**Итоговая работа**

**МДК.02.02 Инструментальные средства разработки программного обеспечения**

**Цель:** инспекция кода модулей проекта, документирование результатов тестирования.

**Задачи:**

1. Составить техническое задание (шаблон прилагается) для программного проекта Будильник

2. Реализовать программный проект Будильник поэтапно (исходя из теоретических сведений, его необходимо доработать)

3. Составить репортинг Ручного тестирования в виде таблицы.

**Выполнение работы**

*1. Текст программы*

Программа «Будильник» написана на языке C# с использованием библиотеки Windows Forms (WinForms) на платформе .NET Framework. В листингах 1-3 приведены фрагменты кода, иллюстрирующие основные принципы работы: создание интерфейса, запуск и остановка будильника, воспроизведение выбранного звукового файла с помощью Windows Media Player.

Комментарии внутри кода поясняют назначение каждого метода и переменных, отвечающих за логику программы. Код разбит на несколько файлов:

* MainForm.cs – основной файл с логикой работы интерфейса и будильника.
* MainForm.Designer.cs – автоматически сгенерированный файл с описанием визуальной части формы.
* AlarmItem.cs – класс, описывающий структуру данных «Будильник».

Само приложение при компиляции представляет собой исполняемый файл (WinForms\_Alarm.exe) и дополнительные библиотеки (ссылки на .NET Framework и Windows Media Player).

Листинг 1 – Код файла MainForm.cs

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* MainForm.cs

\*

\* Основная логика работы будильника.

\* Включает:

\* - Настройку и хранение пути к выбранной музыке

\* - Запуск/остановку будильника

\* - Обработку нажатий кнопок инкремента/декремента

\* - Воспроизведение выбранной музыки (как тест, так и при срабатывании)

\* - Диалог при срабатывании ("Отключить" или "Отложить на 5 минут")

\* - Список запланированных будильников, с возможностью удаления

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

using System;

using System.Collections.Generic;

using System.IO;

using System.Windows.Forms;

using WMPLib;

namespace WinForms\_Alarm

{

public partial class MainForm : Form

{

/\*

Переменные-состояния будильника и музыки:

Music = 1 означает, что музыка играет.

Music = 0 означает, что музыка не играет.

\*/

private int Music = 0;

/\*

Часы, минуты, секунды, введённые пользователем

(то, на какое время будет установлен будильник)

\*/

private string Hour = "";

private string Minutes = "";

private string Seconds = "";

/\*

Текущее системное время (часы, минуты, секунды)

в формате строк

\*/

private string HourNow = "";

private string MinutesNow = "";

private string SecondsNow = "";

/\*

Объект Windows Media Player для воспроизведения mp3

\*/

private WindowsMediaPlayer wplayer = new WindowsMediaPlayer();

/\*

Список всех запланированных будильников.

\*/

private List<AlarmItem> alarms = new List<AlarmItem>();

/\*

Храним здесь "последнее обработанное время" (ЧЧММСС)

\*/

private string lastAlarmCheckTime = "";

/// <summary>

/// Конструктор формы.

/// </summary>

public MainForm()

{

InitializeComponent();

}

/// <summary>

/// Событие загрузки формы.

/// Здесь восстанавливаем путь к музыке из настроек

/// </summary>

private void MainForm\_Load(object sender, EventArgs e)

{

int h = DateTime.Now.Hour;

int m = DateTime.Now.Minute;

int s = DateTime.Now.Second;

string hh = h < 10 ? "0" + h : h.ToString();

string mm = m < 10 ? "0" + m : m.ToString();

string ss = s < 10 ? "0" + s : s.ToString();

textBoxHours.Text = hh;

textBoxMinutes.Text = mm;

textBoxSeconds.Text = ss;

string savedPath = Properties.Settings.Default.MusicPath;

if (!string.IsNullOrEmpty(savedPath) && File.Exists(savedPath))

{

UpdateMusicPath(savedPath);

}

else

{

labelCurrentMusic.Text = "Музыка не выбрана";

}

}

/// <summary>

/// Обновляет метку, показывающую путь к текущей музыке.

/// </summary>

private void UpdateMusicPath(string fullPath)

{

if (string.IsNullOrEmpty(fullPath))

{

labelCurrentMusic.Text = "Музыка не выбрана";

return;

}

string shortName = Path.GetFileName(fullPath);

if (shortName.Length > 28)

{

shortName = shortName.Substring(0, 28) + "...";

}

labelCurrentMusic.Text = $"Текущая музыка: {shortName}";

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Обработка инкремента/декремента для Часов, Минут, Секунд

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

{

int h = int.Parse(textBoxHours.Text);

h = (h + 1) % 24;

textBoxHours.Text = h < 10 ? "0" + h : h.ToString();

}

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

{

int h = int.Parse(textBoxHours.Text);

h = (h - 1) < 0 ? 23 : (h - 1);

textBoxHours.Text = h < 10 ? "0" + h : h.ToString();

}

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

{

int m = int.Parse(textBoxMinutes.Text);

m = (m + 1) % 60;

textBoxMinutes.Text = m < 10 ? "0" + m : m.ToString();

}

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

{

int m = int.Parse(textBoxMinutes.Text);

m = (m - 1) < 0 ? 59 : (m - 1);

textBoxMinutes.Text = m < 10 ? "0" + m : m.ToString();

}

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

{

int s = int.Parse(textBoxSeconds.Text);

s = (s + 1) % 60;

textBoxSeconds.Text = s < 10 ? "0" + s : s.ToString();

}

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

{

int s = int.Parse(textBoxSeconds.Text);

s = (s - 1) < 0 ? 59 : (s - 1);

textBoxSeconds.Text = s < 10 ? "0" + s : s.ToString();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Обработчики полей ввода (Ч, М, С)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

private void textBoxHours\_Leave(object sender, EventArgs e)

{

if (textBoxHours.Text.Length == 1)

{

textBoxHours.Text = "0" + textBoxHours.Text;

}

}

private void textBoxMinutes\_Leave(object sender, EventArgs e)

{

if (textBoxMinutes.Text.Length == 1)

