НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

Институт ЛаПлаз

Кафедра №31 «Прикладная математика»

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

«Объектно-ориентированное программирование»

Общие требования

Для всех вариантов заданий обязательны следующие требования:

1. Класс должен управлять динамической памятью и корректно реализовывать правило пяти:
   - Деструктор;
   - Конструктор копирования;
   - Конструктор перемещения;
   - Оператор присваивания копированием;
   - Оператор присваивания перемещением.
2. Обязательная перегрузка операторов:
   - Арифметические операторы (`+`, `-`, `*`);
   - Операторы сравнения (`==`, `!=`);
   - Операторы ввода-вывода (`<<`, `>>`);
   - Префиксный и постфиксный инкремент (`++`);
   - Префиксный и постфиксный декремент (`--`).
3. Запрещено использовать:
   - Контейнеры стандартной библиотеки (`std::vector`, `std::list` и т.д.);
   - Умные указатели (`std::unique_ptr`, `std::shared_ptr`);
   - Алгоритмы стандартной библиотеки (`std::sort`, `std::find` и т.д.).
4. Вся работа с динамической памятью только через `new`/`delete` или `new[]`/`delete[]`.

Выбор варианта

Номер задания определяется по формуле: $n = k\%3+1$, где k — номер студента в списке группы.

№ Задания	Название
1	Класс Matrix
2	Класс Polynomial
3	Класс String

Задание 1. Класс Matrix

Реализовать класс `Matrix` для работы с матрицами вещественных чисел произвольного размера. Класс должен хранить элементы в одномерном динамическом массиве, количество строк и столбцов.

Семантика обязательных операторов:

- `+` и `-` — поэлементное сложение и вычитание матриц одинакового размера;
- `*` — умножение матриц по правилам линейной алгебры или умножение матрицы на скаляр;

- `==` и `!=` — поэлементное сравнение с учётом погрешности;
- `++m` (префикс) — добавление в начало пустой строки и столбца (расширение матрицы);
- `m++` (постфикс) — добавление в конец пустой строки и столбца (расширение матрицы);
- `--m` (префикс) — удаление первой строки и столбца;
- `m--` (постфикс) — удаление последней строки и столбца;
- `<<` — вывод матрицы в табличном виде;
- `>>` — ввод размеров, затем элементов построчно.

Дополнительные операторы и методы:

- `operator[](i)` — доступ к i-й строке матрицы (подумайте о возвращаемом значении, чтобы была возможность дальнейшей индексации `matrix[i][j]`);
- `operator()(i, j)` или `operator[](i, j)` — доступ к элементу по индексам строки и столбца;
- `operator^(n)` — возведение квадратной матрицы в целую степень `n`;
- `operator+=`, `operator-=`, `operator*=` — составные операторы присваивания;
- `Rows()` — возвращает количество строк матрицы;
- `Cols()` — возвращает количество столбцов матрицы;
- `Transpose()` — возвращает транспонированную матрицу.

Задание 2. Класс Polynomial

Реализовать класс `Polynomial` для работы с полиномами произвольной степени с вещественными коэффициентами. Класс должен хранить коэффициенты в динамическом массиве и степень полинома.

Семантика обязательных операторов:

- `+` и `-` — сложение и вычитание полиномов;
- `*` — умножение полиномов или умножение полинома на скаляр;
- `==` и `!=` — сравнение полиномов;
- `++p` (префикс) — взятие первообразной с нулевой константой;
- `p++` (постфикс) — увеличение степени полинома на 1 (добавление нулевого старшего коэффициента);
- `--p` (префикс) — взятие производной полинома, возврат копии исходного полинома;
- `p--` (постфикс) — уменьшение степени полинома на 1 (удаление старшего коэффициента);
- `<<` — вывод в виде $a_0 + a_1x + a_2x^2 + ...$;

- `>>` — ввод степени, затем коэффициентов.

Дополнительные операторы и методы:

- `operator[](i)` — доступ к коэффициенту при x^i с возможностью изменения и без;
- `operator()(x)` — вычисление значения полинома в точке x;
- `operator^(n)` — возведение полинома в целую неотрицательную степень `n`;
- `operator%` — деление полинома на полином с остатком, возвращает остаток;
- `Degree()` — возвращает степень полинома.

Задание 3. Класс String

Реализовать класс `String` для работы с изменяемыми строками. Класс должен хранить символы в динамическом массиве, текущую длину строки.

Семантика обязательных операторов:

- `+` — конкатенация строк;
- `-` — удаление всех вхождений второй строки из первой;
- `*` — повторение строки заданное число раз;
- `==` и `!=` — лексикографическое сравнение;
- `++s` (префикс) — добавление пробела в начало строки;
- `s++` (постфикс) — добавление пробела в конец строки;
- `--s` (префикс) — удаление первого символа;
- `s--` (постфикс) — удаление последнего символа;
- `<<` — вывод строки;
- `>>` — ввод строки (до пробела или конца строки).

Дополнительные операторы и методы:

- `operator[](i)` — доступ к символу по индексу с возможностью изменения и без;
- `operator()(start, end)` — извлечение подстроки с позиции `start` до позиции `end`;
- `operator^` — посимвольное `XOR` двух строк (операция шифрования);
- `operator<`, `operator>`, `operator<=`, `operator>=` — лексикографическое сравнение;
- `operator-` (унарный) — создание копии строки с обратным порядком символов;
- `Length()` — возвращает текущую длину строки.