          end
%       obj.Nazwa = material;
%       end % set.Material

      function modul = get.Modul(obj)
         ind = find(obj.Zmierzone > 0); % Tylko niezerowe wartosci odksztalcenie
         modul = mean(obj.Zadane(ind)./obj.Zmierzone(ind));
       end % get.Modul

      function disp(obj) ... % disp

      function plot(obj,varargin) ... % plot

    end % methods

end
```

C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk\Pomiar.m

```
classdef Pomiar
    %POMIAR przechowuje pomiary
    % Konstruktor: p = Pomiar(nazwa,numer,zm_niezal,zadane,zm_zal,zm
    %  nazwa     -nazwa badania (np. nazwa materialu)
    %  numer     -numer badania (np. numer badanej probki)
    %  zm_niezal -nazwa i jednostki zmiennej niezaleznej (zadanej, n
    %  zadane    -wektor wartosci niezaleznych (zadanych)
    %  zm_zal    -nazwa i jednostki zmiennej niezaleznej (zmierzonej
    %  zmierzone -wektor wartosci zaleznych (zmierzonych)

    properties
      Nazwa
      NumerPom
    end % properties

    properties (SetAccess = immutable) ... % immutable properties

    properties (Dependent = true) ... % dependent properties

    properties (Access = private) ... % private properties

    methods
      function p = Pomiar(nazwa,numpom,zm_niezal,zadane,zm_zal,zmier

      function obj = set.Zmierzone(obj, zmierz) ...

      function zmierz = get.Zmierzone(obj) ...

      function obj = set.Zadane(obj, zadane) ...

      function zadane = get.Zadane(obj) ...

      function plot(obj,varargin) ... % plot

    end % methods

end
```

PomiarSprez.m | Pomiar.m

Pomiar Ln 12 Col 11 OVR

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | Editor | PL | 1:19 PM

Poprawność nazwy, czyli materiału – test

MATLAB R2011b

File Edit Debug Desktop Window Help

Current Folder: C:\Documents and Settings\Admin\Moje dokumenty\MATLAB\inf12l\wyk

Shortcuts How to Add What's New

Current Folder

« Moje dokumenty ▸ MATLAB ▸ inf12l ▸ wyk

Name

- Pomiar.m
- PomiarElast.m
- PomiarElastycznosci.m
- PomiarSprez.asv
- PomiarSprez.m

Details

Select a file to view details

Variable Editor - pe2

Stack: Base | No valid plots for: pe2.Modul

pe2 <1x1 PomiarSprez>

Property	Value	Min	Max
Modul	1.8333e+05	1.833...	1.833...
Nazwa	'drewno'		
NumerPom	1	1	1
ZmNiezal	'Naprezenie [psi]'		
ZmZal	'Odksztalcenie'		
Zadane	[20000,40000]	20000	40000
Zmierzone	[0.1200,0.2000]	0.1200	0.2000

Command Window

```
>> pe2 = PomiarSprez('stal nierdzewna',1,[2e4 4e4],[0.12 0.20]);
>> pe2
pe2 =
Material: stal nierdzewna
NumerProbki: 1
Modul:1.8333e+005
>> pe2.Odksztalcenie = [0.12 0.20 0.31];
No public field Odksztalcenie exists for class PomiarSprez.

>> pe2.Zmierzone = [0.12 0.20 0.31];
Error using Pomiar/set.Zmierzone (line 55)
Liczba wartosci zmierzonych nie moze byc mniejsza niz liczba wartosci zadanych

>> pe2.Nazwa = 'drewno'
pe2 =
Material: drewno
NumerProbki: 1
Modul:1.8333e+005
>>
```

Workspace

No valid plots for: pe2

Name	Value
pe2	<1x1 PomiarSprez>

Command History

```
plot(p)
clc
p = Pomiar('Test',2,[1 2 3],[1
p = Pomiar('Test',2,'jedn 1 [m]
plot(p)
p.Zmierzone = [1 4]
plot(p)
help Pomiar
clear p
help Pomiar
%-- 3/30/2012 9:59 AM --%
clc
pe2 = PomiarSprez('stal nierdze
pe2
pe2.Odksztalcenie = [0.12 0.20
pe2.Zmierzone = [0.12 0.20 0.31
pe2.Nazwa = 'drewno'
```

Record

Start | Serwer pocztowy - M... | inf12l | MATLAB R2011b | Editor | PL | 1:43 PM

Polimorfizm

- Właśnie poznaliśmy technikę OOP nazywaną **polimorfizmem**
  - napisaliśmy metodę, która przyjmuje różne formy
    - w zależności od klasy obiektu