{

textBoxMinutes.Text = "0" + textBoxMinutes.Text;

}

}

private void textBoxSeconds\_Leave(object sender, EventArgs e)

{

if (textBoxSeconds.Text.Length == 1)

{

textBoxSeconds.Text = "0" + textBoxSeconds.Text;

}

}

private void textBoxHours\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Back) return;

if (!char.IsDigit(e.KeyChar) || textBoxHours.Text.Length >= 2)

{

e.Handled = true;

}

}

private void textBoxMinutes\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Back) return;

if (!char.IsDigit(e.KeyChar) || textBoxMinutes.Text.Length >= 2)

{

e.Handled = true;

}

}

private void textBoxSeconds\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Back) return;

if (!char.IsDigit(e.KeyChar) || textBoxSeconds.Text.Length >= 2)

{

e.Handled = true;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Главное меню (Файл: Выбрать музыку, Прослушать, Выход;

\* Справка: О программе)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

{

if (Music == 1)

{

wplayer.controls.stop();

Music = 0;

}

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

string file = openFileDialog1.FileName;

string extension = Path.GetExtension(file);

if (!extension.Equals(".mp3", StringComparison.OrdinalIgnoreCase))

{

MessageBox.Show("Выберите mp3 файл!", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

Properties.Settings.Default.MusicPath = file;

Properties.Settings.Default.Save();

UpdateMusicPath(file);

MessageBox.Show("Музыка успешно выбрана и сохранена!",

"Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

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

{

string currentPath = Properties.Settings.Default.MusicPath;

if (string.IsNullOrEmpty(currentPath) || !File.Exists(currentPath))

{

MessageBox.Show("Музыка не выбрана или файл недоступен!",

"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (Music == 0)

{

wplayer.URL = currentPath;

wplayer.controls.play();

Music = 1;

previewMusicToolStripMenuItem.Text = "Остановить прослушивание";

}

else

{

wplayer.controls.stop();

Music = 0;

previewMusicToolStripMenuItem.Text = "Прослушать музыку";

}

}

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

{

Application.Exit();

}

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

{

MessageBox.Show(

"Будильник (WinForms)\n\n" +

"Версия: 1.0.0\n" +

"Разработчик: Ваше\_Имя\n\n" +

"Данный будильник позволяет:\n" +

"- Установить время срабатывания\n" +

"- Выбрать и сохранить MP3-файл\n" +

"- Прослушать выбранную мелодию\n" +

"- Запланировать несколько будильников\n" +

"- При срабатывании будильника: отключить или перенести на 5 минут\n" +

"- Работает даже после перезапуска, т.к. путь к музыке сохраняется",

"О программе",

MessageBoxButtons.OK,

MessageBoxIcon.Information

);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Логика добавления/удаления будильников

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

{

if (string.IsNullOrWhiteSpace(textBoxHours.Text) ||

string.IsNullOrWhiteSpace(textBoxMinutes.Text) ||

string.IsNullOrWhiteSpace(textBoxSeconds.Text))

{

MessageBox.Show("Укажите часы, минуты и секунды!", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

int h = int.Parse(textBoxHours.Text);

int m = int.Parse(textBoxMinutes.Text);

int s = int.Parse(textBoxSeconds.Text);

if (h > 23)

{

MessageBox.Show("Часы не могут превышать 23!", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (m > 59 || s > 59)

{

MessageBox.Show("Минуты и секунды не могут превышать 59!", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

var newAlarm = new AlarmItem(h, m, s);

alarms.Add(newAlarm);

listBoxAlarms.Items.Add(newAlarm);

MessageBox.Show("Будильник добавлен в список!", "Информация",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

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

{

if (listBoxAlarms.SelectedIndex >= 0)

{

int idx = listBoxAlarms.SelectedIndex;

alarms.RemoveAt(idx);

listBoxAlarms.Items.RemoveAt(idx);

MessageBox.Show("Будильник удалён!", "Информация",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Выберите будильник для удаления!", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Логика запуска/остановки будильника (кнопка "Запустить")

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

{

if (buttonStart.Text == "Стоп")

{

timer1.Stop();

textBoxHours.ReadOnly = false;

textBoxMinutes.ReadOnly = false;

textBoxSeconds.ReadOnly = false;

buttonStart.Text = "Запустить";

if (Music == 1)

{

wplayer.controls.stop();

Music = 0;

previewMusicToolStripMenuItem.Text = "Прослушать музыку";

}

return;

}

if (alarms.Count == 0)

{

MessageBox.Show("Список будильников пуст! Сначала добавьте будильник.", "Ошибка",

MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

textBoxHours.ReadOnly = true;

textBoxMinutes.ReadOnly = true;

textBoxSeconds.ReadOnly = true;

buttonStart.Text = "Стоп";

timer1.Start();

}

/// <summary>

/// Срабатывает каждые 500 мс, проверяет текущее время

/// и сравнивает с каждым будильником из списка.

/// Если время совпало, запускаем диалог.

/// </summary>

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

{

int h = DateTime.Now.Hour;

int m = DateTime.Now.Minute;

int s = DateTime.Now.Second;

HourNow = h < 10 ? "0" + h : h.ToString();

MinutesNow = m < 10 ? "0" + m : m.ToString();

SecondsNow = s < 10 ? "0" + s : s.ToString();

// Собираем текущее время в "ЧЧММСС"

string currentTime = HourNow + MinutesNow + SecondsNow;

// Если уже обрабатывали это время (за ту же секунду), выходим

if (currentTime == lastAlarmCheckTime)

{

return;

}

// Иначе запоминаем, что это время мы уже обрабатываем

lastAlarmCheckTime = currentTime;

// Далее перебираем все будильники

for (int i = 0; i < alarms.Count; i++)

{

var alarm = alarms[i];

string ah = alarm.Hour < 10 ? "0" + alarm.Hour : alarm.Hour.ToString();

string am = alarm.Minute < 10 ? "0" + alarm.Minute : alarm.Minute.ToString();

string asx = alarm.Second < 10 ? "0" + alarm.Second : alarm.Second.ToString();

if (ah == HourNow && am == MinutesNow && asx == SecondsNow)

