МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»

Направление подготовки 09.03.04
«Программная инженерия»

Утверждаю
Заведующий кафедрой ИС

_____Р.А.Валиев

_____г.

КУРСОВАЯ РАБОТА

по дисциплине:
«Программирование»
на тему:
«Разработка прикладной программы с использованием объектно-ориентированной технологии»

Автор:
студент группы 2171121

_____ А.Ф.Фатихов

Оценка: _____

Руководитель:
к.т.н., доцент кафедры ИС

_____ Е.В. Зубков

Дата защиты:
_____г.

Набережные Челны
2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»
Направление подготовки 09.03.04
«Программная инженерия»

Утверждаю
Заведующий кафедрой ИС

_____Р.А.Валиев

_____г.

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент
Фатихов Айрат Флюрович
1 Тема
«Разработка прикладной программы с использованием объектно-ориентированной технологии»
2 Срок представления к защите
_____г.
3 Исходные данные

4 Перечень подлежащих разработке вопросов

Задание выдано _____г. _____ Е.В. Зубков

Задание принято _____г. _____ А.Ф.Фатихов

Содержание

1.Проектирование программного обеспечения

Процесс проектирования программного обеспечения начинается с его структуры, которая отражает состав компонентов и их взаимодействие разрабатываемого программного обеспечения.

Для представления иерархии в программном обеспечении, была разработана структура взаимодействия интерфейса с классами. В список интерфейса входят:

1. OneGameShow;
2. twoSettingShow;
3. threeOutputForm;
4. allMethodTwoForm;
5. allMethodOneForm.

Первый интерфейс включает в себя метод «SettingOneFormShow», который реализуется с помощью наследования класса «GameShow». Этот класс отвечает за запуск новой формы, при нажатии на кнопку.

Второй интерфейс включает в себя метод «SettingTwoFormShow», который реализуется при наследовании классом «SettingShow», для вызова нового окна, но для другого пункта.

Третий интерфейс содержит в себе метод «SettingOutputClose», но реализуется при наследовании классом «OutputClose», для выхода с программы.

Четвертый интерфейс содержит в себе пять методов, которые реализуются с помощью класса «SettingTwoFormButtonMethod»:

1. MethodHideOkButton;
2. MethodHideCancelButton;
3. MethodAddOkFailGo;
4. MethodCancelFailCancel;
5. MethodScrollBar.

Первый метод при нажатии на кнопку «Ok», скрывает запущенное окно, также и метод «MethodHideCancelButton» предназначен для скрытия окна.

Третий метод «MethodAddOkFailGo», реализует выбор директории.

Четвертый метод «MethodCancelFailCancel» очищает список, который был добавлен с директории.

Пятый метод, реализует прокручивание добавленного списка сверху вниз, а также снизу вверх.

Последний интерфейс «allMethodOneForm», определяет метод «musicListPlay», который наследуется от класса «SettingTwoFormShow», с помощью которого при нажатии на кнопку процесс ставится на паузу, также в этом классе метод «musicPlay», при нажатии на кнопку продолжает процесс, а при методе «musicNext», при нажатии на кнопку переходит на другой процесс.

После описания взаимодействия интерфейса и класса, покажем эту реализацию на UML диаграмме, которая даст полное понимание содержимого.

