

[**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**](https://dogm.mos.ru/)

Центральный административный округ

Государственное бюджетное профессиональное образовательное учреждение города Москвы "Московский автомобильно-дорожный колледж им. А.А. Николаева"

**Учебная Практика №2**

**Подготовил Ткачёв Денис**

**Группа 2/11ИП**

**Специальность: “Информационные системы и программирование (09.02.07)”**

**Практическое задание №3 (игра Пятнашки с Регистрацией)**

**Отчет 3**

**Задача № 3 «Пятнашки, с сохранением рекордов»**

**Задание:**

Реализовать хранение рекордов игры пятнашки в созданный для этого файл данных, предусмотреть возможность авторизации и регистрации игроков, реализовать хранения логинов и паролей игроков в отдельном зашифрованном файле. Рекорды должны перезаписываться. Реализовать возможность просмотра рекордов, всех играющих.

**Описание:**

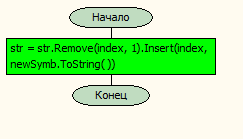
**Using System: -** Объявление **using System** дает возможность ссылаться на классы, которые находятся в пространстве имен **System**, так что их можно использовать, не добавляя **System**. перед именем типа.

**Функции:**

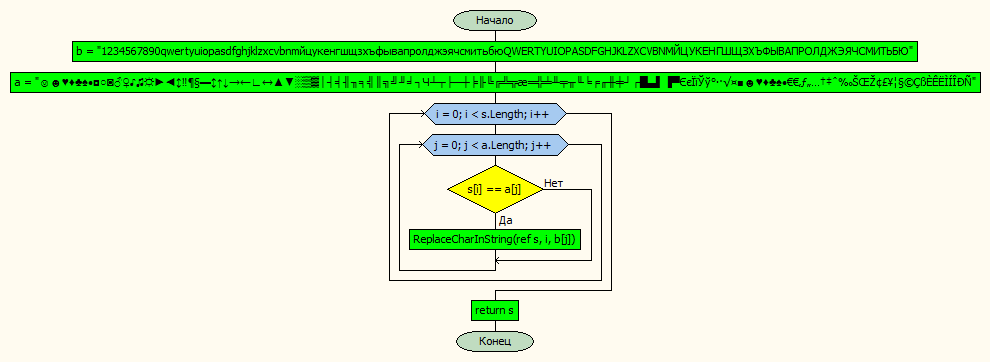
1. **void ReplaceCharInString (ref String str, int index, Char newSymb)** **–** метод выполняет замену символа в строке на определенный индекс.
2. **string shifrovka (string s) –** метод, который отвечает за замену символа логина и пароля, на другие символы, тем самым создавая шифрование.
3. **string deshifrovka (string s) –** метод выполняет шифрованных данных, на оригинальный логин и пароль.
4. **bool Isregistrated () –** метод разделяет потоки, строку на логин пароль и рекорды. Если логин и пароль подходит, то метод закрывает поток.
5. **void Createacount () –** Метод отвечает за создание нового аккаунта. Присутствует защита, при не вводе значений в поля логин и пароль, будет выдаваться сообщение об ошибке ввода.
6. **void Loadacount () –** Метод отвечает за чтение и ожидание ввода данных. При совпадении данных логина и пароля, то загружается аккаунт.
7. **void Saveacount () –** Метод отвечает за сохранение аккаунта и всех его значений.
8. **private void button19\_Click(object sender, EventArgs e) –** метод отвечает за вход – кнопку входа.
9. **private void button20\_Click (object sender, EventArgs e) –** Метод отвечает за Кнопку и открытие вкладки рекорды. И в вкладке находятся рекорды, лучшее время прохождения данной игры.