{

if (Music == 0)

{

wplayer.URL = Properties.Settings.Default.MusicPath;

wplayer.controls.play();

Music = 1;

previewMusicToolStripMenuItem.Text = "Остановить прослушивание";

}

DialogResult dr = MessageBox.Show(

"Будильник сработал!\n\nДа - Отключить\nНет - Отложить на 5 минут",

"Сработка будильника",

MessageBoxButtons.YesNo,

MessageBoxIcon.Question

);

wplayer.controls.stop();

Music = 0;

previewMusicToolStripMenuItem.Text = "Прослушать музыку";

if (dr == DialogResult.Yes)

{

alarms.RemoveAt(i);

listBoxAlarms.Items.RemoveAt(i);

i--;

}

else

{

DateTime dt = DateTime.Now.AddMinutes(5);

alarm.Hour = dt.Hour;

alarm.Minute = dt.Minute;

alarm.Second = dt.Second;

listBoxAlarms.Items[i] = alarm;

}

break;

}

}

}

}

}

Листинг 2 – Код файла MainForm.Designer.cs

namespace WinForms\_Alarm

{

partial class MainForm

{

private System.ComponentModel.IContainer components = null;

private System.Windows.Forms.MenuStrip menuStrip1;

private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem chooseMusicToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem previewMusicToolStripMenuItem;

private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;

private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem helpToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem aboutToolStripMenuItem;

private System.Windows.Forms.Label labelTitle;

private System.Windows.Forms.Label labelSignal;

private System.Windows.Forms.Label labelCurrentMusic;

private System.Windows.Forms.Label labelHours;

private System.Windows.Forms.Label labelMinutes;

private System.Windows.Forms.Label labelSeconds;

private System.Windows.Forms.TextBox textBoxHours;

private System.Windows.Forms.TextBox textBoxMinutes;

private System.Windows.Forms.TextBox textBoxSeconds;

private System.Windows.Forms.Button btnIncrementHours;

private System.Windows.Forms.Button btnDecrementHours;

private System.Windows.Forms.Button btnIncrementMinutes;

private System.Windows.Forms.Button btnDecrementMinutes;

private System.Windows.Forms.Button btnIncrementSeconds;

private System.Windows.Forms.Button btnDecrementSeconds;

private System.Windows.Forms.Button buttonAddAlarm;

private System.Windows.Forms.Button buttonRemoveAlarm;

private System.Windows.Forms.ListBox listBoxAlarms;

private System.Windows.Forms.Button buttonStart;

private System.Windows.Forms.Timer timer1;

private System.Windows.Forms.OpenFileDialog openFileDialog1;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

/// <summary>

/// Метод, автоматически сгенерированный конструктором форм.

/// </summary>

private void InitializeComponent()

{

this.components = new System.ComponentModel.Container();

this.menuStrip1 = new System.Windows.Forms.MenuStrip();

this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.chooseMusicToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.previewMusicToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.toolStripSeparator1 = new System.Windows.Forms.ToolStripSeparator();

this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.helpToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.aboutToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.labelTitle = new System.Windows.Forms.Label();

this.labelSignal = new System.Windows.Forms.Label();

this.labelCurrentMusic = new System.Windows.Forms.Label();

this.labelHours = new System.Windows.Forms.Label();

this.labelMinutes = new System.Windows.Forms.Label();

this.labelSeconds = new System.Windows.Forms.Label();

this.textBoxHours = new System.Windows.Forms.TextBox();

this.textBoxMinutes = new System.Windows.Forms.TextBox();

this.textBoxSeconds = new System.Windows.Forms.TextBox();

this.btnIncrementHours = new System.Windows.Forms.Button();

this.btnDecrementHours = new System.Windows.Forms.Button();

this.btnIncrementMinutes = new System.Windows.Forms.Button();

this.btnDecrementMinutes = new System.Windows.Forms.Button();

this.btnIncrementSeconds = new System.Windows.Forms.Button();

this.btnDecrementSeconds = new System.Windows.Forms.Button();

this.buttonAddAlarm = new System.Windows.Forms.Button();

this.buttonRemoveAlarm = new System.Windows.Forms.Button();

this.listBoxAlarms = new System.Windows.Forms.ListBox();

this.buttonStart = new System.Windows.Forms.Button();

this.timer1 = new System.Windows.Forms.Timer(this.components);

this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();

this.menuStrip1.SuspendLayout();

this.SuspendLayout();

//

// menuStrip1

//

this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.fileToolStripMenuItem,

this.helpToolStripMenuItem});

this.menuStrip1.Location = new System.Drawing.Point(0, 0);

this.menuStrip1.Name = "menuStrip1";

this.menuStrip1.Padding = new System.Windows.Forms.Padding(5, 2, 0, 2);

this.menuStrip1.Size = new System.Drawing.Size(584, 24);

this.menuStrip1.TabIndex = 0;

//

// fileToolStripMenuItem

//

this.fileToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.chooseMusicToolStripMenuItem,

this.previewMusicToolStripMenuItem,

this.toolStripSeparator1,

this.exitToolStripMenuItem});

this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";

this.fileToolStripMenuItem.Size = new System.Drawing.Size(48, 20);

this.fileToolStripMenuItem.Text = "Файл";

//

// chooseMusicToolStripMenuItem

//

this.chooseMusicToolStripMenuItem.Name = "chooseMusicToolStripMenuItem";

this.chooseMusicToolStripMenuItem.Size = new System.Drawing.Size(188, 22);

this.chooseMusicToolStripMenuItem.Text = "Выбрать музыку";

this.chooseMusicToolStripMenuItem.Click += new System.EventHandler(this.chooseMusicToolStripMenuItem\_Click);

//

// previewMusicToolStripMenuItem

//

this.previewMusicToolStripMenuItem.Name = "previewMusicToolStripMenuItem";

this.previewMusicToolStripMenuItem.Size = new System.Drawing.Size(188, 22);

this.previewMusicToolStripMenuItem.Text = "Прослушать музыку";

this.previewMusicToolStripMenuItem.Click += new System.EventHandler(this.previewMusicToolStripMenuItem\_Click);