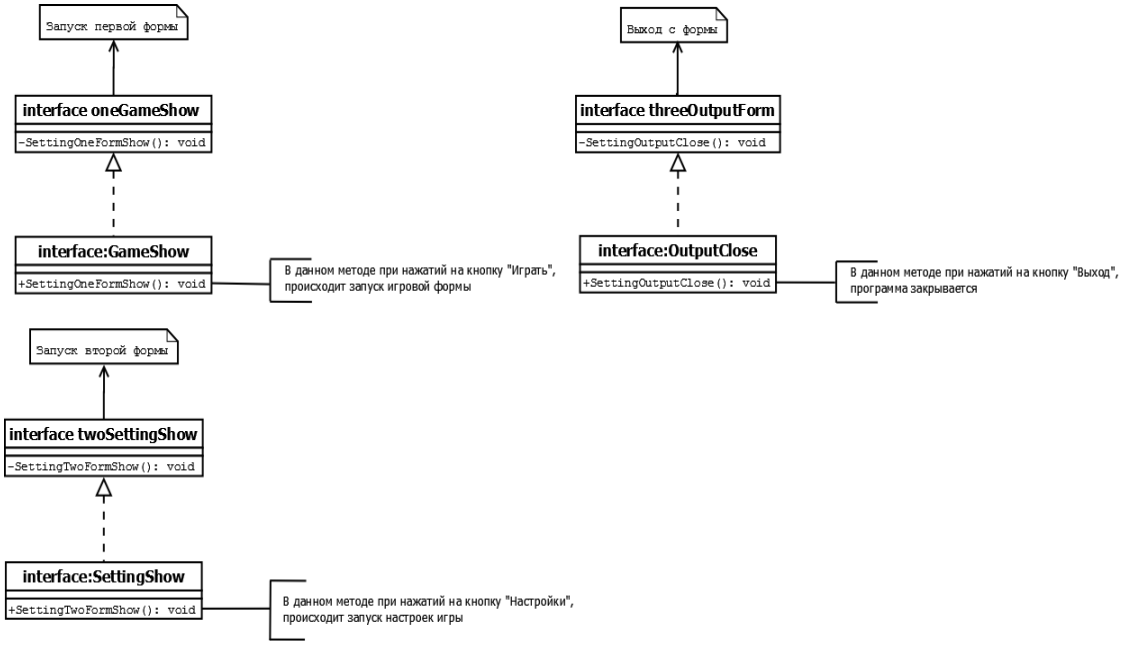

Рисунок 1 – UML диаграмма первой части

Первая и вторая часть UML диаграммы разделены на две части для того чтобы показать более примитивные действия такие как вызов нового окна, закрытия окна, а вторая часть показывает более сложные команды и их выполнение.

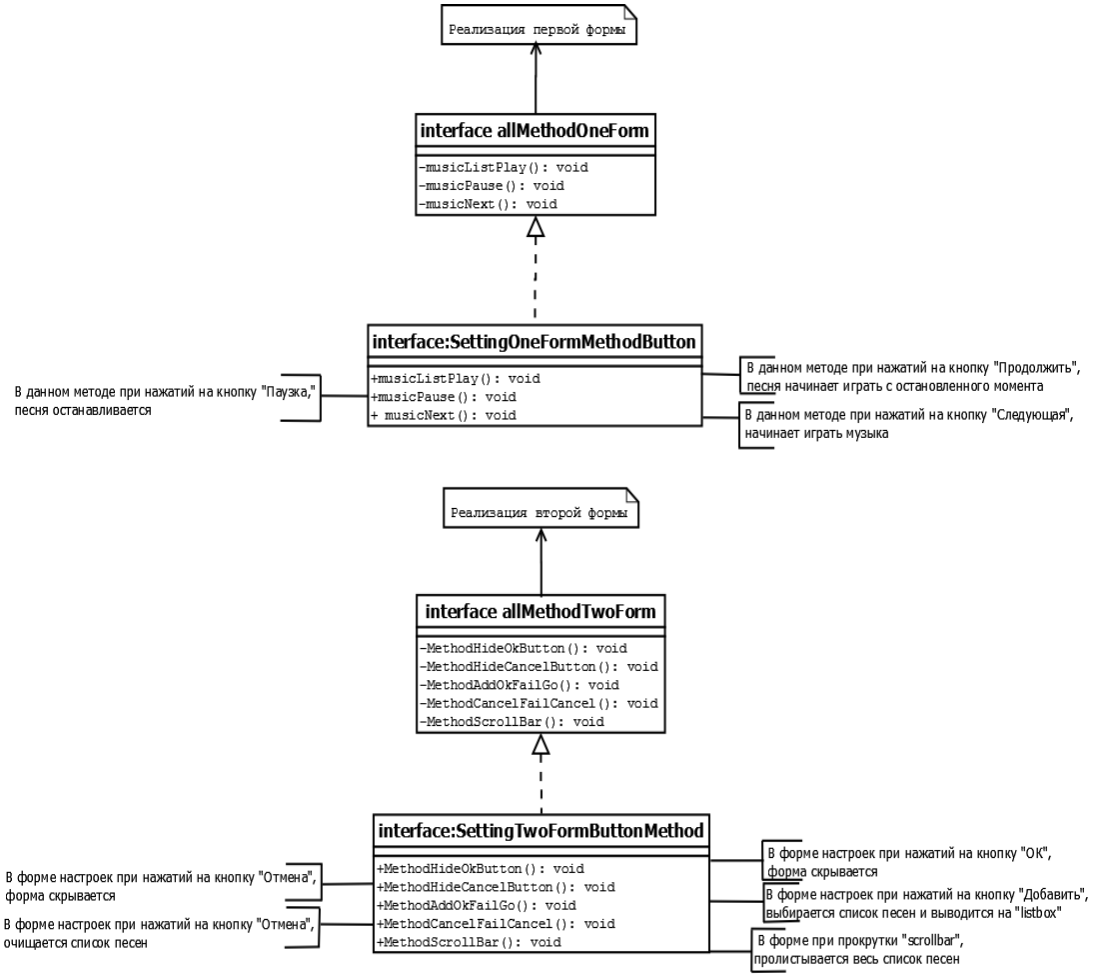

Рисунок 2 – UML диаграмма второй части

2. Графическое представление

Алгоритм можно представить разными способами:

1. с помощью графического представления;
2. с помощью словесного описания.