**Блок Схемы:**

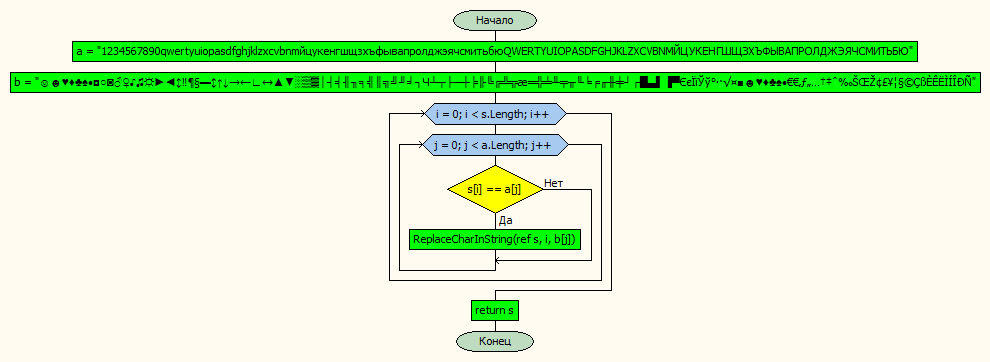
**void ReplaceCharInString**



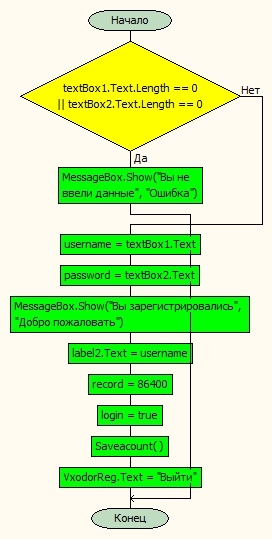
**Deshifrovka**

****

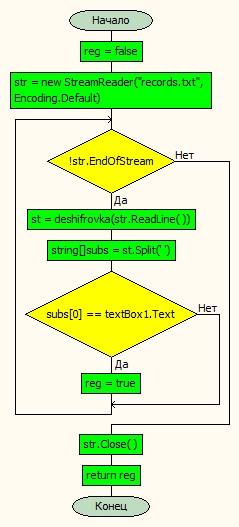
**Shifrovka**

****

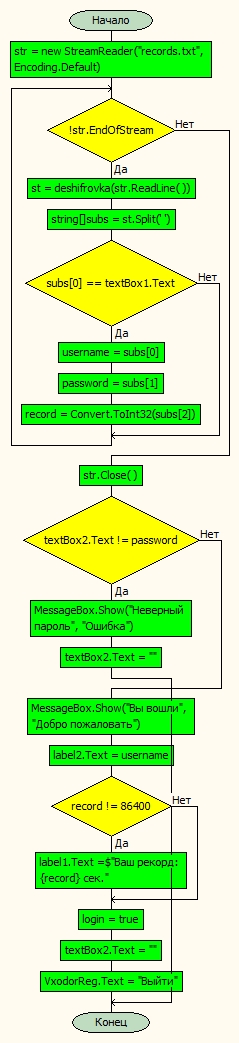
**Creatacount**

****

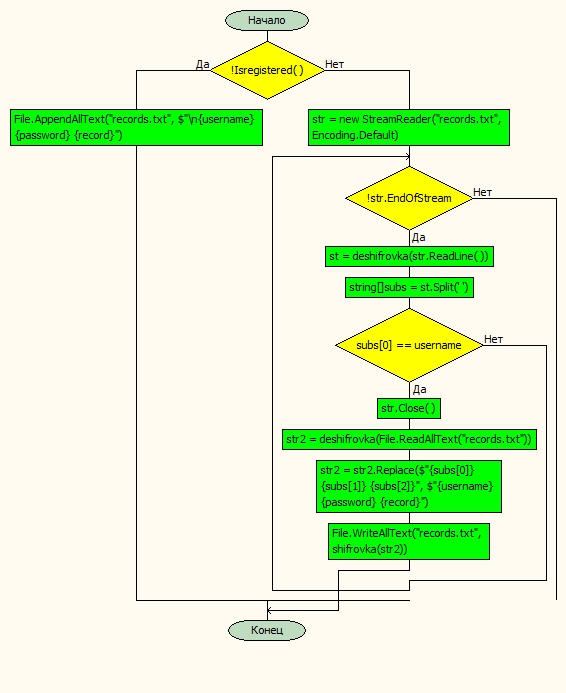
**Isregistrated**

****

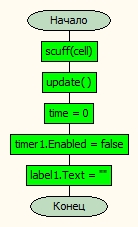
**Loadacount**

****

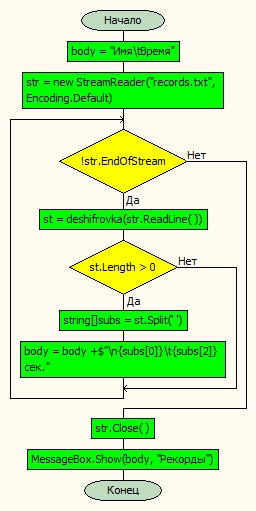
**Saveacount**

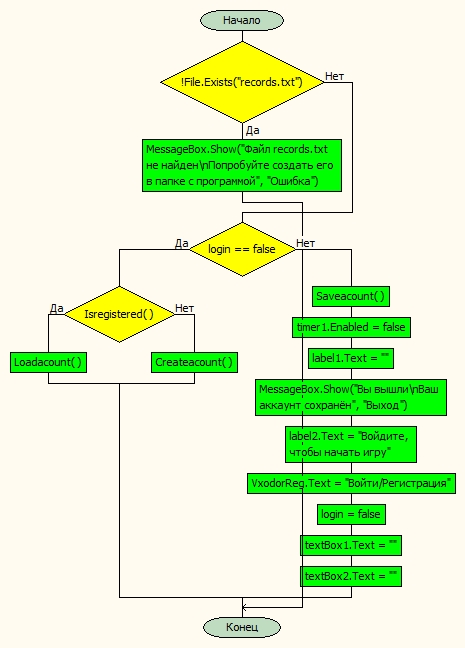
****

**Peremeshat\_Click\_Click**

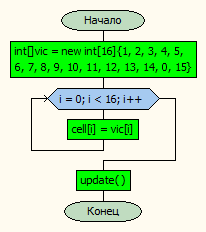
****

**Records\_Click**

****

**VxodorReg\_Click **

**Win\_Click\_Click**

****

**Код Программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO; // работа с файлами

namespace WinFormsApp1

{

public partial class Game15 : Form

{

bool login = false; // проверяем вошёл или нет

string username; // имя пользователя

string password; // пароль

int record = 86400; // рекорд

int time = 0;

int[] cell = new int[16] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 };

public Game15()

{

InitializeComponent();

update();

label1.Text = "";

}

void ReplaceCharInString(ref String str, int index, Char newSymb) // замена символа в строке по индексу

{

str = str.Remove(index, 1).Insert(index, newSymb.ToString());

}

string shifrovka(string s) // зашифровать строку

{

string a = "1234567890qwertyuiopasdfghjklzxcvbnmйцукенгшщзхъфывапролджэячсмитьбюQWERTYUIOPASDFGHJKLZXCVBNMЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮ";

