
Задание

Общее:

- 1) Составить на языке C++ описания классов для указанных объектов.
- 2) В среде Visual Studio разработать консольную программу, иллюстрирующую использование объектов заданных классов.
- 3) Программа должна иметь меню с пунктами для ввода исходных данных, вывода результатов, выполнения требуемых операций над объектами.

Индивидуальное:

Таблица на основе одномерного массива.

Добавление / исключение строки / столбца, ввод элемента, копирование таблицы.

Описание работы программы

Для выполнения своего задания я создал класс *таблица*, на основе одномерного динамического массива. В этом классе создал, как полагается конструктор/деструктор и функции члена класса для работы с объектами класса. В основной программе для реализации работы реализовал цикл и вложенный в него оператор выбора `switch...case...` Для того чтобы работа программы выполнялась многократно, в точности, пока пользователь сам не решит завершить ее. При выборе пользователем определенных действий вызываются функции члена класса и преобразуют таблицу по условиям задания.

Алгоритмы выполнения основных операций на псевдокоде

1) Ввод таблицы:

Начало;
Ввод<m,n>;
Цикл<до m*n>;
 Ввод элементов таблицы;
Конец.

2) Ввод/замена элемента в таблице:

Начало;
Ввод индексов элемента, который необходимо изменить<m,n>;

Ввод элемента $\langle s \rangle$;
Запись элемента в таблицу $\langle [m, n] \rightarrow s \rangle$;
Конец.

3) Копирование таблицы

Начало;
Задание размеров для новой таблицы;
Цикл по все элемента копируемой таблицы $\langle m, n \rangle$;
 Ввод элементов в новую таблицу;
Конец.

4) Удаление столбца

Начало;
Ввод номера удаляемого столбца $\langle t \rangle$;
Цикл по всем элементам таблицы $\langle m, n \rangle$;
 Удаление в таблице каждого элемента в столбце t ;
 Сдвиг элементов влево;
Задание новых размеров таблицы;
Сокращение выделенной памяти
Конец.

5) Добавление столбца

Начало;
Выделение памяти под столбец
Цикл по всем элементам таблицы $\langle m, n \rangle$;
 Если конец строки, то добавить ячейку и
 сдвинуть элементы вправо;
 Иначе продолжить;
Задать новые размеры таблицы;
Конец.

6) Удаление строки

Начало;
Ввод номера удаляемой строки $\langle t \rangle$;
Цикл по всем элементам таблицы $\langle m, n \rangle$;
 Удаление элементов, если они принадлежат строке t ;
Сдвиг элементов после этой строки влево;
Задание новых размеров таблицы;
Сокращение выделенной памяти
Конец.

7) Добавление строки

Начало;
Выделение памяти под строку
Добавление в конец таблицы строки с n (число столбцов) элементами;
Изменение размеров таблицы;
Конец.

Тесты

№	Таблица	Добавлен ие столбца	Добавление строки	Удалени е столбца	Удаление строки	Изменен ие элемента	Копирование таблицы
1	3 2 2 d f l s _ d	3 2 2 _ d f l _ s _ d _	3 2 2 _ d f l _ s _ d _ -----	2 2 _ f l _ _ d _ -----	2 2 _ _ d _ -----	2 2 _ _ e _ -----	2 2 _ _ e _ -----
2	2 t k 2 p l	2 t k _ 2 p l _	2 t k _ 2 p l _ -----	2 t _ 2 p _ -----	2 p _ -----	2 p u -----	2 p u -----
3	_ _ F _ _ _	_ _ _ _ F _ _ _ _	_ _ _ _ F _ _ _ _ _ _ _	_ _ F _ _ _ _ _	_ _ _ _ _ _	_ k _ _ _ _	_ k _ _ _ _

Распечатки экранов при работе программы

Ввод таблицы:

```

-----Menu-----
1 - Input table
2 - Input/change element
3 - Delete column
4 - Delete string
5 - Add column
6 - Add string
7 - Copy table
8 - Show table
9 - Finish the programm
Select an action:
1
Input number column: 3

Input number string: 3

a b c
d f g
i k q
Select an action:
8

---TABLE---
  a b c
  d f g
  i k q
---TABLE---
Select an action:

```

Ввод/замена элемента:

```
Select an action:
```

```
2
```

```
Input index: 3 2
```

```
Input volume: p
```

```
Select an action:
```

```
8
```

```
---TABLE---
```

```
  a b c
```

```
  d f g
```

```
  i p q
```

```
---TABLE---
```

```
Select an action:
```

Добавление столбца:

```
Select an action:
```

```
5
```

```
Column added!
```

```
Select an action:
```

```
8
```

```
---TABLE---
```

```
  a b c _
```

```
  d f g _
```

```
  i k q _
```

```
---TABLE---
```

Удаление столбца:

```
Select an action:
```

```
3
```

```
Enter the number column to delete:
```

```
1
```

```
Select an action:
```

```
8
```

```
---TABLE---
```

```
  b c _
```

```
  f g _
```

```
  k q _
```

```
---TABLE---
```

Добавление строки:

```
Select an action:
```

```
6
```

```
String added!
```

```
Select an action:
```

```
8
```

```
---TABLE---
```

```
  b c _
```

```
  f g _
```

```
  k q _
```

```
  _ _ _
```

```
---TABLE---
```

```
Select an action:
```

Удаление строки:

```
Select an action:
4
Enter the number string to delete:
2
Select an action:
8

---TABLE---
  b c _
  k q _
  _ _ _
---TABLE---
Select an action:
```

Копирование таблицы:

```
7
Table is copied
Select an action:
```

Листинг программы

ФАЙЛ MAIN

```
#include <iostream>
#include "Table.h"

using namespace std;
int main() {
    Table Tab;
    Table Buf;
    int a, n;
    cout << "-----Menu-----" << endl;
    cout << "1 - Input table" << endl;
    cout << "2 - Input/change element" << endl;
    cout << "3 - Delete column" << endl;
    cout << "4 - Delete string" << endl;
    cout << "5 - Add column" << endl;
    cout << "6 - Add string" << endl;
    cout << "7 - Copy table" << endl;
    cout << "8 - Show table" << endl;
    cout << "9 - Finish the programm" << endl;
    cout << "Select an action: " << endl;
    cin >> a;
    while (a != 9) {
        switch (a) {
            case 1:
                Tab.Input_Table();
                cout << "Select an action:" << endl;
                cin >> a;
                break;
            case 2:
                Tab.Input_Element();
                cout << "Select an action:" << endl;
```

```

        cin >> a;
        break;
    case 3:
        cout << "Enter the number column to delete: " << endl;
        cin >> n;
        Tab.Delete_column(n);
        cout << "Select an action:" << endl;
        cin >> a;
        break;
    case 4:
        cout << "Enter the number string to delete: " << endl;
        cin >> n;
        Tab.Delete_string(n);
        cout << "Select an action:" << endl;
        cin >> a;
        break;

    case 5:
        Tab.Add_column();
        cout << "Column added!" << endl;
        cout << "Select an action:" << endl;
        cin >> a;
        break;
    case 6:
        Tab.Add_string();
        cout << "String added!" << endl;
        cout << "Select an action:" << endl;
        cin >> a;
        break;
    case 7:
        Tab.Copy(Buf);
        cout << "Table is copied" << endl;
        cout << "Select an action:" << endl;
        cin >> a;
        break;
    case 8:
        Tab.Print();
        cout << "Select an action:" << endl;
        cin >> a;
        break;
    default:
        a = 9;
        break;
    }
}
cout << "Program finished =) ";
return 0;
}