Для данного программного обеспечения графическое представление описано с помощью блок-схемы.

Как описывалось в первой главе, загрузка списка с директории выводится на «listbox» - это самый главный процесс, в программе.

Для выбора списка с директории создается объект класса «». Следующим шагом выполняется условие, если в каталогах выбран список с расширением «mp3», то музыка добавляется в массив. До того как список будет загружен в «listbox» его нужно очистить и следующим шагом будет очистка «listbox». После того список песен с директории будет выгружен в «listbox». Таким образом, когда список уже выгружен, необходимо очистить массив для быстрой работы программы, если каждый раз в этот массив добавлять больше и больше песен, то программа начнет медленно работать.

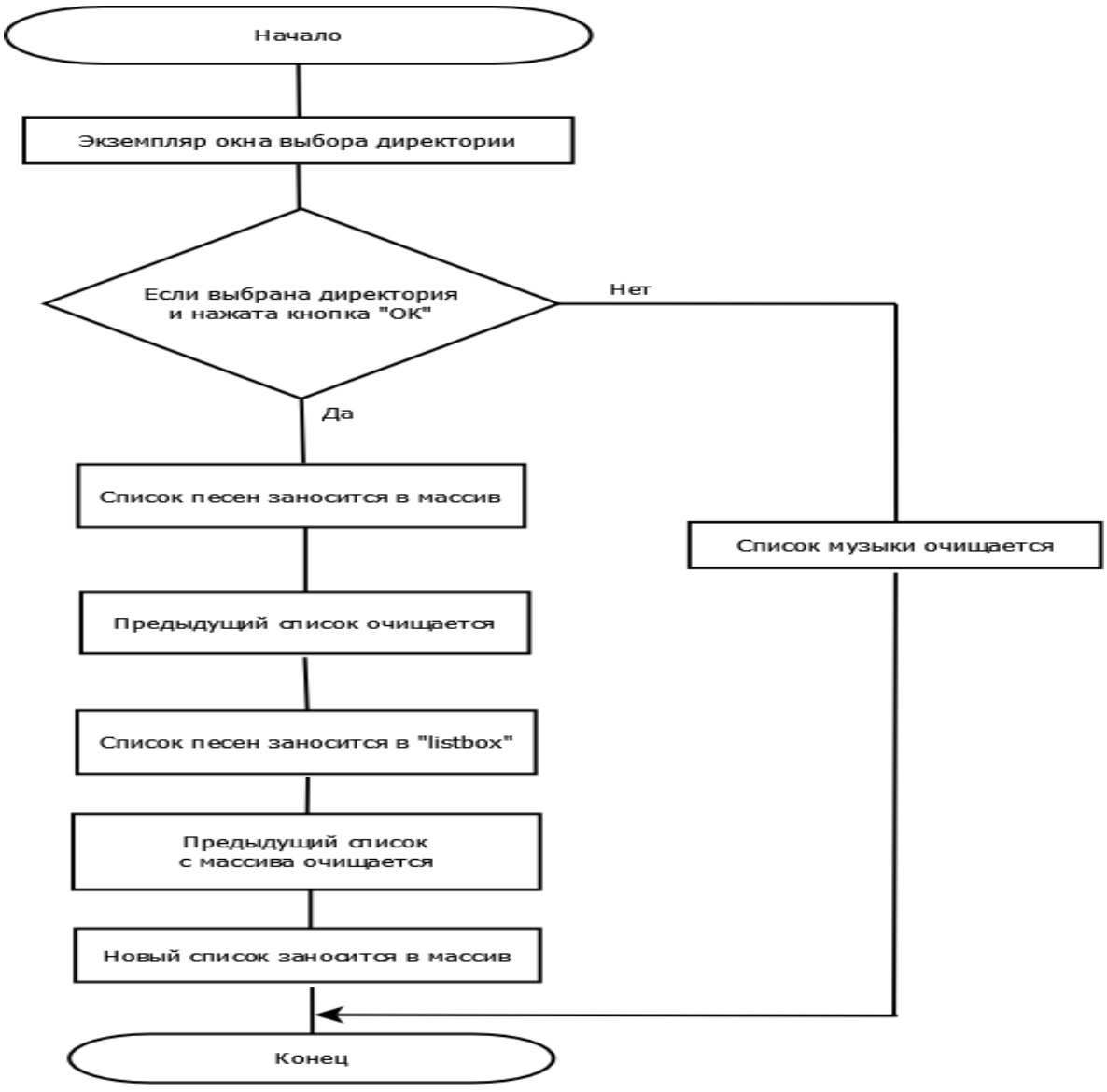

Рисунок 3 – Блок-схема выгрузки списка с директории

Следующей важной операцией в программе является угадывание песни.

