

## Task

To create type Polynomial – polynomial in one variable

**Polynomial** shows polynomial in one variable that can be represented in general formula

$c_0 + c_1 * x^1 + c_2 * x^2 + \dots + c_m * x^m$ , where

$c_1, c_2, \dots, c_m$  – real type coefficient,

$x$  – polynomial variable,

$c_0, c_1 * x^1, c_2 * x^2, c_m * x^m$  – monomial of polynomial,

coefficients and degrees – real numbers (double).

Polynomial degree is maximum degree of its monomials

Monomial can be added, deleted in polynomial, also one can determine number of included monomials with nonzero coefficient and polynomial degree. Corresponding methods as well as indexer can be used for adding and deleting monomials.

Coefficient value of predetermined monomial can be changed by indexer in the following way:

- if new coefficient value does not equal 0, and monomial with predetermined degree exists – monomial degree changes;
- if new coefficient value does not equal 0, and monomial with predetermined degree does not exist – new monomial appears in polynomial;
- if new coefficient value equals 0, and monomial with predetermined degree does not exist – monomial with predetermined degree is deleted, as its coefficient goes to 0.

Polynomial can be added, deducted, and multiplied by methods as well as by overloaded operators “+”, “-”, “\*”.

Polynomial monomials can be obtained as array of monomials.

Task has two levels of **Low** and **Advanced** complexity

**Low** level requires implementing the following functionalities:

- Constructors for creating polynomials based on one or several monomials (excluding Tuple parameters)
- Obtaining number of nonzero monomials
- Obtaining polynomial degree
- Adding predefined monomial (excluding parameter Tuple)
- Deleting monomial with predefined degree
- Checking if there is a monomial with predefined degree
- Obtaining monomial by degree.
- Indexer
- Obtaining monomials in a form of array
- Addition, deduction, and multiplication of polynomial by methods and overloaded operations (excluding parameter Tuple)

**Advanced** level requires implementing the following functionalities:

- All completed **Low** level tasks
- Generating of exceptions, defined in xml-comments to class methods
- Constructors and class methods with parameter Tuple