string b = "☺☻♥♦♣♠•◘○◙♂♀♪♫☼►◄↕‼¶§▬↨↑↓→←∟↔▲▼░▒▓│┤╡╢╖╕╣║╗╝╜╛┐Ч┴┬├─┼╞╟╚╔╩╦æ═╬╧╨╤╥╙╘╒╓╫╪┘┌█▄▌▐▀ЄєЇїЎў°∙·√¤■☻♥♦♣♠•€€‚ƒ„…†‡ˆ‰ŠŒŽ¢£¥¦§©ÇßÈÊËÌÍÎÐÑ";

for (int i = 0; i < s.Length; i++) // перебираем все симаолы строки

{

for (int j = 0; j < a.Length; j++) // сравниваем с строкой a и шифруем

{

if (s[i] == a[j]) ReplaceCharInString(ref s, i, b[j]);

}

}

return s;

}

string deshifrovka(string s) // дешифровать строку

{

string b = "1234567890qwertyuiopasdfghjklzxcvbnmйцукенгшщзхъфывапролджэячсмитьбюQWERTYUIOPASDFGHJKLZXCVBNMЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЭЯЧСМИТЬБЮ";

string a = "☺☻♥♦♣♠•◘○◙♂♀♪♫☼►◄↕‼¶§▬↨↑↓→←∟↔▲▼░▒▓│┤╡╢╖╕╣║╗╝╜╛┐Ч┴┬├─┼╞╟╚╔╩╦æ═╬╧╨╤╥╙╘╒╓╫╪┘┌█▄▌▐▀ЄєЇїЎў°∙·√¤■☻♥♦♣♠•€€‚ƒ„…†‡ˆ‰ŠŒŽ¢£¥¦§©ÇßÈÊËÌÍÎÐÑ";

for (int i = 0; i < s.Length; i++)

{

for (int j = 0; j < a.Length; j++)

{

if (s[i] == a[j]) ReplaceCharInString(ref s, i, b[j]);

}

}

return s;

}

bool Isregistered()

{

bool reg = false;

StreamReader str = new StreamReader("records.txt", Encoding.Default); // построчно читаем файл

while (!str.EndOfStream)

{

string st = deshifrovka(str.ReadLine()); // дешифруем

string[] subs = st.Split(' '); // разделяем строку на "логин" "пароль" "рекорд"

if(subs[0] == textBox1.Text) reg = true; // если "логин" подходит

}

str.Close(); // закрываем поток

return reg;

}

void Createacount()

{

if (textBox1.Text.Length == 0 || textBox2.Text.Length == 0) // если текстовые поля пустые

{

MessageBox.Show("Вы не ввели данные", "Ошибка");

return;

}

username = textBox1.Text;

password = textBox2.Text;

MessageBox.Show("Вы зарегистрировались", "Добро пожаловать");

label2.Text = username;

record = 86400;

login = true;