Если первый игрок нажмет кнопку «A» на клавиатуре, то ему присваивается балл, но игрок может и ошибиться при угадывании песни, то в этом случае, очко снимается с игрока.

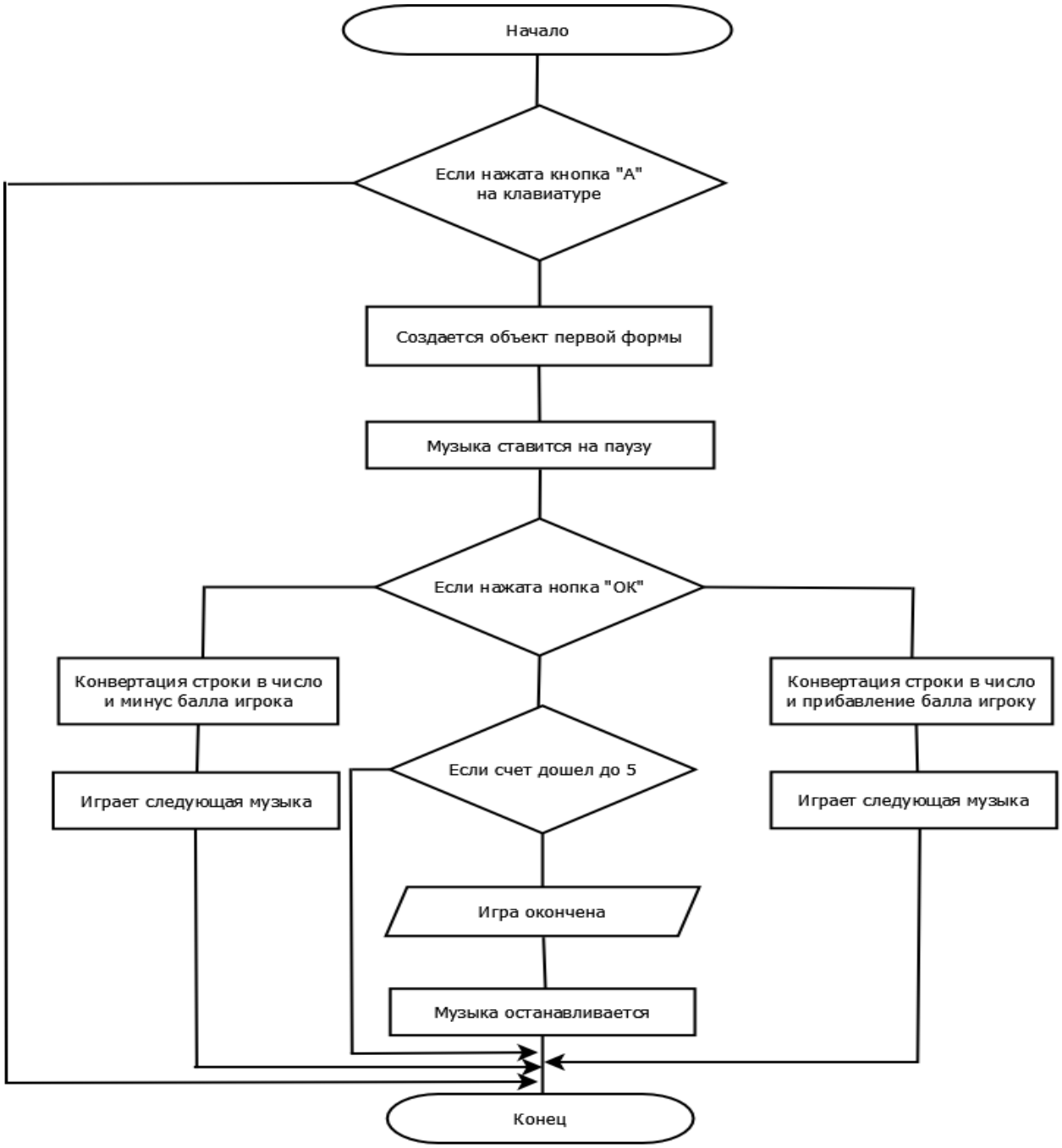

Рисунок 4 – блок-схема при нажатии на кнопку «A»

Для второго игрока происходит такая же операция, но он должен нажать кнопку «B» на клавиатуре.

