

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ (МИИТ)**

**ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**

**Лабораторная работа №4**

**по дисциплине** «**Методы программирования**»

«Работа с календарём»

**Выполнил:** ст. гр. ТКИ-342

Речкалов Р.Ю.

Ярий А.А.

**Проверил:** к.т.н. Сафронов А.И.

**Москва – 2024 г**

# **1. Цель работы**

Освоение навыков настройки элемента «Календарь на месяц» (*MonthCalendar*).

# **2. Формулировка задачи**

В интегрированной среде разработки *Microsoft Visual Studio* на языке *Visual C#* разработать программу в режиме *Windows Forms Application*, представляющую собой средство взаимодействия с текстовым файлом, необходимым для отображения на календаре найденных в нём дат (в зависимости от варианта), а также справочной информации.

Предусмотреть комбинированный список, в который будут записаны те же даты из файла, что и в сам календарь. Даты в списке должны быть упорядочены и выбор каждой даты из списка переводит собственный курсор календаря на такую же дату.

На форме должны быть размещены два календаря вне зависимости от того, будут ли оба использованы для реализации основного и индивидуального заданий. В случае, когда календарь не используется, необходимо заблокировать его доступность.

Предусмотреть по нажатии сочетания клавиш «*Ctrl+H*» вызов справки о работе с программой из файла в статический по размеру ярлык

 На правом (или нижнем) календаре выполнить выделение всех выходных дней указанного календарного года без учёта праздничных дней. Сведения о выходных днях допустимо хранить в текстовых файлах.