Saveacount();

VxodorReg.Text = "Выйти";

}

void Loadacount()

{

StreamReader str = new StreamReader("records.txt", Encoding.Default); // построчно читаем файл

while (!str.EndOfStream)

{

string st = deshifrovka(str.ReadLine()); // дешифруем

string[] subs = st.Split(' ');

if (subs[0] == textBox1.Text) // если логин совпал, загружаем данные об аккаунте

{

username = subs[0];

password = subs[1];

record = Convert.ToInt32(subs[2]);

}

}

str.Close(); // закрываем поток

if (textBox2.Text != password) // если пароль неверный

{

MessageBox.Show("Неверный пароль", "Ошибка");

textBox2.Text = "";

return;

}

MessageBox.Show("Вы вошли", "Добро пожаловать");

label2.Text = username;

if (record != 86400) label1.Text = $"Ваш рекорд: {record} сек.";

login = true;

textBox2.Text = "";

VxodorReg.Text = "Выйти";

}

void Saveacount() // сохранение аккаунта

{

if (!Isregistered()) File.AppendAllText("records.txt", $"\n{username} {password} {record}"); // если не зарегистрирован, добавляем текст в конец файла

else

{

StreamReader str = new StreamReader("records.txt", Encoding.Default); // читаем построчно файл

while (!str.EndOfStream)

{

string st = deshifrovka(str.ReadLine());

string[] subs = st.Split(' ');

if (subs[0] == username) // ищем строку с логином

{

str.Close();

string str2 = deshifrovka(File.ReadAllText("records.txt"));

str2 = str2.Replace($"{subs[0]} {subs[1]} {subs[2]}", $"{username} {password} {record}"); // перезаписываем строку с инфой о юзере

File.WriteAllText("records.txt", shifrovka(str2));

return;

}

}

}

}

string celltext(int a)

{

string str;

if (a != 0) str = Convert.ToString(a); // преобразуем число в текст

else str = ""; // если число = 0, оставляем пустую ячейку

return str;

}

void update() // изменяем текст ячеек

{

field1.Text = celltext(cell[0]); field2.Text = celltext(cell[1]); field3.Text = celltext(cell[2]); field4.Text = celltext(cell[3]);

field5.Text = celltext(cell[4]); field6.Text = celltext(cell[5]); field7.Text = celltext(cell[6]); field8.Text = celltext(cell[7]);

field9.Text = celltext(cell[8]); field10.Text = celltext(cell[9]); field11.Text = celltext(cell[10]); field12.Text = celltext(cell[11]);

field13.Text = celltext(cell[12]); field14.Text = celltext(cell[13]); field15.Text = celltext(cell[14]); field16.Text = celltext(cell[15]);

}

void replace(int selected)

{

// если рядом с выбранной ячейкой пустая

// то меняем их местами

if (login == false)

{

MessageBox.Show("Вы не вошли", "Ошибка");

return;

}

if (selected == 0) // первая ячейка

{

if (cell[selected + 4] == 0) { cell[selected + 4] = cell[selected]; cell[selected] = 0; } // вниз

if (cell[selected + 1] == 0) { cell[selected + 1] = cell[selected]; cell[selected] = 0; } // вправо

}

else if (selected <= 3) // верхний ряд

{

if (cell[selected + 4] == 0) { cell[selected + 4] = cell[selected]; cell[selected] = 0; } // вниз

if (cell[selected + 1] == 0) { cell[selected + 1] = cell[selected]; cell[selected] = 0; } // вправо

if (cell[selected - 1] == 0) { cell[selected - 1] = cell[selected]; cell[selected] = 0; } // влево

}

else if (selected == 15) // последняя ячейка

{

if (cell[selected - 4] == 0) { cell[selected - 4] = cell[selected]; cell[selected] = 0; } // вверх

if (cell[selected - 1] == 0) { cell[selected - 1] = cell[selected]; cell[selected] = 0; } // влево

}

else if (selected >= 12) // нижний ряд

{

if (cell[selected - 4] == 0) { cell[selected - 4] = cell[selected]; cell[selected] = 0; } // вверх

if (cell[selected + 1] == 0) { cell[selected + 1] = cell[selected]; cell[selected] = 0; } // вправо

if (cell[selected - 1] == 0) { cell[selected - 1] = cell[selected]; cell[selected] = 0; } // влево

}

else // остальные ячейки

{

if (cell[selected - 4] == 0) { cell[selected - 4] = cell[selected]; cell[selected] = 0; } // вверх

if (cell[selected + 4] == 0) { cell[selected + 4] = cell[selected]; cell[selected] = 0; } // вниз

if (cell[selected + 1] == 0) { cell[selected + 1] = cell[selected]; cell[selected] = 0; } // вправо

if (cell[selected - 1] == 0) { cell[selected - 1] = cell[selected]; cell[selected] = 0; } // влево