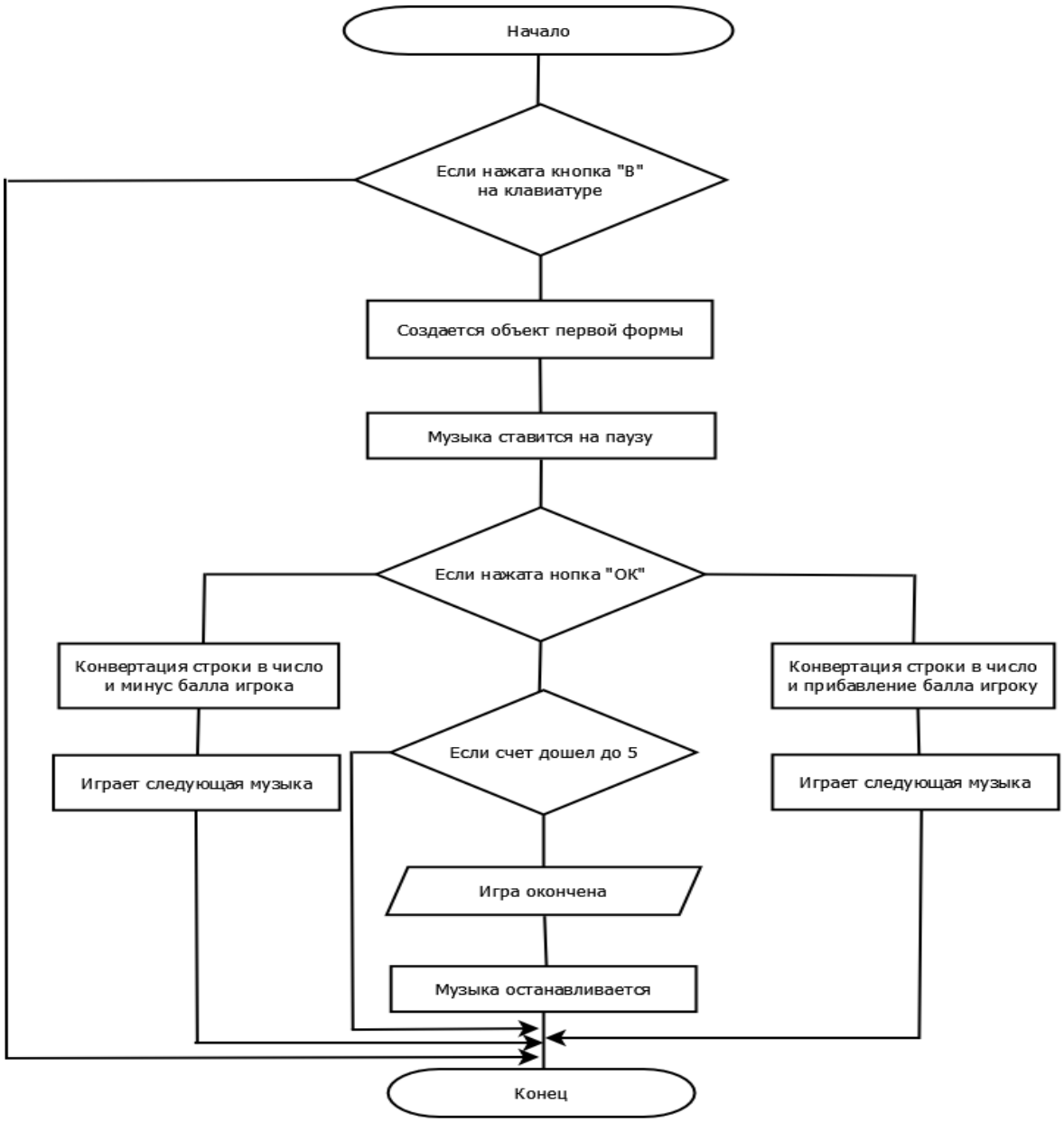

Рисунок 5 – блок-схема при нажатии на кнопку «B»

3. Листинг программы

В листинге представлен исходный код программы написанный на языке C#, который дает понятие реализацию программы.

