

Теория.

Бином (лат. *bis* — дважды, *nomēn* — имя) или **двучлен** — частный случай **полинома** (многочлена), состоящего из двух слагаемых **мономов** (одночленов). Например:

$$a + b, a - b, a^2 + b^2, 3b - 4b^3.$$

Для вычисления степеней биномов используется **бином Ньютона**.

Например, в случае второй степени:

$$(a + b)^2 = a^2 + 2ab + b^2.$$

Также используются формулы

$$(ax + b)(cx + d) = acx^2 + (ad + bc)x + bd,$$

$$(a + b)(a - b) = a^2 - b^2.$$

Источник: <https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BD%D0%BE%D0%BC>

Реализация.

Биномы — частный случай полиномов. Поэтому я сначала реализовал более общих класс для работы с полиномами `polynom`, а затем с его использованием сделал класс `binom`:

D:\CloudMail\Projects\Binom\polynom.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Insert Comment % Indent Breakpoints Run Run and Advance Run and Time

```
1 classdef polynom
2     %Класс для работы с многочленами
3     %Класс позволяет работать с многочленами. Реализованы операции сложения,
4     %вычитания, умножения, деления.
5     properties (Access = private) % блок описания собственных полей, не доступных вне класса
6         coefs %коэффициенты многочлена
7     end
8     methods (Access = public) % общедоступные методы
9         function this = polynom(coefficients)
10             % конструктор. принимает массив коэффициентов
11             if nargin == 0 %проверяем, переданы ли аргументы
12                 this.coefs = [ ]; %если не переданы
13             elseif isa(coefficients, 'polynom') %проверяем, передан вектор или объект полинома
14                 this = coefficients; %если объект
15             else
16                 this.coefs = coefficients(:).'; %если вектор
17             end
18         end
19         function result = getCoefficients(n)
```

polynom / mrddivide Ln 84 Col 31

D:\CloudMail\Projects\Binom\binom.m

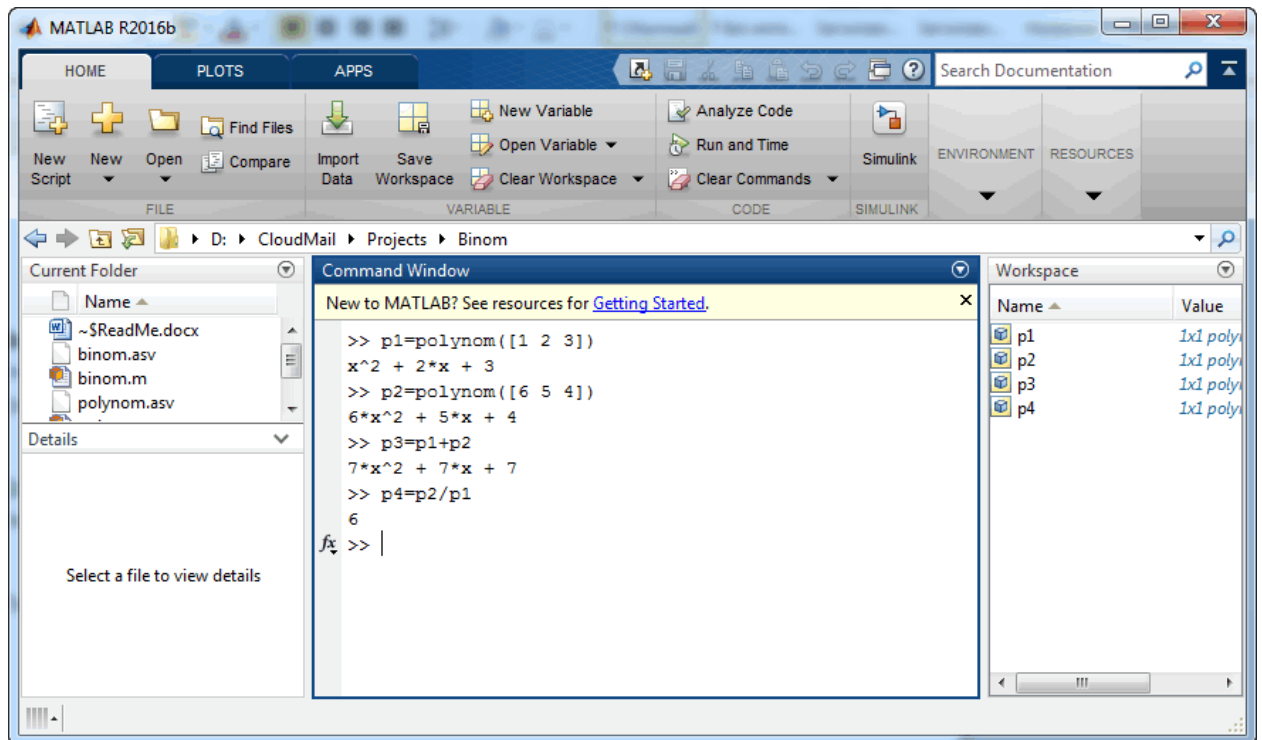
EDITOR PUBLISH VIEW

New Open Save Find Files Compare Go To Find Insert Comment % Indent Breakpoints Run Run and Advance Run and Time

```
1 classdef binom
2     %Класс для работы с биномами
3     %Класс позволяет работать с биномами. Реализованы операции сложения,
4     %вычитания, умножения, деления.
5     properties (Access = private) % блок описания собственных полей, не доступных вне класса
6         pol %объект многочлена
7     end
8     methods (Access = public) % общедоступные методы
9         function this = binom(coefficients, pows)
10             % конструктор. принимает массив коэффициентов и массив степеней
11             if nargin == 0
12                 this.pol = polynom();
13             elseif nargin == 1
14                 if isa(coefficients, 'polynom')
15                     this.pol = coefficients;
16                 elseif isa(coefficients, 'binom')
17                     this = coefficients;
18                 else
```

binom Ln 5 Col 16

Пример использования класса `polynom`:



Над полиномами реализованы все необходимые операции: сложения, вычитания, деления, умножения, возведения в степень. Реализована функция `display()`, в конструкторе производится проверка аргументов. Если аргументы не переданы – создаётся пустой объект, если в качестве аргумента передан полином – то создаётся его копия, если в качестве аргумента передан вектор, то он используется как вектор коэффициентов полинома.

Всё это справедливо и для класса `binom` – также реализованы все необходимые операции: сложение, вычитание, умножение, деление, возведение в степень, клонирование.

Однако при выполнении операций над биномами нужно понимать, что результат операции далеко не всегда будет являться биномом. В этом случае будет показана ошибка: «Результат операции не является биномом».

Конструктор класса `бином` принимает два вектора: вектор коэффициентов и вектор степеней, при которых эти коэффициенты стоят. Вектора должны быть длины 2, иначе это будет не БИном. Проверка длины векторов тоже производится. При неподходящей длине будет создан пустой объект. Также в конструктор класса `бином` можно передать другой объект-бином, либо объект-полином.

В коде есть небольшие комментарии, да и названия функций говорят сами за себя – легко понять, какая функция что делает.

Использование класса `бином`:

