

1. Задание

8. Создать класс Board для описания шахматной доски. В нём предусмотреть массив 8x8 элементов и метод для перевода цифр 1-8 в буквы А-Н и обратно. На основе класса Board создать класс Composition для составления шахматной композиции. В нём предусмотреть возможность многократного добавления/удаления фигур на доску, распечатку композиции.

2. Листинг программы

```
#include <iostream>
#include<time.h>
#include<iomanip>
#include<vector>
#include<cstring>

#define _CRT_SECURE_NO_WARNINGS

using namespace std;

class Board
{
private:

public:
    const int N = 8, M = 8;
    char** arr = new char* [N+1];

    Board();
    ~Board();

};

Board::Board()
{
    srand(time(NULL));

    for (int i = 1; i <= N; i++)
    {
        arr[i] = new char[M+1];
    }

    for (int i = 1; i <= N; i++)
    {
        for (int j = 1; j <= M; j++)
        {
            if ((i + j) % 2 == 0)
            {
                arr[i][j] = '#'; // белая клетка
            }
            else
            {
                arr[i][j] = '@'; // чёрная клетка
            }
        }
    }
}

Board::~~Board()
{
    cout << "Вызвался деструктор" << endl;
```

```

        for (int i = 1; i <= N; i++)
        {
            delete[] arr[i];
        }
    }

class Composition : public Board
{
private:

public:

    void placement(const int n, const int m, const char value);
    void deleteplace(int n, int m);
    void print();
    int translate(const char* value);

};

void Composition::placement(const int n, const int m, const char value)
{
    arr[n][m] = value;
}

void Composition::deleteplace(int n, int m)
{
    if ((n + m) % 2 == 0)
    {
        arr[n][m] = '#'; // белая клетка
    }
    else
    {
        arr[n][m] = '@'; // чёрная клетка
    }
}

void Composition::print()
{
    cout << endl;
    for (int i = 1; i <= N; i++)
    {
        for (int j = 1; j <= N; j++)
        {
            cout << setw(5) << arr[i][j];
        }
        cout << endl;
    }
}

int Composition::translate(const char* value)
{
    int i = atoi(value);
    if (i >= 1 && i <= 8)
    {
        return i;
    }
    else if (!(strcmp(value, "A")) | !(strcmp(value, "a")))
    {
        return 1;
    }
    else if (!(strcmp(value, "B")) | !(strcmp(value, "b")))
    {
        return 2;
    }
    else if (!(strcmp(value, "C")) | !(strcmp(value, "c")))

```

```

    {
        return 3;
    }
    else if (!(strcmp(value, "D")) | !(strcmp(value, "d")))
    {
        return 4;
    }
    else if (!(strcmp(value, "E")) | !(strcmp(value, "e")))
    {
        return 5;
    }
    else if (!(strcmp(value, "F")) | !(strcmp(value, "f")))
    {
        return 6;
    }
    else if (!(strcmp(value, "G")) | !(strcmp(value, "g")))
    {
        return 7;
    }
    else if (!(strcmp(value, "H")) | !(strcmp(value, "h")))
    {
        return 8;
    }
    else
    {
        cout << "Ошибка ввода";
        abort();
    }
}

void menu()
{
    cout << "МЕНЮ:" << endl;
    cout << "1. Вывод композиции" << endl;
    cout << "2. Размещение фигуры" << endl;
    cout << "3. Удаление фигуры" << endl;
    cout << "0. Вывод композиции" << endl << endl;
}

int main()
{
    system("color F1");
    setlocale(LC_ALL, "Russian");

    Composition a;
    int choose = 0;
    char m[2], n[2];
    do
    {
        menu();

        cout << "Выберите действие: ";
        cin >> choose;
        cout << endl;
        if (choose < 0 || choose > 3)
        {
            cout << "Такого пункта в меню нет" << endl;
        }

        switch (choose)
        {
            case 1: // "Вывод композиции"
            {
                cout << "\t\tШахматная доска" << endl;
                a.print();
            }
        }
    }
}

```

```

        cout << endl;
        break;
    }
    case 2: // "размещение фигуры"
    {
        char str;
        cout << "Введите позицию фигуры [строка,столбец] и имя размещаемой
фигуры" << endl;
        cout << "Строка:" << endl;
        cin >> n;
        int i = a.translate(n);
        cout << "Столбец:" << endl;
        cin >> m;
        int j = a.translate(m);
        cout << "Фигура:" << endl;
        cin >> str;
        a.placement(i, j, str);
        cout << endl;
        break;
    }
    case 3:
    {
        cout << "Введите позицию фигуры [строка,столбец] для удаления" <<
endl;
        cout << "Строка:" << endl;
        cin >> n;
        int i = a.translate(n);
        cout << "Столбец:" << endl;
        cin >> m;
        int j = a.translate(m);
        a.deleteplace(i, j);
        cout << endl;
        break;
    }
}
} while (choose != 0);

system("pause");
return 0;
}

```

3. Приложение

МЕНЮ:

1. Вывод композиции
2. Размещение фигуры
3. Удаление фигуры
0. Вывод композиции

Выберите действие: 1

Шахматная доска

#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#

Рисунок 1 – Демонстрация работы программы

МЕНЮ:

1. Вывод композиции
2. Размещение фигуры
3. Удаление фигуры
0. Вывод композиции

Выберите действие: 2

Введите позицию фигуры [строка,столбец] и имя размещаемой фигуры

Строка:

1

Столбец:

1

Фигура:

К

Рисунок 2 – Демонстрация работы программы

МЕНЮ:

1. Вывод композиции
2. Размещение фигуры
3. Удаление фигуры
0. Вывод композиции

Выберите действие: 1

Шахматная доска

К	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#

Рисунок 3 – Демонстрация работы программы

МЕНЮ:

- 1. Вывод композиции
- 2. Размещение фигуры
- 3. Удаление фигуры
- 0. Вывод композиции

Выберите действие: 3

Введите позицию фигуры [строка,столбец] для удаления
Строка:

1

Столбец:

1

МЕНЮ:

- 1. Вывод композиции
- 2. Размещение фигуры
- 3. Удаление фигуры
- 0. Вывод композиции

Выберите действие: 1

Шахматная доска

#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#

Рисунок 4 – Демонстрация работы программы

Выберите действие: 2

Введите позицию фигуры [строка,столбец] и имя размещаемой фигуры

Строка:

A

Столбец:

2

Фигура:

P

МЕНЮ:

1. Вывод композиции
2. Размещение фигуры
3. Удаление фигуры
0. Вывод композиции

Выберите действие: 1

Шахматная доска

#	P	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#

Рисунок 5 – Демонстрация работы программы

1. Вывод композиции
2. Размещение фигуры
3. Удаление фигуры
0. Вывод композиции

Выберите действие: 3

Введите позицию фигуры [строка,столбец] для удаления
Строка:

A

Столбец:

1

МЕНЮ:

1. Вывод композиции
2. Размещение фигуры
3. Удаление фигуры
0. Вывод композиции

Выберите действие: 1

Шахматная доска

#	d	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#
#	@	#	@	#	@	#	@
@	#	@	#	@	#	@	#

Рисунок 6 – Демонстрация работы программы