Код:

```
namespace ArtApLL_LIcense_
{
   public partial class FormOne : Form
   {
      public FormOne()
      {
```

```
            InitializeComponent();
            SaveForm.form = this; //присваиваем ссылку на форму для доступа внутри класса
            SaveForm.settingoneform = new SettingOneForm();//создаётся объект первой формы
            SaveForm.settingtwoform = new SettingTwoForm();//создаётся объект второй формы
        }
        private void Form1_Load(object sender, EventArgs e)
        {
        }
        private void buttonGame_Click(object sender, EventArgs e)
        {
            GameShow gameShowButton = new GameShow();
            gameShowButton.SettingOneFormShow();
        }
        private void buttonSetting_Click(object sender, EventArgs e)
        {
            SettingShow settingShowButton = new SettingShow();
            settingShowButton.SettingTwoFormShow();
        }
 private void buttonOutput_Click(object sender, EventArgs e)
        {
            OutputClose outputCloseFormButton = new OutputClose();
            outputCloseFormButton.SettingOutputClose();
        }
    }
    /// <summary>
    /// Создание ссылки и объекта на форму
    /// </summary>
    public struct SaveForm //создаём структуру класса для хранения объекта формы
    {
        public static Form form; //объект формы
        public static SettingOneForm settingoneform; //объект первой формы
        public static SettingTwoForm settingtwoform;//объект второй формы
    }

    /// <summary>
    /// интерфейс содержащий метод для вызова первой формы
    /// </summary>
    public interface oneGameShow
    {
        void SettingOneFormShow();
```

```
}
/// <summary>
/// интерфейс для вызова второй формы
/// </summary>
public interface twoSettingShow
{
    void SettingTwoFormShow();
}
/// <summary>
/// интерфейс для закрытия формы
/// </summary>
public interface threeOutputForm
{
    void SettingOutputClose();
}
/// <summary>
/// интерфейс содержащий методы для работы со второй формой
/// </summary>
public interface allMethodTwoForm
{
    void MethodHideOkButton();
    void MethodHideCancelButton();
    void MethodAddOkFailGo();
    void MethodCancelFailCancel();
    void MethodScrollBar();
}
public interface allMethodOneForm
{
    void musicListPlay();
    void musicPause();
    void musicNext();
}
/// <summary>
/// класс для вызова первой формы
/// </summary>
public class GameShow : oneGameShow //класс для вызова первой формы
{
    public void SettingOneFormShow()
    {
        SaveForm.settingoneform.ShowDialog();
    }
}
/// <summary>
/// класс для вызова второй формы
/// </summary>
```

```
    public class SettingShow : twoSettingShow //класс для вызова второй формы
    {
        public void SettingTwoFormShow()
        {
            SaveForm.settingtwoform.ShowDialog();
        }
    }
    /// <summary>
    /// класс для закрытия формы
    /// </summary>
    public class OutputClose : threeOutputForm //класс для закрытия формы
    {
        public void SettingOutputClose()
        {
            SaveForm.form.Close();
        }
    }
    /// <summary>
    /// класс в котором будут все методы к второй форме
    /// </summary>
    public class SettingTwoFormButtonMethod : allMethodTwoForm //класс в
котором будут все методы к второй форме
    {
        public void MethodHideOkButton() //метод для скрытия формы
        {
            SaveForm.settingtwoform.Hide();
        }
        public void MethodHideCancelButton() //метод для скрытия формы
        {
            SaveForm.settingtwoform.Hide();
        }
        public void MethodAddOkFailGo() //метод в котором проверяется, если
нажал кнопку ок то папка выводится на листбокс, иначе очищается
        {
            FolderBrowserDialog folderbrowserdialog = new FolderBrowserDialog();
            if (folderbrowserdialog.ShowDialog() == DialogResult.OK) //если
выбрана нужная дериктория, то музыка добавляется
            {

                string[] massivFail =
Directory.GetFiles(folderbrowserdialog.SelectedPath, "*.mp3",
SaveForm.settingtwoform.checkBoxOne.Checked ? SearchOption.AllDirectories :
SearchOption.TopDirectoryOnly);
                SaveForm.settingtwoform.listBoxOne.Items.Clear();
                SaveForm.settingtwoform.listBoxOne.Items.AddRange(massivFail);
```

```
                SettingOneFormMethodButton.massivMusic.Clear();
                SettingOneFormMethodButton.massivMusic.AddRange(massivFail);

            }
            else if (folderbrowserdialog.ShowDialog() == DialogResult.Cancel) //иначе музыку просто очищается
            {
                SaveForm.settingtwoform.listBoxOne.Items.Clear();
            }
        }
        public void MethodCancelFailCancel() //если музыку добавил, но она неправильная, то при нажатий отмена она исчезнет
        {
            SaveForm.settingtwoform.listBoxOne.Items.Clear();
        }
        public void MethodScrollBar() //Метод дял перемещения списка вверх и вниз
        {
            SaveForm.settingtwoform.listBoxOne.TopIndex = SaveForm.settingtwoform.vScrollBar1.Value;
            SaveForm.settingtwoform.listBoxOne.TopIndex = SaveForm.settingtwoform.listBoxOne.Items.Count - 1;
            SaveForm.settingtwoform.listBoxOne.SelectedIndex = SaveForm.settingtwoform.listBoxOne.Items.Count - 1;
        }
    }

    /// <summary>
    /// Класс первой формы, в котором будут все методы реализации программы
    /// </summary>
    public class SettingOneFormMethodButton : allMethodOneForm
    {
        static public List<string> massivMusic = new List<string>(); //Добавляем музыку в объект List
        public void musicListPlay()
        {
            Random rnd = new Random(); //Воспроизводим песни по рандому
            int number = rnd.Next(0, SettingOneFormMethodButton.massivMusic.Count);
            SaveForm.settingoneform.WMP.URL = SettingOneFormMethodButton.massivMusic[number];
            SettingOneFormMethodButton.massivMusic.RemoveAt(number);
            SaveForm.settingoneform.labelNumberMusic.Text = SettingOneFormMethodButton.massivMusic.Count.ToString();
```

```
        }
        public void musicPause() //Ставим на паузу музыку
        {
            SaveForm.settingoneform.WMP.Ctlcontrols.pause();
        }
        public void musicNext() //Продолжаем музыку
        {
            SaveForm.settingoneform.WMP.Ctlcontrols.play();
        }
    }
}


public partial class SettingOneForm : Form
    {
        public SettingOneForm()
        {
            InitializeComponent();
        }

        private void buttonNext_Click(object sender, EventArgs e)
        {
            SettingOneFormMethodButton sofmb = new
SettingOneFormMethodButton();
            sofmb.musicListPlay();
        }

        private void labelNumberMusic_Click(object sender, EventArgs e)
        {
            SettingOneFormMethodButton sofmb = new
SettingOneFormMethodButton();
            sofmb.musicListPlay();
        }

        private void buttonPause_Click(object sender, EventArgs e)
        {
            SettingOneFormMethodButton sofmb = new
SettingOneFormMethodButton();
            sofmb.musicPause();
        }
        private void buttonContinue_Click(object sender, EventArgs e)
        {
            SettingOneFormMethodButton sofmb = new
SettingOneFormMethodButton();
            sofmb.musicNext();
```

```
        }

        private void SettingOneForm_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyData == Keys.A) //если нажали букву A, то выходит форма с кнопкой Ок или Выход
            {
                SettingOneFormMethodButton sofmb = new SettingOneFormMethodButton();
                sofmb.musicPause();
                if (MessageBox.Show("Ответ первого игрока", "", MessageBoxButtons.OKCancel) == DialogResult.OK) //если ОК, то увел. счет и музыка играет след
                {
                    labelCountOne.Text = Convert.ToString(Convert.ToInt32(labelCountOne.Text)+1);
                    sofmb.musicNext();
                }
                else //иначе счет уменьшается
                {
                    labelCountOne.Text = Convert.ToString(Convert.ToInt32(labelCountOne.Text) - 1);
                    sofmb.musicNext();
                }
                if (labelCountOne.Text == Convert.ToString(5)) //если счет дошел до 5, то просто выводим сообщение и ост.песню
                {
                    MessageBox.Show("Игра окончена, первый игрок выйграл!!!");
                    sofmb.musicPause();
                }
            }
            if (e.KeyData == Keys.B) //если нажали букву B, то выходит форма с кнопкой ОК или Выход
            {
                SettingOneFormMethodButton sofmb = new SettingOneFormMethodButton();
                sofmb.musicPause();
                if (MessageBox.Show("Ответ второго игрока!!!", "", MessageBoxButtons.OKCancel) == DialogResult.OK) //если ОК, то увел. счет и музыка играет след
                {
                    labelCountTwo.Text = Convert.ToString(Convert.ToInt32(labelCountTwo.Text)+1);
                    sofmb.musicNext();
                }
```

```
            else //иначе счет уменьшается
            {
                labelCountTwo.Text =
Convert.ToString(Convert.ToInt32(labelCountTwo.Text) - 1);
                sofmb.musicNext();
            }
            if (labelCountTwo.Text == Convert.ToString(5)) //если счет дошел до
5, то просто выводим сообщение и ост.песню
            {
                MessageBox.Show("Игра окончена, второй игрок выйграл!!!");
                sofmb.musicPause();
            }
        }
        }

    private void labelCountOne_Click(object sender, EventArgs e)
    {
        SettingOneFormMethodButton sofmb = new
SettingOneFormMethodButton();
        sofmb.musicPause();
    }

    private void labelCountTwo_Click(object sender, EventArgs e)
    {
        SettingOneFormMethodButton sofmb = new
SettingOneFormMethodButton();
        sofmb.musicPause();
    }
    private void label1_Click(object sender, EventArgs e)
    {
    }
    }
}
```