```

ФАЙЛ TABLE.h

```

#pragma once
using namespace std;

class Table {
private:
    int m, n; //Количество строк и столбцов
    char* T; //таблица
public:
    void allocate_memory(int size);
    Table(int a = 0, int b = 0); //конструктор по умолчанию
    ~Table(); //деструктор
    Table(const Table& tb); //конструктор копирования
    void Delete_column(int a);
    void Delete_string(int a);
    void Add_column();
    void Add_string();
}

```

```

void Print();
void Input_Table();
void Input_Element();
void Copy(Table tb);

```

```
};
```

ФАЙЛ TABLE.cpp

```

using namespace std;
#include <iostream>
#include "Table.h"
using namespace std;
Table::Table(int a, int b) {
    int size = a * b;
    m = a;
    n = b;
    allocate_memory(size);
}

Table::~Table() {
    delete [] T;
    m = 0;
    n = 0;
}

Table::Table(const Table& tb) {
    m = tb.m;
    n = tb.n;
    allocate_memory(m*n);
    for (int i = 0; i < (m*n); ++i) {
        T[i] = tb.T[i];
    }
}

void Table::Delete_column(int a) {
    int j = 0;
    int k = a;
    for (int i = 0; i < m * n; ++i) {
        if (k == (i + 1)){
            k += n;
            continue;
        }
        T[j] = T[i];
        j += 1;
    }
    n = n - 1;
    realloc(T, m * n); //урезание памяти после удаления столбца
}

void Table::Print() {
    cout << endl;
    cout << "----TABLE----";
    cout << " ";
    for (int i = 0; i < m * n; ++i) {
        if (i % n == 0) {
            cout << endl;
            cout << " ";
        }
        cout << T[i] << " ";
    }
    cout << endl;
    cout << "----TABLE----"<<endl;
}

```

```

void Table::Input_Element(){
    char s;
    int x,y;
    cout << "Input index: ";
    cin >> x >> y;
    cout << "Input volume: ";
    cin >> s;
    T[((x-1)*n + y - 1)] = s;
}

void Table::Copy(Table tb)
{
    tb.m = m;
    tb.n = n;
    tb.allocate_memory(m * n);
    for (int i = 0; i < (m * n); ++i) {
        tb.T[i] = T[i];
    }
}

void Table::allocate_memory(int size) {
    T = new char [size];
    for (int i = 0; i < size; ++i) {
        T[i] = '_';
    }
}

void Table::Delete_string(int a) {
    int j = 0;
    for (int i = 0; i < m*n; ++i) {
        if((((a - 1)*n + 1) <= (i+1)) && ((i + 1)<= ((a-1)*n + n))){
            continue;
        }
        T[j] = T[i];
        j += 1;
    }
    m = m - 1;
    realloc(T, m * n); //урезание памяти после удаления строки
}

void Table::Add_column() {
    n = n + 1;
    int j = 0;
    int data;
    int size = 0;
    realloc(T, m * n); //выделение памяти под новый столбец
    for (int i = m * (n - 1); i < m * n; ++i) {
        T[i] = '_';
    }
    for (int i = 0; i < m * n; i++) {
        if (((i+1) % n) == 0){
            data = T[i];
            T[j] = '_';
            for (int k = m * (n-1) + size; k > i+1; --k) {
                T[k] = T[k - 1];
            }
            size++;
            T[j + 1] = data;
            j = j + 2;
            i = i + 1;
            continue;
        }
        T[j] = T[i];
        j++;
    }
}

```



```

}

void Table::Add_string() {
    m = m + 1;
    realloc(T, m * n); //выделение памяти под новую строку
    for (int i = (m - 1)*n; i < m*n; ++i) {
        T[i] = '_';
    }
}

void Table::Input_Table() {
    cout << "Input number column: ";
    cin >> n;
    cout << endl;
    cout << "Input number string: ";
    cin >> m;
    cout << endl;
    for (int i = 0; i < m * n; ++i) {
        cin >> T[i];
    }
}

```