//

// toolStripSeparator1

//

this.toolStripSeparator1.Name = "toolStripSeparator1";

this.toolStripSeparator1.Size = new System.Drawing.Size(185, 6);

//

// exitToolStripMenuItem

//

this.exitToolStripMenuItem.Name = "exitToolStripMenuItem";

this.exitToolStripMenuItem.Size = new System.Drawing.Size(188, 22);

this.exitToolStripMenuItem.Text = "Выход";

this.exitToolStripMenuItem.Click += new System.EventHandler(this.exitToolStripMenuItem\_Click);

//

// helpToolStripMenuItem

//

this.helpToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.aboutToolStripMenuItem});

this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";

this.helpToolStripMenuItem.Size = new System.Drawing.Size(65, 20);

this.helpToolStripMenuItem.Text = "Справка";

//

// aboutToolStripMenuItem

//

this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";

this.aboutToolStripMenuItem.Size = new System.Drawing.Size(149, 22);

this.aboutToolStripMenuItem.Text = "О программе";

this.aboutToolStripMenuItem.Click += new System.EventHandler(this.aboutToolStripMenuItem\_Click);

//

// labelTitle

//

this.labelTitle.AutoSize = true;

this.labelTitle.Font = new System.Drawing.Font("Segoe UI", 22F, System.Drawing.FontStyle.Bold);

this.labelTitle.ForeColor = System.Drawing.Color.MediumBlue;

this.labelTitle.Location = new System.Drawing.Point(188, 37);

this.labelTitle.Name = "labelTitle";

this.labelTitle.Size = new System.Drawing.Size(207, 41);

this.labelTitle.TabIndex = 1;

this.labelTitle.Text = "БУДИЛЬНИК";

//

// labelSignal

//

this.labelSignal.AutoSize = true;

this.labelSignal.Font = new System.Drawing.Font("Segoe UI", 14F);

this.labelSignal.ForeColor = System.Drawing.Color.DarkSlateGray;

this.labelSignal.Location = new System.Drawing.Point(166, 78);

this.labelSignal.Name = "labelSignal";

this.labelSignal.Size = new System.Drawing.Size(248, 25);

this.labelSignal.TabIndex = 2;

this.labelSignal.Text = "Сигнал сработает в (Ч:М:С)";

//

// labelCurrentMusic

//

this.labelCurrentMusic.AutoSize = true;

this.labelCurrentMusic.Font = new System.Drawing.Font("Segoe UI", 10F);

this.labelCurrentMusic.ForeColor = System.Drawing.Color.DarkOliveGreen;

this.labelCurrentMusic.Location = new System.Drawing.Point(12, 463);

this.labelCurrentMusic.Name = "labelCurrentMusic";

this.labelCurrentMusic.Size = new System.Drawing.Size(137, 19);

this.labelCurrentMusic.TabIndex = 3;

this.labelCurrentMusic.Text = "Музыка не выбрана";

//

// labelHours

//

this.labelHours.AutoSize = true;

this.labelHours.Font = new System.Drawing.Font("Segoe UI", 14F);

this.labelHours.Location = new System.Drawing.Point(179, 143);

this.labelHours.Name = "labelHours";

this.labelHours.Size = new System.Drawing.Size(25, 25);

this.labelHours.TabIndex = 7;

this.labelHours.Text = "Ч";

//

// labelMinutes

//

this.labelMinutes.AutoSize = true;

this.labelMinutes.Font = new System.Drawing.Font("Segoe UI", 14F);

this.labelMinutes.Location = new System.Drawing.Point(326, 143);

this.labelMinutes.Name = "labelMinutes";

this.labelMinutes.Size = new System.Drawing.Size(29, 25);

this.labelMinutes.TabIndex = 8;

this.labelMinutes.Text = "М";

//

// labelSeconds

//

this.labelSeconds.AutoSize = true;

this.labelSeconds.Font = new System.Drawing.Font("Segoe UI", 14F);

this.labelSeconds.Location = new System.Drawing.Point(473, 143);

this.labelSeconds.Name = "labelSeconds";

this.labelSeconds.Size = new System.Drawing.Size(24, 25);

this.labelSeconds.TabIndex = 9;

this.labelSeconds.Text = "С";

//

// textBoxHours

//

this.textBoxHours.Font = new System.Drawing.Font("Segoe UI", 18F);

this.textBoxHours.Location = new System.Drawing.Point(101, 139);

this.textBoxHours.MaxLength = 2;

this.textBoxHours.Name = "textBoxHours";

this.textBoxHours.Size = new System.Drawing.Size(73, 39);

this.textBoxHours.TabIndex = 4;

this.textBoxHours.TextAlign = System.Windows.Forms.HorizontalAlignment.Center;

this.textBoxHours.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBoxHours\_KeyPress);

this.textBoxHours.Leave += new System.EventHandler(this.textBoxHours\_Leave);

//

// textBoxMinutes

//

this.textBoxMinutes.Font = new System.Drawing.Font("Segoe UI", 18F);

this.textBoxMinutes.Location = new System.Drawing.Point(249, 139);

this.textBoxMinutes.MaxLength = 2;

this.textBoxMinutes.Name = "textBoxMinutes";

this.textBoxMinutes.Size = new System.Drawing.Size(73, 39);

this.textBoxMinutes.TabIndex = 5;

this.textBoxMinutes.TextAlign = System.Windows.Forms.HorizontalAlignment.Center;

this.textBoxMinutes.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBoxMinutes\_KeyPress);

this.textBoxMinutes.Leave += new System.EventHandler(this.textBoxMinutes\_Leave);

//

// textBoxSeconds

//

this.textBoxSeconds.Font = new System.Drawing.Font("Segoe UI", 18F);

this.textBoxSeconds.Location = new System.Drawing.Point(396, 139);

this.textBoxSeconds.MaxLength = 2;

this.textBoxSeconds.Name = "textBoxSeconds";

this.textBoxSeconds.Size = new System.Drawing.Size(73, 39);

this.textBoxSeconds.TabIndex = 6;

this.textBoxSeconds.TextAlign = System.Windows.Forms.HorizontalAlignment.Center;

this.textBoxSeconds.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBoxSeconds\_KeyPress);

this.textBoxSeconds.Leave += new System.EventHandler(this.textBoxSeconds\_Leave);