4.Результат работы программного обеспечения

После запуска программного обеспечения, который представляет викторину, где необходимо выбрать список музыки, и кто первый отгадает мелодию, тому присваивается балл. Кто из игроков достиг определенного количества балла, этот игрок выигрывает.

Для конкретного понимания программы, пройдемся по всем пунктам программы. Перед пользователем стоит три выбора, закрыть программу,

настроить и выбрать в нем список песен и самый главный пункт это отгадать песню.

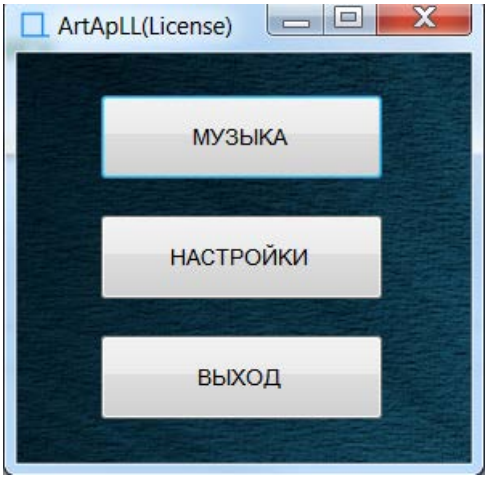

Рисунок 6 – Меню программы

При переходе по кнопке «Настройки», предстоит выбрать каталог песен, но для этого необходимо дойти до самой конечной папки, где находится список песен, но если до выбора поставить флажок на «дерево папок», то выбрав одну директорию, программа сама найдет все списки песен, которые лежат в дереве папок.