# **3. Составление диаграммы классов, входящих в состав решения.**

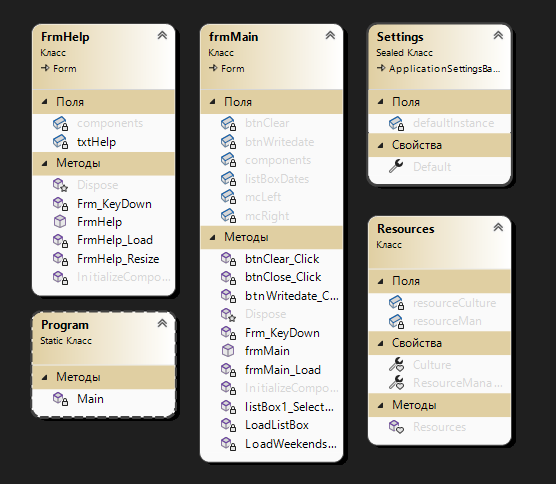


Рисунок 1 – Диаграмма классов

# **4. Составление сети Петри запрограммированного технологического процесса.**

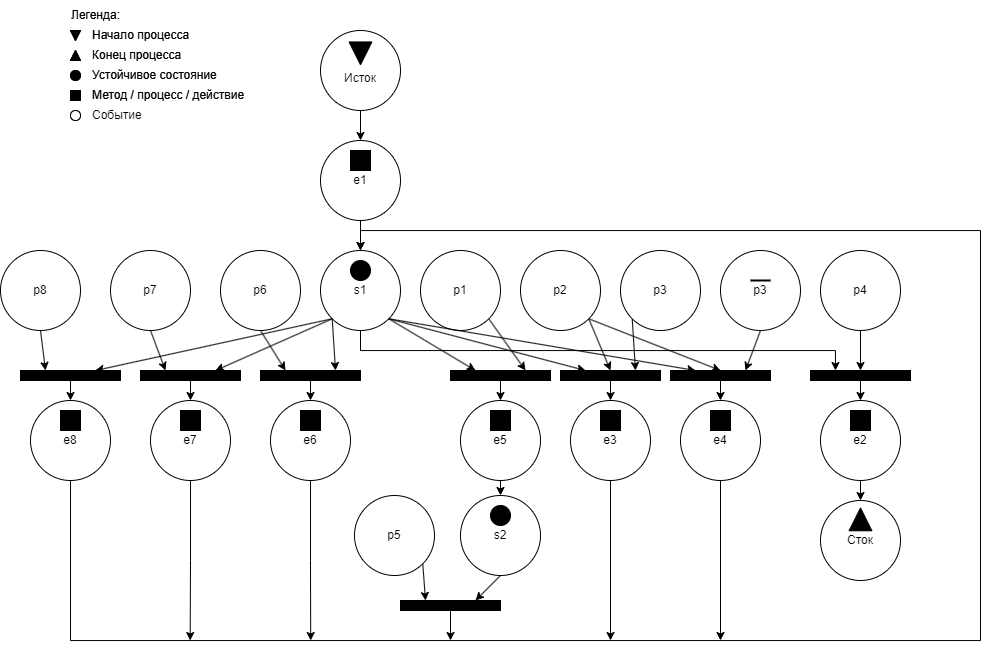


Рисунок 2 – Сеть Петри основной формы

**Описание сети Петри**

- состояния (states)

s1 – форма ожидает действий пользователя

s2 – форма свернута

- действия (effects)

e1 – из файлов загружается необходимая информация;

на правом календаре выделяются выходные дни

e2 – закрытие формы

e3 – форма максимизируется

e4 – форма переходит в нормальное состояние

e5 – форма сворачивается

e6 – запись даты в файл

e7 – очистка файла с датами

e8 – высвечивается информация о выбранном элементе

- события (prompts)

p1 – нажата кнопка «свернуть в трей»

p2 – нажата кнопка «максимизировать/минимизировать»

p3 – форма находится в нормальном состоянии

p4 – нажата кнопка закрытия

p5 – пользователь разворачивает форму

p6 – нажата кнопка записи в файл

p7 – нажата кнопка очистки файла

p8 – пользователь удерживает курсор на одном из элементов

# **5. Составление схем алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка ).**



Рисунок 3 – Схема алгоритма записи информации в файл

# **6. Подбор тестовых примеров.**

Тестовые примеры:

Проверка кнопки «максимизировать/минимизировать»

Проверка записи в файл

Проверка вывода справки комбинацией клавиш ctrl+h

# **7. Листинг (код) составленного программного обеспечения.**

Основная форма:

using System;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace Task4

{

public partial class frmMain : Form

{

public frmMain()

{

InitializeComponent();

LoadListBox();

LoadWeekendsOnly();

this.KeyDown += new KeyEventHandler(Frm\_KeyDown);

this.KeyPreview = true;

}

private void Frm\_KeyDown(object sender, KeyEventArgs e)

{

if (e.Control && e.KeyCode == Keys.H)

{

FrmHelp fr2 = new FrmHelp();

fr2.Show();

Hide();

}

}

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

{

Application.Exit();

}

private void LoadWeekendsOnly()

{

for (DateTime i = new DateTime(DateTime.Now.Year, 1, 1); i <= new DateTime(DateTime.Now.Year, 12, 31); i = i.AddDays(1))

{

if (i.DayOfWeek == DayOfWeek.Saturday || i.DayOfWeek == DayOfWeek.Sunday)

{

mcRight.AddBoldedDate(i);

mcLeft.AddBoldedDate(i);

}

}

mcRight.UpdateBoldedDates();

mcLeft.UpdateBoldedDates();

}

private void LoadListBox()

{

var dates = File.ReadAllLines("writtendates.txt")

.Where(line => !string.IsNullOrWhiteSpace(line))

.Select(line => DateTime.Parse(line))

.OrderBy(date => date)

.ToList();

foreach (var date in dates)

{

listBoxDates.Items.Add(date.ToString("d"));

}

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

if (listBoxDates.SelectedItem != null)

{

DateTime selectedDate = DateTime.Parse(listBoxDates.SelectedItem.ToString());

mcLeft.SelectionStart = selectedDate;

mcLeft.SelectionEnd = selectedDate;

mcLeft.Enabled = true;

}

}

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

{

DateTime selected = mcLeft.SelectionStart;

StreamWriter sw = new StreamWriter("writtendates.txt", true);

sw.WriteLine(selected.ToString("d"));

sw.Close();

listBoxDates.Items.Add(selected.ToString("d"));

listBoxDates.Sorted = true;

mcLeft.AddBoldedDate(selected);

mcLeft.UpdateBoldedDates();