//

// btnIncrementHours

//

this.btnIncrementHours.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.btnIncrementHours.Font = new System.Drawing.Font("Segoe UI", 10F, System.Drawing.FontStyle.Bold);

this.btnIncrementHours.Location = new System.Drawing.Point(101, 108);

this.btnIncrementHours.Name = "btnIncrementHours";

this.btnIncrementHours.Size = new System.Drawing.Size(73, 26);

this.btnIncrementHours.TabIndex = 10;

this.btnIncrementHours.Text = "▲";

this.btnIncrementHours.UseVisualStyleBackColor = true;

this.btnIncrementHours.Click += new System.EventHandler(this.btnIncrementHours\_Click);

//

// btnDecrementHours

//

this.btnDecrementHours.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.btnDecrementHours.Font = new System.Drawing.Font("Segoe UI", 10F, System.Drawing.FontStyle.Bold);

this.btnDecrementHours.Location = new System.Drawing.Point(101, 178);

this.btnDecrementHours.Name = "btnDecrementHours";

this.btnDecrementHours.Size = new System.Drawing.Size(73, 26);

this.btnDecrementHours.TabIndex = 11;

this.btnDecrementHours.Text = "▼";

this.btnDecrementHours.UseVisualStyleBackColor = true;

this.btnDecrementHours.Click += new System.EventHandler(this.btnDecrementHours\_Click);

//

// btnIncrementMinutes

//

this.btnIncrementMinutes.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.btnIncrementMinutes.Font = new System.Drawing.Font("Segoe UI", 10F, System.Drawing.FontStyle.Bold);

this.btnIncrementMinutes.Location = new System.Drawing.Point(249, 108);

this.btnIncrementMinutes.Name = "btnIncrementMinutes";

this.btnIncrementMinutes.Size = new System.Drawing.Size(73, 26);

this.btnIncrementMinutes.TabIndex = 12;

this.btnIncrementMinutes.Text = "▲";

this.btnIncrementMinutes.UseVisualStyleBackColor = true;

this.btnIncrementMinutes.Click += new System.EventHandler(this.btnIncrementMinutes\_Click);

//

// btnDecrementMinutes

//

this.btnDecrementMinutes.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.btnDecrementMinutes.Font = new System.Drawing.Font("Segoe UI", 10F, System.Drawing.FontStyle.Bold);

this.btnDecrementMinutes.Location = new System.Drawing.Point(249, 178);

this.btnDecrementMinutes.Name = "btnDecrementMinutes";

this.btnDecrementMinutes.Size = new System.Drawing.Size(73, 26);

this.btnDecrementMinutes.TabIndex = 13;

this.btnDecrementMinutes.Text = "▼";

this.btnDecrementMinutes.UseVisualStyleBackColor = true;

this.btnDecrementMinutes.Click += new System.EventHandler(this.btnDecrementMinutes\_Click);

//

// btnIncrementSeconds

//

this.btnIncrementSeconds.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.btnIncrementSeconds.Font = new System.Drawing.Font("Segoe UI", 10F, System.Drawing.FontStyle.Bold);

this.btnIncrementSeconds.Location = new System.Drawing.Point(396, 108);

this.btnIncrementSeconds.Name = "btnIncrementSeconds";

this.btnIncrementSeconds.Size = new System.Drawing.Size(73, 26);

this.btnIncrementSeconds.TabIndex = 14;

this.btnIncrementSeconds.Text = "▲";

this.btnIncrementSeconds.UseVisualStyleBackColor = true;

this.btnIncrementSeconds.Click += new System.EventHandler(this.btnIncrementSeconds\_Click);

//

// btnDecrementSeconds

//

this.btnDecrementSeconds.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.btnDecrementSeconds.Font = new System.Drawing.Font("Segoe UI", 10F, System.Drawing.FontStyle.Bold);

this.btnDecrementSeconds.Location = new System.Drawing.Point(396, 178);

this.btnDecrementSeconds.Name = "btnDecrementSeconds";

this.btnDecrementSeconds.Size = new System.Drawing.Size(73, 26);

this.btnDecrementSeconds.TabIndex = 15;

this.btnDecrementSeconds.Text = "▼";

this.btnDecrementSeconds.UseVisualStyleBackColor = true;

this.btnDecrementSeconds.Click += new System.EventHandler(this.btnDecrementSeconds\_Click);

//

// buttonAddAlarm

//

this.buttonAddAlarm.BackColor = System.Drawing.Color.Honeydew;

this.buttonAddAlarm.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.buttonAddAlarm.Font = new System.Drawing.Font("Segoe UI", 11F, System.Drawing.FontStyle.Bold);

this.buttonAddAlarm.Location = new System.Drawing.Point(171, 243);

this.buttonAddAlarm.Name = "buttonAddAlarm";

this.buttonAddAlarm.Size = new System.Drawing.Size(103, 39);

this.buttonAddAlarm.TabIndex = 17;

this.buttonAddAlarm.Text = "Добавить";

this.buttonAddAlarm.UseVisualStyleBackColor = false;

this.buttonAddAlarm.Click += new System.EventHandler(this.buttonAddAlarm\_Click);

//

// buttonRemoveAlarm

//

this.buttonRemoveAlarm.BackColor = System.Drawing.Color.MistyRose;

this.buttonRemoveAlarm.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.buttonRemoveAlarm.Font = new System.Drawing.Font("Segoe UI", 11F, System.Drawing.FontStyle.Bold);

this.buttonRemoveAlarm.Location = new System.Drawing.Point(300, 243);

this.buttonRemoveAlarm.Name = "buttonRemoveAlarm";

this.buttonRemoveAlarm.Size = new System.Drawing.Size(103, 39);

this.buttonRemoveAlarm.TabIndex = 18;

this.buttonRemoveAlarm.Text = "Удалить";

this.buttonRemoveAlarm.UseVisualStyleBackColor = false;

this.buttonRemoveAlarm.Click += new System.EventHandler(this.buttonRemoveAlarm\_Click);