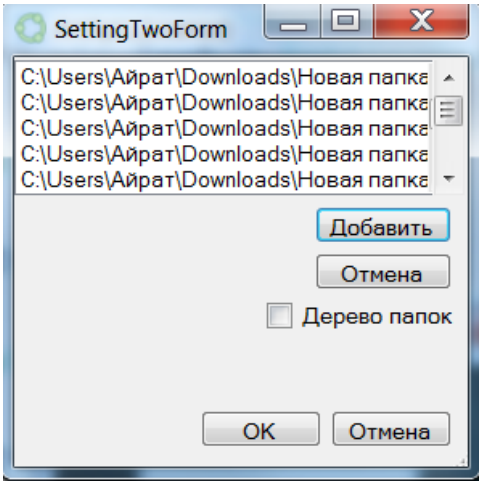

Рисунок 7 – Меню настроек

После того как выбрали список песен, при нажатии кнопки «Ок», песни сохраняются. Дальше переходим в последний пункт «Музыка», где происходят самые основные действия.

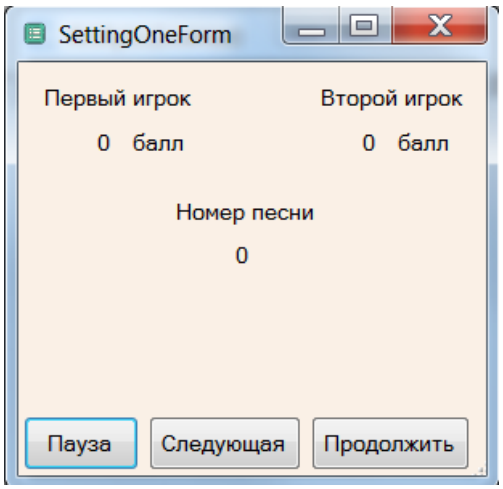

Рисунок 8 – меню «музыка»

В данном окне играют два игрока, у каждого имеется свое количество баллов. Снизу три кнопки: пауза, следующая, продолжить. При нажатии на кнопку «пауза» песня останавливается и если угадал песню первый игрок, то он должен нажать на клавиатуре кнопку «A» и при правильном ответе ему присваивается один балл. Если первый ответил второй игрок, то он должен нажать на клавиатуре кнопку «B».

При нажатии на кнопку следующая производится следующая песня в списке и также отгадывается. В том случае, если игрок ответил не правильно и не отгадал мелодию, то с него снимается один балл и отгадывание песни продолжается до тех пор, пока игроки не отгадают мелодию.

Также в окне отображается количество песен, чтобы игроки знали, сколько раундов им осталось до конца игры.

После окончания игры, программа вы водит результат игрока и определяется победитель викторины, и выходим с игры при нажатии на кнопку «выход».

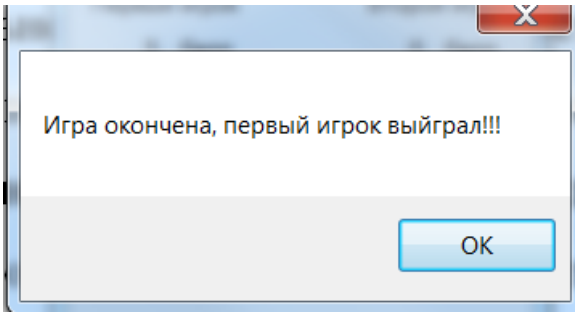

Рисунок 9 – результат игры

Заключение

В ходе выполнения курсовой работы было спроектировано программное обеспечение, которая описывает, из каких классов будет состоять приложение, модулей, переменных и их взаимодействие.

Выполнение всех операции, было представлено в блок-схеме, которая описывается в графическом представлении программы.

Код приложения, был описан в заголовке листинг программы, также добавлены комментарий для понимания работы приложения и описаны подсказки для каждого класса и метода.

Так же, для полного представления, как выглядит программное обеспечение, какой у него интерфейс, какая форма, это все описывается в главе результаты работы программного обеспечения.

Таким образом, выполняя все этапы разработки программного обеспечения, на выходе мы получили приложение, который заносит список песен с директории в форму приложения, затем два игрока при угадывании песни получают по баллу, а набрав определенное количество баллов, определяется победитель игры.