```
sprawdzNazwe(obj,nazwa)
```

Jeśli `obj` jest **klasy pochodnej, która ma** metodę `sprawdzNazwe` (np. `PomiarSprez`):

```
function wynik = sprawdzNazwe(~,material)
    if ~(strcmpi(material,'aluminum') || strcmpi(material,'stal'))
        error('Akceptowany material to …')
    end
    wynik = true;
end
```

Jeśli `obj` jest **klasy `Pomiar`** lub **klasy pochodnej, która nie ma** metody `sprawdzNazwe`:

```
function wynik = sprawdzNazwe(~,~)
    wynik = true;
end
```

Polimorfizm (2)

- **Metody (funkcje) wirtualne**
  - metody, które można wywoływać polimorficznie

- **Po co polimorfizm?**
  - rozszerzalność kodu
    - programista klasy `Pomiar` nie musi nic wiedzieć o regułach poprawności w klasach pochodnych

- Programowanie zorientowane obiektowo (OOP)

  OOP = enkapsulacja + dziedziczenie + polimorfizm

OOP – znajdź sam(a)

- wywoływanie metody klasy bazowej
  - wewnątrz metody klasy pochodnej

- metody czysto wirtualne (abstrakcyjne)
- klasy abstrakcyjne i interfejsy

- dziedziczenie wielokrotne

Klasa zwykła i „referencyjna” w Matlabie

- **klasa zwykła** (klasa wartości, ang. *value class*)
  - przekazuje obiekty przez wartość (przez kopie)
    - np. klasy numeryczne Matlaba

```
>> a = int32(7);   % konstruktor tworzy obiekt i zwraca go do a
>> b = a;
>> a = a^4;
>> b
ans = 7          % b przechowało kopię a sprzed potęgowania
```

    - domyślnie klasy użytkownika

http://www.mathworks.com/help/techdoc/matlab_oop/brfylwk-1.html

Klasa zwykła i „referencyjna” w Matlabie (2)

- **klasa referencyjna** (klasa uchwytowa, ang. *handle class*)
  - przekazuje obiekty przez referencje (uchwyty)
    - np. klasy graficzne Matlaba

```
>> x = 1:.1:10; y = sin(x);
>> h1 = line(x,y);                % line zwraca uchwyt h1
>> h2 = h1;                       % h2 odnosi się do
                                  % tej samej linii co h1

>> set(h1,'Color','green')        % linia jest zielona
>> set(h2,'Color','red')          % linia jest czerwona

>> delete(h2)                     % linia została usunięta...
>> set(h1,'Color','green')        %
Invalid or deleted object.        % ...więc dostaliśmy błąd
```

    - klasy użytkownika dziedziczące po klasie `handle`

http://www.mathworks.com/help/techdoc/matlab_oop/brfylwk-1.html

Przykład klasy uchwytowej

- Problem: pacjent w szpitalu
  - sytuacja: przeniesienie z Intensywnej Terapii na Kardiologię
    - przenosimy pacjenta, a nie jego kopię

```
classdef pacjent < handle
   properties
      PESEL = ''
      Oddzial = '';
   end

   methods
      function p = pacjent(pesel,oddz)
         p.PESEL = pesel;
         p.Oddzial = oddz;
      end % pacjent

      function przeniesienie(obj,nowy_oddz)
         obj.Oddzial = nowy_oddz;
      end % przeniesienie
   end

end
```

http://www.mathworks.com/help/techdoc/matlab_oop/brfylwk-1.html

Przykład klasy uchwytowej (2)

- Problem: pacjent w szpitalu
  - sytuacja: przeniesienie z Intensywnej Terapii na Kardiologię
    - <u>przenosimy pacjenta, a nie jego kopię</u>

```
>> p = pacjent('54093001218','Intensywna Terapia');
                                % konstruktor zwraca uchwyt
                                % pacjent przywieziony z zawałem serca

>> p2 = p;                      % kopiujemy uchwyt, ale nie pacjenta
                                % przekazujemy uchwyt p2 rodzinie
                                % aby mogła odnaleźć chorego