//

// listBoxAlarms

//

this.listBoxAlarms.Font = new System.Drawing.Font("Segoe UI", 12F);

this.listBoxAlarms.FormattingEnabled = true;

this.listBoxAlarms.ItemHeight = 21;

this.listBoxAlarms.Location = new System.Drawing.Point(78, 299);

this.listBoxAlarms.Name = "listBoxAlarms";

this.listBoxAlarms.Size = new System.Drawing.Size(419, 88);

this.listBoxAlarms.TabIndex = 19;

//

// buttonStart

//

this.buttonStart.BackColor = System.Drawing.Color.LightGray;

this.buttonStart.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

this.buttonStart.Font = new System.Drawing.Font("Segoe UI", 14F);

this.buttonStart.Location = new System.Drawing.Point(225, 406);

this.buttonStart.Name = "buttonStart";

this.buttonStart.Size = new System.Drawing.Size(141, 43);

this.buttonStart.TabIndex = 16;

this.buttonStart.Text = "Запустить";

this.buttonStart.UseVisualStyleBackColor = false;

this.buttonStart.Click += new System.EventHandler(this.buttonStart\_Click);

//

// timer1

//

this.timer1.Interval = 500;

this.timer1.Tick += new System.EventHandler(this.timer1\_Tick);

//

// openFileDialog1

//

this.openFileDialog1.FileName = "openFileDialog1";

this.openFileDialog1.Filter = "MP3 Files|\*.mp3|All Files|\*.\*";

//

// MainForm

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.AliceBlue;

this.ClientSize = new System.Drawing.Size(584, 491);

this.Controls.Add(this.menuStrip1);

this.Controls.Add(this.labelTitle);

this.Controls.Add(this.labelSignal);

this.Controls.Add(this.labelCurrentMusic);

this.Controls.Add(this.textBoxHours);

this.Controls.Add(this.textBoxMinutes);

this.Controls.Add(this.textBoxSeconds);

this.Controls.Add(this.labelHours);

this.Controls.Add(this.labelMinutes);

this.Controls.Add(this.labelSeconds);

this.Controls.Add(this.btnIncrementHours);

this.Controls.Add(this.btnDecrementHours);

this.Controls.Add(this.btnIncrementMinutes);

this.Controls.Add(this.btnDecrementMinutes);

this.Controls.Add(this.btnIncrementSeconds);

this.Controls.Add(this.btnDecrementSeconds);

this.Controls.Add(this.buttonStart);

this.Controls.Add(this.buttonAddAlarm);

this.Controls.Add(this.buttonRemoveAlarm);

this.Controls.Add(this.listBoxAlarms);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;

this.MaximizeBox = false;

this.Name = "MainForm";

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

this.Text = "Будильник";

this.Load += new System.EventHandler(this.MainForm\_Load);

this.menuStrip1.ResumeLayout(false);

this.menuStrip1.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

}

}

Листинг 3 – Код файла AlarmItem.cs

namespace WinForms\_Alarm

{

/// <summary>

/// Класс, описывающий один будильник (часы, минуты, секунды).

/// </summary>

public class AlarmItem

{

public int Hour;

public int Minute;

public int Second;

/// <summary>

/// Конструктор с заданием времени будильника.

/// </summary>

public AlarmItem(int h, int m, int s)

{

Hour = h;

Minute = m;

Second = s;

}

/// <summary>

/// Отображать время будильника в формате ЧЧ:ММ:СС.

/// </summary>

public override string ToString()

{

return $"{Hour:00}:{Minute:00}:{Second:00}";

}

}

}

*2. Описание программы (логическая структура и функционирование)*

Программа «Будильник» имеет классическую архитектуру WinForms-приложения, где основной класс MainForm отвечает за события пользовательского интерфейса. При запуске приложения выполняются следующие ключевые действия:

* Инициализация формы: подгружаются элементы управления (текстовые поля для ввода часов, минут, секунд, кнопки добавления, удаления, запуска будильника и пр.).
* Считывание настроек пользователя (если ранее была выбрана музыка, её путь восстанавливается).
* Подписка на события:
  1. Клик по кнопкам инкремента и декремента корректирует значения часов, минут, секунд.
  2. Клик по кнопке «Запустить» активирует таймер, регулярно (каждые полсекунды) проверяющий текущее время на совпадение с установленными будильниками.
  3. При совпадении времени программа воспроизводит заранее выбранный mp3-файл и выводит диалоговое окно, позволяющее «Отключить» будильник либо «Отложить на 5 минут».
* Вся логика проверки будильников и воспроизведения звука содержится в обработчике «timer1\_Tick». Если системное время совпадает с одним из будильников, музыка запускается, а пользователю выводится диалог с выбором дальнейших действий.

Логическая структура состоит из основных блоков:

* Интерфейс пользователя (WinForms).
* Бизнес-логика (сравнение времени, запуск/остановка музыкального проигрывателя).
* Данные (список будильников – объектов класса AlarmItem, сохранённый путь к звуковому файлу).

На рисунке 1 приведена блок-схема, описывающей общий алгоритм работы программы «Будильник».

