



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 7
по курсу «Языки и методы программирования»
«Разработка простейшего класса на C++»

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Выполнение лабораторной работы заключается в составлении на языке C++ программы, состоящей из трёх файлов: - заголовочный файл `declaration.h` с объявлением одного из классов, приведённых в таблицах 1 – 16;

- файл `implementation.cpp` с определениями методов класса;
- файл `main.cpp`, содержащий функцию `main` и, возможно, вспомогательные функции для проверки работоспособности класса.

Реализация класса не должна опираться на стандартные контейнерные классы C++, то есть внутреннее состояние объектов класса должно быть реализовано через обычные массивы. Соответственно, в классе обязательно требуется реализовать:

- конструктор копий;
- деструктор (должен быть объявлен виртуальным);
- операцию присваивания.

Проверку работоспособности класса требуется организовать в функции `main`, размещённой в файле `main.cpp`.

Проверка должна включать в себя:

- создание объекта класса в автоматической памяти;
- передачу объекта класса по значению в функцию;
- присваивание объекта класса переменной.

Доска для игры в крестики-нолики размером $n \times n$ с операциями: 1. получение размера доски; 2. получение ссылки на клетку с указанными координатами; 3. определение, является ли текущая позиция финальной.

2 Результаты

Исходный код программы представлен в листинге 1, 2, 3

Листинг 1 — main.cpp

```
1 #include <iostream>
2 #include "TicTacToeBoard.h"
3 using namespace std;
4 int main() {
5     int n = 3;
6     TicTacToeBoard board(n);
7     board.getCell(0, 0) = 'X';
8     board.getCell(0, 0) = 'O';
9     board.getCell(1, 1) = 'O';
10    board.getCell(0, 2) = 'O';
11    board.getCell(2, 2) = 'X';
12    cout << "Size: " << board.getSize() << endl;
13    cout << "Cell (0, 0): " << board.getCell(0, 0) << endl;
14    cout << "Cell (1, 1): " << board.getCell(1, 1) << endl;
15    cout << "Is final position: " << board.isFinalPosition() << endl;
16    return 0;
17 }
```

Листинг 2 — класс TicTacToeBoard

```
1 class TicTacToeBoard {
2 private:
3     char **board;
4     int size;
5 public:
6     TicTacToeBoard(int n) {
7         size = n;
8         board = new char*[size];
9         for (int i = 0; i < size; i++) {
10             board[i] = new char[size];
11             for (int j = 0; j < size; j++) {
12                 board[i][j] = '_';
13             }
14         }
15     }
16     int getSize() {
17         return size;
18     }
19     char &getCell(int i, int j) {
20         return board[i][j];
21     }
22     bool isFinalPosition() {
23         for (int i = 0; i < size; i++) {
24             int j = 0;
25             while (j < size - 1 && board[i][j] == board[i][j + 1]) {
26                 j++;
27             }
28             if (j == size - 1 && board[i][j] != '_') {
29                 return true;
30             }
31         }
32     }
33 }
```

Листинг 3 — класс TicTacToeBoard(продолжение)

```
1      for (int j = 0; j < size; j++) {
2          int i = 0;
3          while (i < size - 1 && board[i][j] == board[i + 1][j]) {
4              i++;
5          }
6          if (i == size - 1 && board[i][j] != '_') {
7              return true;
8          }
9      }
10     int i = 0;
11     while (i < size - 1 && board[i][i] == board[i + 1][i + 1]) {
12         i++;
13     }
14     if (i == size - 1 && board[i][i] != '_') {
15         return true;
16     }
17     i = 0;
18     while (i < size - 1 && board[i][size - i - 1] == board[i + 1][
size - i - 2]) {
19         i++;
20     }
21     if (i == size - 1 && board[i][size - i - 1] != '_') {
22         return true;
23     }
24     for (int i = 0; i < size; i++) {
25         for (int j = 0; j < size; j++) {
26             if (board[i][j] == '_') {
27                 return false;
28             }
29         }
30     }
31     return true;
32 }
33 ~TicTacToeBoard() {
34     for (int i = 0; i < size; i++) {
35         delete [] board[i];
36     }
37     delete [] board;
38 }
39 };
```