}

if (timer1.Enabled == false) { timer1.Enabled = true; time = 0; label1.Text = "Игра началась!"; }

update();

if (Win() == true)

{

if (record > time) record = time;

label1.Text = $"Победа! ({time} сек., рекорд {record})";

timer1.Enabled = false;

MessageBox.Show("Ты выиграл!");

Saveacount();

}

}

bool Win() // проверка на победу

{

bool Win1 = true;

int[] vic = new int[16] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0 }; // отсортированный массив, "идеальное" расположение костей

//int[] vic2 = new int[16] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 100 };

for (int i = 0; i < 16; i++)

{

if (cell[i] != vic[i]) Win1 = false; // сравниваем каждое значение в ячейке

}

return Win1;

}

void scuff(int[] arr) // перемешать элементы массива: метод Фишера — Йетса

{

// создаем экземпляр класса Random для генерирования случайных чисел

Random rand = new Random();

for (int i = arr.Length - 1; i >= 1; i--)

{

int j = rand.Next(i + 1);

int tmp = arr[j];

arr[j] = arr[i];

arr[i] = tmp;

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

time += 1;

label1.Text = $"Времени прошло: {time} сек.";

}

private void peremeshat\_Click\_Click(object sender, EventArgs e)

{

scuff(cell); // перемешиваем

update(); // обновляем ячейки

time = 0; // сброс таймера

timer1.Enabled = false;

label1.Text = "";

}

private void Win\_Click\_Click(object sender, EventArgs e)

{

int[] vic = new int[16] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 0, 15 }; // отсортированный массив, 1 перемещение для победы

for (int i = 0; i < 16; i++)

{

cell[i] = vic[i]; // изменяем значения в ячейках для достижения победы

}

update(); // обновляем ячейки

}

private void VxodorReg\_Click(object sender, EventArgs e)

{

if (!File.Exists("records.txt")) // проверяем существует ли файл

{

MessageBox.Show("Файл records.txt не найден\nПопробуйте создать его в папке с программой", "Ошибка");

return;

}

if (login == false) // если не вошёл

{

if (Isregistered()) Loadacount(); // если зарегистрирован, загружаем акк

else Createacount(); // регистрируем

}

else // если уже вошёл и нажал на кнопку

{

Saveacount(); // сохраняем аккаунт

timer1.Enabled = false;

label1.Text = "";

MessageBox.Show("Вы вышли\nВаш аккаунт сохранён", "Выход");

label2.Text = "Войдите, чтобы начать игру";

VxodorReg.Text = "Войти/Регистрация";

login = false; // выходим из аккаунта

textBox1.Text = "";

textBox2.Text = "";

}

}

private void Records\_Click(object sender, EventArgs e)

{

string body = "Имя\tВремя";

StreamReader str = new StreamReader("records.txt", Encoding.Default); // построчно читаем файл

while (!str.EndOfStream)

{

string st = deshifrovka(str.ReadLine()); // дешифруем

if (st.Length > 0)

{

string[] subs = st.Split(' '); // делим на логин пароль рекорд

body = body + $"\n{subs[0]}\t{subs[2]} сек."; // преобразуем и добавляем к списку рекордов

}

}

str.Close();

MessageBox.Show(body, "Рекорды"); // показываем пользователю

}

private void field1\_Click\_1(object sender, EventArgs e) { replace(0); }

private void field2\_Click(object sender, EventArgs e) { replace(1); }

private void field3\_Click(object sender, EventArgs e) { replace(2); }

private void field4\_Click(object sender, EventArgs e) { replace(3); }

private void field5\_Click(object sender, EventArgs e) { replace(4); }

private void field6\_Click(object sender, EventArgs e) { replace(5); }

private void field7\_Click(object sender, EventArgs e) { replace(6); }

private void field8\_Click(object sender, EventArgs e) { replace(7); }

private void field9\_Click(object sender, EventArgs e) { replace(8); }

private void field10\_Click(object sender, EventArgs e) { replace(9); }

private void field11\_Click(object sender, EventArgs e) { replace(10); }

private void field12\_Click(object sender, EventArgs e) { replace(11); }

private void field13\_Click(object sender, EventArgs e) { replace(12); }

private void field14\_Click(object sender, EventArgs e) { replace(13); }

private void field15\_Click(object sender, EventArgs e) { replace(14); }

private void field16\_Click(object sender, EventArgs e) { replace(15); }

}

}