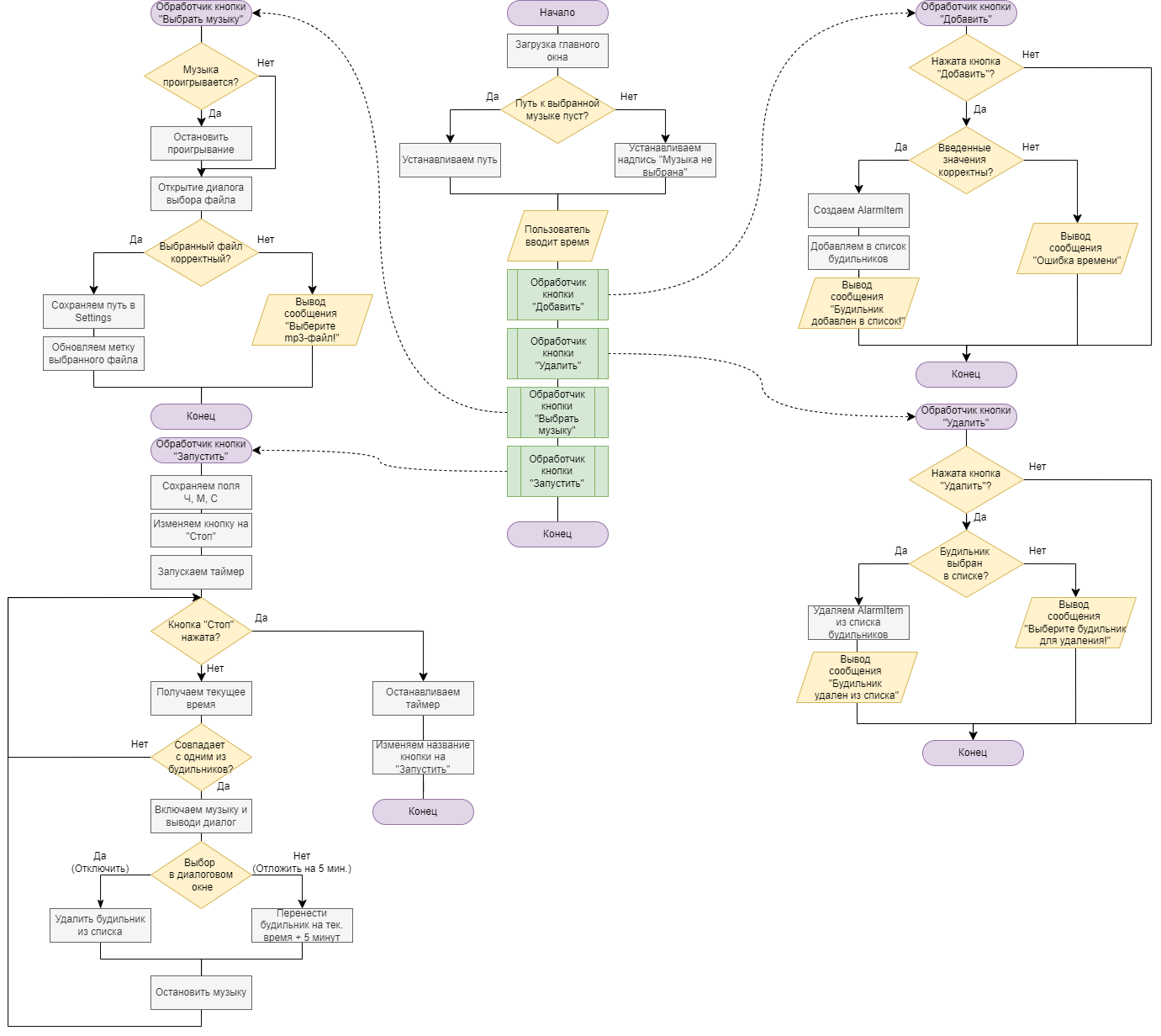


Рисунок 1 – Блок-схема работы программы «Будильник»

*3. Общее описание функционирования, обоснование решений*

Программа работает по следующему алгоритму:

1. При запуске программы отображается главная форма, куда автоматически подставляется текущее системное время. Если у пользователя ранее был выбран звуковой файл, его путь восстанавливается из настроек.
2. Пользователь вводит время (часы, минуты, секунды) и нажимает «Добавить», чтобы зафиксировать новый будильник. В списке будильников появляется новая запись.
3. По кнопке «Запустить» включается таймер с интервалом 500 мс. На каждом тике таймера программа получает текущее системное время.
4. Если текущее время совпадает с каким-либо будильником (из списка), будильник срабатывает, воспроизводя выбранный mp3-файл.
5. Пользователь видит диалоговое окно с предложением «Отключить» или «Отложить». При «Отключить» будильник удаляется из списка; при «Отложить» – его значение переводится на текущее время плюс 5 минут. Музыка отключается до следующего срабатывания.
6. Пользователь может остановить все будильники кнопкой «Стоп», что остановит таймер и даст возможность редактировать время в полях.

Обоснование некоторых принятых решений:

* Использование WinForms выбрано ради простоты и наглядности создания десктопных приложений под ОС Windows.
* Таймер с интервалом 500 мс позволяет реагировать на совпадение времени с точностью в секунду и гарантирует, что программа не «пропустит» момент срабатывания.
* Для воспроизведения музыки используется штатная библиотека Windows Media Player (wmp.dll), поскольку она легко интегрируется в WinForms, не требует дополнительных установок и поддерживает формат mp3.

*4. Эксплуатационные документы*

*4.1. Описание применения*

Программа «Будильник» предназначена для пользователей, которым требуется автоматический звуковой сигнал в определённые моменты времени.

Основные характеристики:

* Приложение функционирует под управлением ОС Windows (версии 7 и выше).
* Возможность выбора пользовательского mp3-файла для сигнала.
* Множество будильников: каждый будильник сохраняет время и, при желании, автоматически переносится на 5 минут по команде «Отложить».
* Ограничения по применению: требуется установленный .NET Framework и Windows Media Player.

Условия применения включают: наличие звуковых устройств для воспроизведения (наушники, колонки) и поддержка операционной системы Windows. Входные данные – время срабатывания и путь к mp3-файлу, выходные данные – воспроизведение выбранного сигнала по расписанию.

*4.2. Руководство системного программиста*

Для проверки и обеспечения функционирования требуется убедиться, что:

* На рабочей машине установлены .NET Framework и Windows Media Player.
* Пользователь имеет права на чтение/воспроизведение звуковых файлов.
* Если планируется развёртывание на нескольких компьютерах, возможно копирование исполняемого файла вместе с дополнительными библиотеками (либо установка через привычные способы распространения .NET-приложений).

*4.3. Руководство программиста*

Программа не требует существенного сопровождения со стороны программиста, однако при необходимости:

* Исходный код можно доработать, например, расширяя функционал «отложить» или добавляя другие интервалы.
* Допускается интеграция с иными библиотеками воспроизведения звука.
* Формат входных и выходных данных стандартный: время (часы, минуты, секунды) в виде строк, mp3-файл в локальной файловой системе.
* При возникновении ошибок с чтением файла выводится соответствующее сообщение, программисту нужно проверить корректность пути или целостность файла.