>> przeniesienie(p,'Kardiologia');    % metoda przeniesienie zmienia p
>> p2.Oddzial                   % rodzina sprawdza, gdzie leży chory
ans = Kardiologia
```

Przykład klasy uchwytowej (2)

- Problem: pacjent w szpitalu
  - sytuacja: przeniesienie z Intensywnej Terapii na Kardiologię
    - <u>przenosimy pacjenta, a nie jego kopię</u>

```
>> p = pacjent('54093001218','Intensywna Terapia');
                                % konstruktor zwraca uchwyt

>> p2 = p;                      % kopiuje uchwyt, ale nie pacjenta

>> przeniesienie(p,'Kardiologia');    % metoda przeniesienie zmienia p
>> p2.Oddzial
ans = Kardiologia
```

- Gdyby pacjent był klasą zwykłą (*value class*)

```
>> przeniesienie(p,'Kardiologia');
>> p.Oddzial
ans = Intensywna Terapia

>> p = przeniesienie(p,'Kardiologia');        % metoda przeniesienie
>> p.Oddzial                                  % tworzy zmienioną kopię p
ans = Kardiologia
```

Kiedy klasa referencyjna?

- Przykład
  - student rozwiązuje zadanie na kolokwium
    - może, choć to **nieetyczne** zajrzeć do koleżanki/kolegi i przepisać rozwiązanie
      - wniosek: rozwiązanie zadania jest przekazywane przez kopię
  - studentowi odmówił posłuszeństwa długopis
    - może pożyczyć od koleżanki/kolegi
      - wniosek: długopis jest przekazywany przez referencję

Kiedy klasa referencyjna? (2)

- Gdy dwa obiekty nie mogą znajdować się takim samym stanie
  - pacjent nie może leżeć na dwóch oddziałach szpitalnych jednocześnie
  - nie mogą istnieć dwa węzły listy lub drzewa (struktury danych) posiadające identyczne połączenia z innymi węzłami
- Gdy obiekt reprezentuje urządzenie lub zasób
  - nie można skopiować drukarki
  - nie ma sensu tworzyć nowej kopii pliku tekstowego po każdej operacji
- Gdy chcemy kontrolować liczbę obiektów danej klasy
- Gdy obiekt zgłasza lub obsługuje zdarzenia
- Gdy chcemy skorzystać z metod klasy `handle`

http://www.mathworks.com/help/techdoc/matlab_oop/brfylyl-1.html

Klasa referencyjna zapisująca plik

```
classdef ZapisPliku < handle

   properties (SetAccess = private, GetAccess = private)
      IdPliku
   end % properties

   methods
      function obj = ZapisPliku(nazwa_pliku)
      % Konstruuje obiekt i zapisuje identyfikator pliku

         obj.IdPliku = fopen(nazwa_pliku,'a');
      end

      function zapisDoPliku(obj,tekst)
         fprintf(obj.IdPliku,'%s\n',tekst);
      end

      function delete(obj)
      % Metoda delete (destruktor) jest zawsze uruchamiana
      % gdy usuwana jest ostatnia instancja klasy

         fclose(obj.IdPliku);
      end

   end  % methods

end % class
```

http://www.mathworks.com/help/techdoc/matlab_oop/brid518-1.html

Klasa `ZapisPliku` – użycie

```
>> zp = ZapisPliku('moj_plik.txt');   % konstruktor - otwieramy plik

>> zp2 = zp;

>> zp.zapisDoPliku('Lubie programowanie obiektowe.');
>> zp2.zapisDoPliku('To takie proste!');

>> clear zp;                          % destruktor - zamykamy plik

>> type moj_plik.txt
Lubie programowanie obiektowe.
To takie proste!
```

- Ok, wiemy, że
  - klasa obsługująca zasób (np. plik) powinna być referencyjna
- Ale po co nam taka klasa?
  - użytkownik nie musi pamiętać o otwieraniu i zamykaniu pliku

http://www.mathworks.com/help/techdoc/matlab_oop/brid518-1.html

Dziś najważniejsze było...

- Programowanie zorientowane obiektowo =
  - enkapsulacja +
  - dziedziczenie +
  - polimorfizm

- Poznaliśmy zaledwie podstawy OOP w Matlabie
  - możemy śmiało próbować tworzyć własne klasy
  - ale do ekspertów nam daleko
    - tym bardziej, że OOP w Matlabie bywa pogmatowane

A za 2 tygodnie...

- Graficzny interfejs użytkownika
- Tymczasem
  - dobrych świąt Wielkiej Nocy :-)