Dr. Frank Schöpfer

Einführung in Matlab
Übung 12

Aufgabe 1: Erzeugen Sie eine **Polynom-Klasse** zum Rechnen mit Polynomen,

$$p(x) = \sum_{k=0}^{n} a_k \cdot x^k = a_n \cdot x^n + \ldots + a_2 \cdot x^2 + a_1 \cdot x + a_0$$

zu deren Kenntnis ja der **Koeffizienten-Vektor** genügt,

$$p \longleftrightarrow [a_n, \ldots, a_0]$$

z.B.

$$p_1(x) = 2x + 1 \longleftrightarrow [2, 1] \quad , \quad p_2(x) = -x^3 + 2x \longleftrightarrow [-1, 0, 2, 0]$$

Implementieren Sie dies wie folgt als **Klasse** `polynom` mit der **Eigenschaft** `koeff` und **Methoden** für: **Addition, Subtraktion, Multiplikation, Vorzeichen-Plus/-Minus, Potenzieren, Vergleich, Ableitung, Stammfunktion, schöne Ausgabe, Auswertung an einer Stelle/einem Vektor**.

```
classdef polynom
 properties (SetAccess=private)
         koeff
 end
 methods
  % Konstruktor
  function p=polynom(a)
  n=find(a,1);
  if isempty(n)
    p.koeff=0;
  else
    p.koeff=a(n:end);
  end
  end
  % weitere Methoden
  function p=plus(p1,p2)...
  function p=minus(p1,p2)...
  function p=mtimes(p1,p2)...
  function p=uplus(p)...
  function p=uminus(p)...
  function pn=mpower(p,n)...
  function e=eq(p1,p2)
  function dp=ableitung(p)...
  function P=stammfunktion(p)...
  function disp(p)...
  function pt=auswerten(p,t)...
end
end
```

Hinweise:

- Der **Konstruktor** ist hier so gemacht, dass **unnötige Nullen am Anfang des Koeffizientenvektors gelöscht** werden (zum Auffinden des ersten Nicht-Null-Eintrages dient `n=find(a,1)`), z.B.

```
>> p=polynom([0,0,1,1,1])
p =
 + 1*x^2 + 1*x + 1
```

- **Summe zweier Polynome** entspricht **Addition der Koeffizienten-Vektoren**, wobei **gegebenenfalls Nullen vorne angehängt** werden müssen, um gleiche Länge zu erhalten,

$$p_1(x) + p_2(x) = \sum_{k=0}^{n} (a_k^1 + a_k^2) \cdot x^k$$

  mit $n = \max\{\mathrm{grad}(p_1), \mathrm{grad}(p_2)\}$.

- **Produkt zweier Polynome** entspricht **Faltung der Koeffizienten-Vektoren**,

$$p_1(x) \cdot p_2(x) = \sum_{k=0}^{n} a_k \cdot x^k \quad , \quad a_k = \sum_{j=0}^{k} a_j^1 \cdot a_{k-j}^2$$

  mit $n = \mathrm{grad}(p_1) + \mathrm{grad}(p_2)$. Für die **Faltung der Koeffizienten-Vektoren kann man den Matlab-Befehl `conv` verwenden**.

- **Einmal ableiten** ergibt

$$p'(x) = \sum_{k=1}^{n} a_k \cdot k \cdot x^{k-1}$$

- Eine **Stammfunktion** $P(x)$ von $p(x)$ ist

$$P(x) = \sum_{k=0}^{n} \frac{a_k}{k+1} \cdot x^{k+1}$$

- Für die **Auswertung an einer Stelle/einem Vektor kann man den Matlab-Befehl `polyval` verwenden** (Dieser basiert auf dem Horner-Schema).

- Ein **Test** könnte so aussehen

```
>> x=polynom([1 0])
x =
 + 1*x
>> ableitung(-x^4+5-2*(x+3)^2)
ans =
 - 4*x^3 - 4*x - 12
>> stammfunktion(2+4*(x+1)^3)
ans =
 + 1*x^4 + 4*x^3 + 6*x^2 + 6*x
>> (x-1)^2 == +x^2-2*x+1
ans =
  logical
   1
>> x^2==4
ans =
  logical
   0
>> x^2+1==x
ans =
  logical
   0
>> t=linspace(-2,2,1000); y=auswerten(x*(x^2-1),t); plot(t,y)
```