МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(национальный исследовательский университет)

Факультет «Компьютерные технологии, управление и радиоэлектроника»

Кафедра «Электронные вычислительные машины»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по дисциплине «Объектно-ориентированное программирование»

|  |  |
| --- | --- |
|  | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Беляков А.Е.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.  Автор проекта  Студент группы КЭ-114  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ковин К.П.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.  Работа защищена с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Беляков А.Е.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

Челябинск

2019

Южно-Уральский государственный университет

Факультет «Компьютерные технологии, управление и радиоэлектроника»

Кафедра «Электронные вычислительные машины»

Специальность «Информатика и вычислительная техника»

**ЗАДАНИЕ**

на курсовой проект

Студента группы КЭ-114

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дисциплина: Объектно-ориентированное программирование

Тема работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи работы: «\_\_» 2019 г.

Срок сдачи работы: «\_\_» \_\_\_\_\_\_\_\_\_ 2019 г.

Задание:

Написать программу, реализующую игру «Крестики-нолики» между двумя игроками: пользователем и компьютером (роботом). В программе использовать двусвязные списки.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание

[1. ОПИСАНИЕ РЕАЛИЗАЦИИ ШАБЛОНА 4](#_Toc413925142)

[2. Схема фрагмента алгоритма программы 5](#_Toc413925143)

[3. Листинг программы с комментариями 6](#_Toc413925144)

[4. Тестовые примеры 7](#_Toc413925145)

[5. Список литературы 8](#_Toc413925146)

# 1. ОПИСАНИЕ РЕАЛИЗАЦИИ ШАБЛОНА

Шаблоны позволяют определить конструкции (функции, классы), которые используют определенные типы данных, но на момент написания кода точно не известно, что это будут за типы. Другими словами, шаблоны позволяют определить универсальные конструкции, которые не зависят от определенного типа данных.

В конкретной задаче двусвязный список реализуется с помощью класса **dl\_list**. Класс **dl\_list** хранит структуру **unit**, указатели на первый (**head**) и последний (**tail**) элемент типа **unit**. Структура **unit** хранит в себе шаблонную переменную **data**, хранит указатели на предыдущий (**prev**) и следующий (**next**) элементы типа **unit**. Класс **dl\_list** мы собираемся типизировать структурой **game** который хранит в себе символ **cell** и индикатор свободного поля **occupied**.

Методы класса **dl\_list**:

**add** - добавление элемента в двусвязный список;

**remove** - удаление элемента из двусвязного списка;

**getHead** - возвращает указатель на начало двусвязного списка;

**getTail** - возвращает указатель на конец двусвязного списка;

Данного функционала более, чем достаточно для работы с двусвязным списком, а реализация класса через шаблон делает его универсальным контейнером для необходимых задач. В конкретном случае, для хранения игровых полей **game** в двусвязном списке.

Класс **tictac** хранит в себе переменную **win**, хранящую символ победителя, переменную **cell** структуры **game**, а также переменную **tmp** типа **dl\_list** типизированный структурой **game**.

Методы класса **tictac**:

**start** - инициализирует двусвязный список состоящий из структур **game**;

**step** - записывает ход в двусвязный список;

**show** - выводит игровое поле на консоль;

**result** - проверяет победную комбинацию;

**check** - проверяет наличие свободных полей;

**computer** - выбирает случайные координаты для хода компьютера;

**getWinner** - возвращает символ победителя;

# 2. Схема фрагмента алгоритма программы

Блок-схема алгоритма программы «Крестики нолики» показана на рисунке 1.



Рисунок 1

# 3. Листинг программы

#include <iostream>

#include <random>

using namespace std;

template <typename T>

class dl\_list

{

public:

struct unit

{

unit(T obj)

{

data = obj;

next = NULL;

prev = NULL;

}

T data;

unit\* next, \* prev;

};

dl\_list()

{

head = NULL;

tail = NULL;

}

~dl\_list()

{

if (head)

{

unit\* cur = head;

while (cur != NULL)

{

unit\* tmp = cur->next;

delete cur;

cur = tmp;

}

}

}

void add(T obj)

{

if (head == NULL)

{

head = tail = new unit(obj);

}

else

{

unit\* tmp = new unit(obj);

tail->next = tmp;

tmp->prev = tail;

tail = tmp;

