

3. Определите АД для хранения информации о таблице базы данных: столбцы (количество и названия столбцов), строки со значениями (все значения в строке таблицы имеют тип `string`), ключи для получения строк таблицы в некотором порядке (для упрощения ключ включает только один столбец, т.е. ключ - это имя или номер столбца, а строки таблицы можно получать последовательно по одной в порядке возрастания значения в указанном столбце). Перечислите методы АД, обеспечивающие последовательный доступ к информации и её изменение, аргументы и возвращаемые значения каждого метода с комментариями.

1. **Получить количество столбцов.**
 - Аргументы: нет.
 - Возвращаемое значение: `int` — количество столбцов.
2. **Получить количество строк.**
 - Аргументы: нет.
 - Возвращаемое значение: `int` — количество строк.
3. **Получить названия всех столбцов.**
 - Аргументы: нет.
 - Возвращаемое значение: список названий столбцов в соответствующем порядке.
4. **Добавить столбец.**
 - Аргументы: название столбца.
 - Возвращаемое значение: `void`.
5. **Удалить столбец.**
 - Аргументы: название столбца или его индекс.
 - Возвращаемое значение: `void`.
6. **Добавить строку.**
 - Аргументы: список значений.
 - Возвращаемое значение: `void`.
7. **Удалить строку.**
 - Аргументы: номер строки.
 - Возвращаемое значение: `void`.
8. **Получить всю строку.**
 - Аргументы: номер строки.
 - Возвращаемое значение: список значений в соответствующей строке.
9. **Получить значение у указанной колонки и строки.**
 - Аргументы: название или индекс столбца и номер строки.
 - Возвращаемое значение: значение в соответствующем поле.
10. **Получить итератор на начало.**
 - Аргументы: нет.
 - Возвращаемое значение: итератор на первую строку таблицы.
11. **Получить итератор на конец.**

- **Аргументы:** нет.
 - **Возвращаемое значение:** итератор на элемент, следующий за последней строкой таблицы.
12. **Отсортировать значения по столбцу.**
- **Аргументы:** название или индекс столбца.
 - **Возвращаемое значение:** void.
13. **Изменить значение в ячейке.**
- **Аргументы:** название или индекс столбца, номер строки и новое значение.
 - **Возвращаемое значение:** void.
14. **Изменить значение в строке.**
- **Аргументы:** номер строки и список новых значений.
 - **Возвращаемое значение:** void.

4. Предложите структуры данных для представления АТД из задания 3. Перечислите поля, их типы и комментарии к каждому полю. Укажите оценку эффективности (амортизированную или среднюю) для каждого метода с учетом использованных структур данных. Хранимая в структуре информация не должна дублироваться. Это отдельная задача. Не объединяете с задачей 3! Не пишите реализацию методов, нужно указать только оценку эффективности.

Структуры данных:

- `vector<string> column` — массив названий столбцов.
- `vector<vector<string>> rows` — массив массивов строк.

Оценка эффективности методов:

1. Получить количество столбцов.
 - Эффективность: $O(1)$
2. Получить количество строк.
 - Эффективность: $O(1)$
3. Получить названия всех столбцов.
 - Эффективность: $O(1)$
4. Добавить столбец.
 - Эффективность: $O(n)$ (— добавление названия столбца $O(1)$ и добавление элемента в конец каждой строки $O(n)$).
5. Удалить столбец.
 - Эффективность: $O(n \cdot m)$ — поиск столбца $O(m)$, удаление названия столбца $O(m)$ и удаление элементов в строках $O(n \cdot m)$.
6. Добавить строку.
 - Эффективность: $O(1)$ (амортизированная)
7. Удалить строку.
 - Эффективность: $O(n)$ — удаление строки по индексу.
8. Получить всю строку.
 - Эффективность: $O(1)$
9. Получить значение у указанной колонки и строки.
 - Эффективность: $O(m)$ — если столбец задан строкой, $O(1)$ — если номером.
10. Получить итератор на начало.
 - Эффективность: $O(1)$
11. Получить итератор на конец.
 - Эффективность: $O(1)$
12. Отсортировать значения по столбцу.
 - Эффективность: $O(n \log n)$ — сортировка строк по указанному столбцу.
13. Изменить значение в ячейке.
 - Эффективность: $O(m)$ — если столбец задан строкой, $O(1)$ — если номером.
14. Изменить значение в строке.

- **Эффективность:** $O(m)$

23. Определите необходимые геометрические объекты и напишите следующую функцию

В декартовой системе координат на плоскости заданы координаты вершин треугольника и ещё одной точки. Определить, принадлежит ли эта точка треугольнику.

Для точки использовать класс из лекций и его методы.

```
#include <iostream>
#include <cmath>

using namespace std;

struct Point
{
    double x, y;
    double len() const { return hypot(x, y); }
    Point operator-(Point p) const
    {
        return {x - p.x, y - p.y};
    }
    double operator*(Point p) const
    {
        return x * p.x + y * p.y;
    }
    double operator^(Point p) const
    {
        return x * p.y - y * p.x;
    }
};

int isPointInTriangle(Point p, Point a, Point b,
Point c) {
    // Векторное произведение для каждого ребра
    треугольника
    double d1 = (b - a) ^ (a - p);
    double d2 = (c - b) ^ (b - p);
    double d3 = (a - c) ^ (c - p);
```

```
    if ((d1 <= 0 && d2 <= 0 && d3 <= 0) ||
(d1 >= 0 && d2 >= 0 && d3 >= 0)) {
        return 1;
    }
    else {
        return 0;
    }
}
```