*4.4. Руководство оператора*

Для использования программы оператор:

1. Запускает приложение «Будильник» (двойной клик по ярлыку/файлу .exe).
2. Выбирает при необходимости звуковой файл в меню «Файл → Выбрать музыку».
3. В полях «Ч», «М», «С» указывает время и нажимает «Добавить», формируя список сигналов.
4. Кнопка «Запустить» активирует будильник. При наступлении заданного времени программа выдаёт звуковой сигнал и диалоговое окно:

* «Отключить» – останавливает звук и удаляет будильник из списка.
* «Отложить» – останавливает звук и переносит будильник на +5 минут.
* Сообщения при ошибках (например, отсутствие mp3-файла) отображаются в виде всплывающих окон с пояснением проблемы и дальнейшими действиями (например, «Выберите mp3-файл!»).

*4.5. Описание языка*

Программа написана на языке C# (синтаксис и семантика языка подробно описаны в официальной документации Microsoft). Документ «Описание языка» не прилагается, так как продукт не реализует собственный язык программирования, а использует существующий язык C#.

*4.6. Руководство по техническому обслуживанию*

Дополнительных программ тестирования и диагностики данное приложение не требует. Оно работает поверх платформы Windows Forms, зависимостей помимо Windows Media Player нет. При появлении неисправностей (например, если музыка не воспроизводится) рекомендуется:

* Проверить установку Windows Media Player;
* Убедиться в корректности звуковой карты и драйверов;
* Убедиться в праве доступа к .NET Framework и mp3-файлам.

*5. Тестирование программы*

Ниже приведена примерная таблица ручного тестирования программы «Будильник» с перечислением основных функций и возможных сценариев проверки.

Таблица 1 – Таблица результатов ручного тестирования программы

|  |  |  |  |
| --- | --- | --- | --- |
| **Действие** | **Входные данные** | **Ожидаемый результат** | **Фактический результат** |
| Запуск программы и автоматическое заполнение полей текущим временем | Запуск «Будильника» при системном времени, например 10:05:12 | Текстовые поля (Ч, М, С) содержат «10», «05», «12». | Текстовые поля показывают актуальное системное время. |
| Проверка ввода корректного времени в поля «Ч», «М», «С» | Ввести вручную «11» «20» «00» | Поля принимают значения; если в поле 1 символ, автоматически добавляется «0» перед числом. | Введённое время отображается корректно (например, «11:20:00»). |
| Добавление будильника в список | Нажать «Добавить» при заполненных полях Ч=11, М=20, С=00 | В списке будильников появляется запись «11:20:00»; отображается сообщение «Будильник добавлен в список!» | Будильник появился в списке, сообщение вывелось. |
| Удаление будильника из списка | Выбрать созданный будильник и нажать «Удалить» | Будильник исчезает из списка, выводится сообщение «Будильник удалён!». | Будильник удалён успешно, сообщение появилось. |
| Проверка выбора mp3-файла (кнопка «Выбрать музыку») | Выбрать корректный mp3 в диалоговом окне (например, song.mp3) | Путь к файлу сохраняется, отображается имя музыки в виде «Текущая музыка: song.mp3…». | Путь сохранился, на метке формы появилось корректное имя файла. |
| Неправильный выбор музыки (не mp3-файл) | При выборе файла .wav или .exe | Появляется сообщение «Выберите mp3 файл!», выбор отменяется. | Сообщение «Выберите mp3 файл!» появилось, музыка не установилась. |
| Прослушивание музыки из меню «Прослушать музыку» | Наличие корректного mp3-файла в настройках | Музыка начинает воспроизводиться, текст пункта меню меняется на «Остановить прослушивание». | Музыка воспроизводится, меню успешно меняет название. |
| Остановка прослушивания музыки из того же пункта меню | Повторный клик по «Остановить прослушивание» | Музыка останавливается, пункт меню меняется обратно на «Прослушать музыку». | Воспроизведение прекращено, название пункта меню восстановилось. |
| Запуск будильника (кнопка «Запустить») при наличии хотя бы одного будильника в списке | Список содержит «11:20:00» и выбран mp3-файл | Поля ввода становятся «read-only», запускается таймер, кнопка меняет текст на «Стоп». | Поля Ч, М, С недоступны для редактирования, будильник «ждёт» срабатывания. |
| Срабатывание будильника в 11:20:00 | Системное время доходит до «11:20:00» | Выводится диалог: «Будильник сработал! Да - Отключить, Нет - Отложить на 5 минут», музыка начинает играть. | Диалоговое окно появилось в 11:20:00, музыка зазвучала. |
| Выбор «Отключить» в диалоговом окне при срабатывании | Клик по кнопке «Да» | Музыка останавливается, выбранный будильник удаляется из списка, окно диалога закрывается. | Музыка остановлена, будильник удалён. |
| Выбор «Отложить» на 5 минут в диалоговом окне | Клик по кнопке «Нет» | Музыка останавливается, будильник переносится на текущее время + 5 минут (отображается в списке), окно диалога закрывается. | Будильник обновил своё время, музыка остановлена. |
| Ситуация, когда в списке будильников нет записей и пользователь жмёт «Запустить» | Список будильников пуст | Появляется сообщение «Список будильников пуст! Сначала добавьте будильник.» | Сообщение появляется, будильник не запускается. |
| Нажатие «Стоп» при активном будильнике | Клик по кнопке «Стоп» | Таймер останавливается, поля ввода Ч, М, С снова доступны, музыка (если звучала) останавливается, кнопка меняет текст обратно на «Запустить». | Будильник завершён, поля доступны, музыка остановлена, кнопка переименована обратно. |
| Выход из программы через меню «Файл → Выход» | Клик по пункту «Выход» | Программа закрывается корректно, при следующем запуске восстанавливается путь к последнему выбранному mp3 (если он не был удалён). | Программа завершила работу, настройки пути к музыке успешно сохранились. |