

Задача (Task) 2

1. Вектор

Разработать класс **Vector**.

Класс должен хранить вектор (массив) целых чисел заданного размера, размещенный в динамической памяти.

Класс должен содержать все необходимые конструкторы, деструктор.

Класс должен предоставлять следующие операции и методы:

- 1) операции присваивания, суммирования и скалярного произведения векторов
- 2) операцию индексации с контролем выхода индекса за границы массива.
- 3) операцию умножения скаляра на вектор
- 4) операции << и >> для сохранения себя в файле и чтения из файла
- 5) вычислить длину вектора.

2. Полином

Разработать класс **Polinom**. Класс должен хранить полином (многочлен) от одной переменной (x), представленный в виде степени полинома и массива действительных коэффициентов полинома, размещенного в динамической памяти.

Класс должен содержать все необходимые конструкторы, деструктор.

Класс должен предоставлять следующие операции и методы:

- 1) операции присваивания и суммирования
- 2) операцию индексации с контролем выхода индекса за границы массива.
- 3) метод вывода себя на консоль в текстовой форме: $1.5+4.8*x+2.4*x^2+ \dots$
- 4) операции << и >> для сохранения себя в файле и чтения из файла
- 5) вычислить значение полинома в заданной точке x ,
- 6) найти полином, являющийся производной исходного полинома.

3. Большое число

Разработать класс **Decimal** для работы с беззнаковыми целыми десятичными числами, используя для представления числа массив из элементов типа `unsigned char`, каждый из которых является десятичной цифрой. Младшая цифра имеет меньший индекс (единицы — в нулевом элементе массива). Размер массива задается при конструировании объектов.

Класс должен содержать все необходимые конструкторы и деструктор.

В классе должны быть перегружены операции:

- присваивания;
- сложения и вычитания;
- сравнения;
- ввода/вывода в поток.

4. Восьмеричное число

Создать класс **Octal** для работы с беззнаковыми целыми восьмеричными числами, используя для представления числа массив элементов типа `unsigned char`, каждый элемент которого является восьмеричной цифрой. Младшая цифра имеет меньший индекс (единицы — в нулевом элементе массива). Размер массива задается как аргумент конструктора.

Реализовать операции:

присваивания;

арифметические: +, -

сравнения

можно попробовать умножение

операцию индексации с контролем выхода индекса за границы массива.
операции << и >> для сохранения себя в файле и чтения из файла

5. Шестнадцатеричное число

Создать класс **Hex** для работы с беззнаковыми целыми шестнадцатеричными числами, используя для представления числа массив из элементов типа `unsigned char`, каждый из которых является шестнадцатеричной цифрой. Младшая цифра имеет меньший индекс. Размер массива задается как аргумент конструктора.

Реализовать операции:

присваивания;

арифметические: +, -;

сравнения

можно попробовать умножение

операцию индексации с контролем выхода индекса за границы массива.

операции << и >> для сохранения себя в файле и чтения из файла

6. Матрица

Разработать класс **Matrix**.

Класс должен хранить квадратную матрицу целых чисел заданного размера.

Класс должен содержать все необходимые конструкторы, деструктор

Класс должен предоставлять следующие операции и методы:

- 1) операцию присваивания
- 2) операцию суммирования двух матриц одинакового размера
- 3) операцию индексации с контролем выхода индекса за границы массива.
- 4) операции << и >> для сохранения себя в файле и чтения из файла
- 5) операцию транспонирования матрицы
- 6) проверить, обладает ли матрица диагональным преобладанием,
- 7) операцию умножения матриц и операцию умножения скаляра на матрицу