}

}

void remove(unit\* obj)

{

if (obj->next != NULL)

{

obj->next->prev = obj->prev;

}

if (obj->prev != NULL)

{

obj->prev->next = obj->next;

}

delete obj;

}

unit\* getHead()

{

return head;

}

unit\* getTail()

{

return tail;

}

private:

unit\* head, \* tail;

};

struct game

{

char cell = '\_';

bool occupied = false;

};

class tictac

{

private:

char win;

dl\_list<game> tmp;

game cell;

public:

void start()

{

for (int i = 1; i < 10; i++)

{

tmp.add(cell);

}

}

void step(int x, int y, char symb, char user)

{

dl\_list<game>::unit\* cur = tmp.getHead();

if (y == 1)

{

for (int i = 0; i < x - 1; i++)

{

cur = cur->next;

}

}

else if (y == 2)

{

for (int i = 0; i < x + 2; i++)

{

cur = cur->next;

}

}

else if (y == 3)

{

for (int i = 0; i < x + 5; i++)

{

cur = cur->next;

}

}

if (cur->data.occupied)

{

if (user == 'c')

{

computer();

}

else

{

cout << "\nCell is occupied\n";

system("pause");

}

}

else

{

cur->data.occupied = true;

cur->data.cell = symb;

}

}

void show()

{

dl\_list<game>::unit\* cur = tmp.getHead();

int i = 1;

int j = 1;

cout << " ";

for (int ii = 1; ii < 4; ii++)

{

cout << " " << ii;

}

cout << "\n" << j << "|";

j++;

for (cur; cur != NULL; cur = cur->next)

{

if (i != 3)

{

cout << cur->data.cell << " ";

}

else

{

if (j != 4)

{

cout << cur->data.cell << "\n" << j << "|";

}

else

{

cout << cur->data.cell << "\n";

}

i = 0;

j++;

}

i++;

}

}

bool result()

{

dl\_list<game>::unit\* x1\_y1 = tmp.getHead();

dl\_list<game>::unit\* x2\_y1 = x1\_y1->next;

dl\_list<game>::unit\* x3\_y1 = x2\_y1->next;

dl\_list<game>::unit\* x1\_y2 = x3\_y1->next;

dl\_list<game>::unit\* x2\_y2 = x1\_y2->next;

dl\_list<game>::unit\* x3\_y2 = x2\_y2->next;

dl\_list<game>::unit\* x1\_y3 = x3\_y2->next;

dl\_list<game>::unit\* x2\_y3 = x1\_y3->next;

dl\_list<game>::unit\* x3\_y3 = tmp.getTail();

if (x1\_y1->data.cell == x2\_y1->data.cell && x1\_y1->data.cell == x3\_y1->data.cell && (x1\_y1->data.cell == 'X' || x1\_y1->data.cell == 'O'))

{

win = x1\_y1->data.cell;

return false;

}

else if (x1\_y2->data.cell == x2\_y2->data.cell && x1\_y2->data.cell == x3\_y2->data.cell && (x1\_y2->data.cell == 'X' || x1\_y2->data.cell == 'O'))

{

win = x1\_y2->data.cell;

return false;

}

else if (x1\_y3->data.cell == x2\_y3->data.cell && x1\_y3->data.cell == x3\_y3->data.cell && (x1\_y3->data.cell == 'X' || x1\_y3->data.cell == 'O'))

{

win = x1\_y3->data.cell;

return false;

}

else if (x1\_y1->data.cell == x2\_y2->data.cell && x1\_y1->data.cell == x3\_y3->data.cell && (x1\_y1->data.cell == 'X' || x1\_y1->data.cell == 'O'))

{

win = x1\_y1->data.cell;

return false;

}

else if (x3\_y1->data.cell == x2\_y2->data.cell && x3\_y1->data.cell == x1\_y3->data.cell && (x3\_y1->data.cell == 'X' || x3\_y1->data.cell == 'O'))

{

win = x3\_y1->data.cell;

return false;

}

else if (x1\_y1->data.cell == x1\_y2->data.cell && x1\_y1->data.cell == x1\_y3->data.cell && (x1\_y1->data.cell == 'X' || x1\_y1->data.cell == 'O'))

{

win = x1\_y1->data.cell;

return false;