Результат запуска представлен на рисунке 1, 2, 3

```

// main.cpp
#include <iostream>
#include "TicTacToeBoard.h"
using namespace std;

int main() {
    int n = 3;
    TicTacToeBoard board(n);
    board.getCell(0, 0) = 'X';
    board.getCell(0, 1) = 'O';
    board.getCell(1, 1) = 'O';
    board.getCell(0, 2) = 'O';
    board.getCell(2, 2) = 'X';
    cout << "Size: " << board.getSize() << endl;
    cout << "Cell (0, 0): " << board.getCell(0, 0) << endl;
    cout << "Cell (1, 1): " << board.getCell(1, 1) << endl;
    cout << "Is final position: " << board.isFinalPosition() << endl;
    return 0;
}

// TicTacToeBoard.h
class TicTacToeBoard {
private:
    char **board;
    int size;
public:
    TicTacToeBoard(int n) {
        size = n;
        board = new char**[size];
        for (int i = 0; i < size; i++) {
            board[i] = new char[size];
            for (int j = 0; j < size; j++) {
                board[i][j] = '_';
            }
        }
    }

    int getSize() {
        return size;
    }

    char &getCell(int i, int j) {
        return board[i][j];
    }

    bool isFinalPosition() {
        for (int i = 0; i < size; i++) {
            int j = 0;
            while (j < size - 1 && board[i][j] == board[i][j + 1]) {
                j++;
            }
            if (j == size - 1 && board[i][j] != '_') {
                return true;
            }
        }
        for (int j = 0; j < size; j++) {
            int i = 0;
            while (i < size - 1 && board[i][j] == board[i + 1][j]) {
                i++;
            }
            if (i == size - 1 && board[i][j] != '_') {
                return true;
            }
        }
        int i = 0;
        while (i < size - 1 && board[i][0] == board[i + 1][0]) {
            i++;
        }
        if (i == size - 1 && board[i][0] != '_') {
            return true;
        }
        int i = 0;
        while (i < size - 1 && board[i][size - 1] == board[i + 1][size - 1]) {
            i++;
        }
        if (i == size - 1 && board[i][size - 1] != '_') {
            return true;
        }
        return false;
    }

    void setCell(int i, int j, char c) {
        board[i][j] = c;
    }

    ~TicTacToeBoard() {
        for (int i = 0; i < size; i++) {
            delete[] board[i];
        }
        delete[] board;
    }
};

```

Рис. 1 — Реализация main.cpp и TicTacToeBoard.h

```

// TicTacToeBoard.h (continued)
    void setCell(int i, int j, char c) {
        board[i][j] = c;
    }

    ~TicTacToeBoard() {
        for (int i = 0; i < size; i++) {
            delete[] board[i];
        }
        delete[] board;
    }
};

```

Рис. 2 — Реализация main.cpp и TicTacToeBoard.h(продолжение)

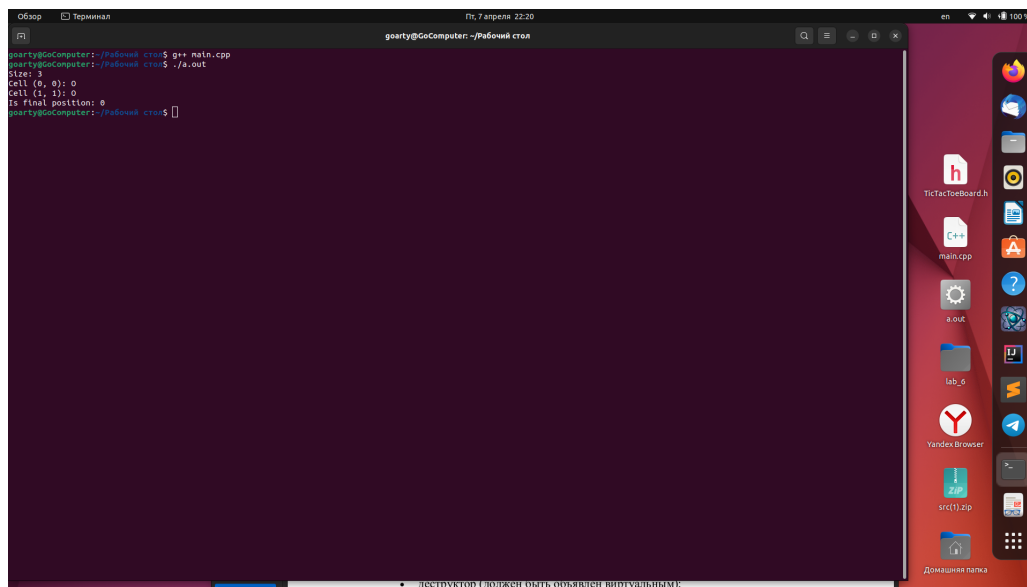


Рис. 3 — Работа программы