

## Лабораторная работа №1

Создание приложения для работы с интерполяционными полиномами.

1. Разработать класс `Polynomial` для работы с произвольными полиномами. Класс должен содержать следующие основные элементы.
  - а. Поле, хранящее коэффициенты полинома в виде списка (`List` или `ArrayList`), либо в виде отображения (`Map`).
  - б. Несколько вариантов публичных конструкторов:
    - по умолчанию, для создания полинома нулевой степени равного 0;
    - с параметром-списком коэффициентов (`List`), либо отображением (`Map`);
    - с переменным количеством вещественнозначных параметров;
    - с параметром-массивом коэффициентов.
  - в. Публичный метод для получения копии списка (либо отображения) коэффициентов.
  - г. Переопределенный метод `toString()` для представления полинома в формате строки вида:  $a_1x^n + a_2x^{n-1} - a_3x^{n-2} + \dots - a_nx + a_{n+1}$ . Необходимо, чтобы запись полинома максимально соответствовала привычному способу записи полинома (за исключением лишь символа «^», необходимого для обозначения степени, поскольку в текстовом режиме невозможно будет указать верхний индекс).
  - д. Переопределенные методы `equals()` и `hashCode()`.
  - е. Приватный метод корректировки списка коэффициентов, для удаления ненужных значений.
  - ж. Публичный метод для получения значения степени полинома.
  - з. Публичные методы `plus`, `minus`, `times` для выполнения соответствующий арифметических операций с полиномами.
  - и. Публичные методы `times`, `div` для выполнения умножения и деления полинома на число.
  - к. Публичный метод `invoke()` или `calc()` для вычисления значения полинома в указанной точке.
2. Разработать класс `InterpolatingPolynomial` – наследник класса `Polynomial` для создания интерполяционного полинома Ньютона по набору точек на плоскости. Класс должен содержать следующие элементы.
  - а. Приватную изменяемую коллекцию точек, через которые должен проходить график интерполяционного полинома.
  - б. Публичный Метод для получения копии списка точек.

## в. Конструкторы:

- по умолчанию;
- с параметром-коллекцией точек, используемых для построения интерполяционного полинома.

г. Приватный метод для вычисления разделенных разностей, используемых в формуле построения полинома.

д. Публичный метод добавления новой точки в коллекцию точек, одновременно перестраивающий сам интерполяционный полином.

е. Публичный метод удаления точки из коллекции. Проверяющий наличие соответствующей точки в коллекции и удаляющий ее при наличии, а также перестраивающий сам полином.

3. Разработать тесты на JUnit5 для проверки работоспособности всех методов обоих классов. Тесты должны обеспечивать максимально возможное покрытие кода.

4. С помощью тестовых функций выполнить замеры времени, требующегося на построение множества интерполяционных полиномов. Сравнить время построения полинома Лагранжа (класс, который был построен на лекциях или практических занятиях) и Ньютона, а также время, требуемое для добавления точки.

(Примечание: степень и количество строимых полиномов должны быть такими, чтобы время построения полинома Лагранжа измерялось несколькими секундами).