}

else if (x2\_y1->data.cell == x2\_y2->data.cell && x2\_y1->data.cell == x2\_y3->data.cell && (x2\_y1->data.cell == 'X' || x2\_y1->data.cell == 'O'))

{

win = x2\_y1->data.cell;

return false;

}

else if (x3\_y1->data.cell == x3\_y2->data.cell && x3\_y1->data.cell == x3\_y3->data.cell && (x3\_y1->data.cell == 'X' || x3\_y1->data.cell == 'O'))

{

win = x3\_y1->data.cell;

return false;

}

return true;

}

bool check()

{

bool answer = true;

for (dl\_list<game>::unit\* cur = tmp.getHead(); cur != NULL; cur = cur->next)

{

if (cur->data.occupied == false)

{

answer = false;

break;

}

}

if (answer == false)

{

return false;

}

else

{

return true;

}

}

void computer()

{

random\_device gen;// Объявляем переменную gen типа (Генератор случайных чисел взят из библиотеки <random>)

uniform\_int\_distribution<int> uid(1, 3);// Применяется для указания диапазона целых чисел для рандома

char symb = 'O';

int x = uid(gen);

int y = uid(gen);

step(x, y, symb, 'c');

}

char getWinner()

{

return win;

}

};

int main()

{

tictac user;

user.start();

bool winner = true;

int x, y;

char symb = 'X';

do

{

if (!user.check())

{

do

{

system("cls");

cout << "Your step. Select cell\n\n";

user.show();

cout << "\nAxis x = "; cin >> x;

cout << "Axis y = "; cin >> y;

if ((x >= 1 && x <= 3) && (y >= 1 && y <= 3))

{

user.step(x, y, symb, 'u');

break;

}

else

{

cout << "\nWrong coordinates. Try again.\n";

system("pause");

}

} while (true);

}

if (!user.check())

{

user.computer();

}

else

{

if (!user.result())

{

winner = true;

}

else

{

winner = false;

}

system("cls");

user.show();

break;

}

if (!user.result())

{

winner = true;

system("cls");

user.show();

break;

}

} while (true);

if (winner)

{

cout << "\nWon \"" << user.getWinner() << "\"!";

}

else

{

cout << "\nThe winner is undetermined.";

}

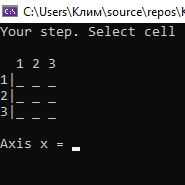
cout << endl << endl;

system("pause");

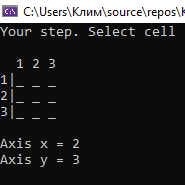
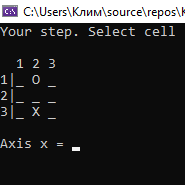
}

# 4. Тестовые примеры

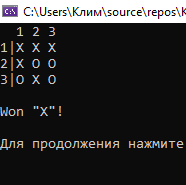
Начало игры:



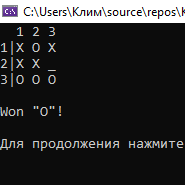
Ход пользователя и автоматический ход компьютера:

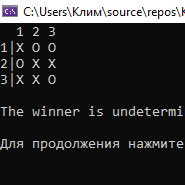
Победа пользователя:



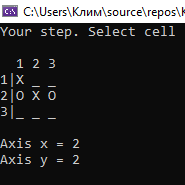
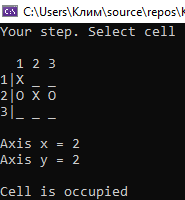
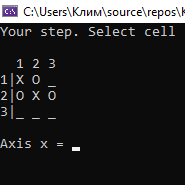
Победа компьютера:



Победитель не определён:

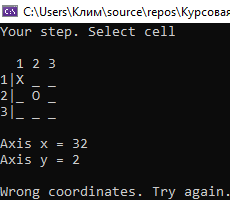


Ход в занятое поле:

В этом случае ход пользователя пропускается.

Пользователь ввёл неправильные координаты:



# 5. Список литературы

1. <http://www.quizful.net/post/random-number-generation-in-cpp11>
2. <http://cppstudio.com/post/5188/>
3. https://edu.susu.ru/pluginfile.php/106768/mod\_resource/content/20/Объектно-ориентированное%20программирование%20на%20C%2B%2B.pdf