}

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

{

File.WriteAllText("writtendates.txt", string.Empty);

mcLeft.RemoveAllBoldedDates();

listBoxDates.Items.Clear();

}

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

{

mcLeft.Enabled = false;

}

}

}

Дополнительная форма:

using System;

using System.IO;

using System.Windows.Forms;

namespace Task4

{

public partial class FrmHelp : Form

{

private Label txtHelp;

public FrmHelp()

{

this.KeyDown += new KeyEventHandler(Frm\_KeyDown);

this.KeyPreview = true;

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

this.SuspendLayout();

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

this.txtHelp.Size = new System.Drawing.Size(700, 350);

this.txtHelp.ForeColor = System.Drawing.SystemColors.ControlText;

this.txtHelp.Font = new System.Drawing.Font("Times New Roman", 14.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.ClientSize = new System.Drawing.Size(700, 400);

this.Controls.Add(this.txtHelp);

this.Name = "FrmHelp";

this.Text = "Справка";

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

this.ResumeLayout(false);

this.KeyPreview = true;

}

private void Frm\_KeyDown(object sender, KeyEventArgs e)

{

if (e.Control && e.KeyCode == Keys.H)

{

txtHelp.Text = null;

}

}

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

{

StreamReader sr = new StreamReader("C:\\Users\\rodio\\Desktop\\Task 4\\bin\\Debug\\help.txt");

txtHelp.Text = sr.ReadToEnd();

sr.Close();

this.Resize += FrmHelp\_Resize;

}

private void FrmHelp\_Resize(object sender, EventArgs e)

{

this.txtHelp.Size = this.ClientSize;

}

}

}

# **8. Графический пользовательский интерфейс программного обеспечения и его описание.**

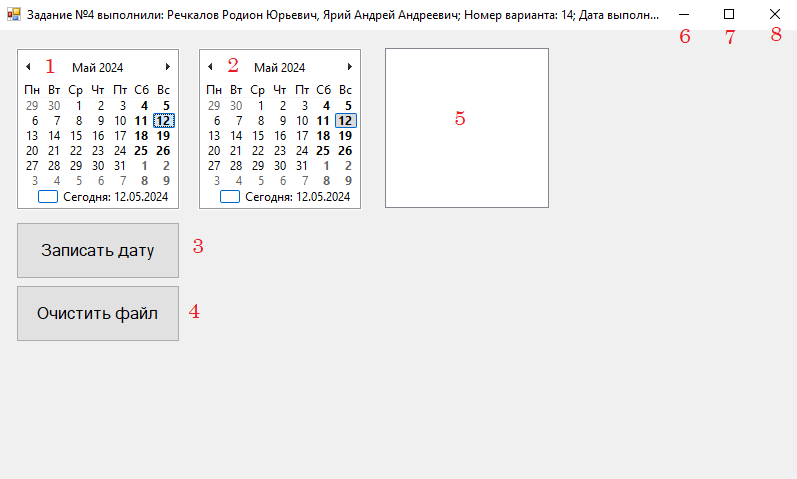


Рисунок 4 – Интерфейс главной формы

1. Левый календарь (для выбора и сохранения дат)
2. Правый календарь (дополнительный календарь)
3. Кнопка записи выбранной в левом календаре даты в файл
4. Кнопка очистки файла с записанными датами
5. Комбинированный список для записи дат
6. Кнопка «свернуть»
7. Кнопка «отрыть на весь экран»
8. Кнопка «закрыть»

# **Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.**

1. Заголовок главной формы должен содержать надпись вида: «Задание №4 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».



Рисунок 5 – Заголовок главной формы

2. Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт.



Рисунок 6 – Дата на момент выполнения задания

3. Предусмотреть комбинированный список, в который будут записаны те же даты из файла, что и в сам календарь. Даты в списке должны быть упорядочены и выбор каждой даты из списка переводит собственный курсор календаря на такую же дату.

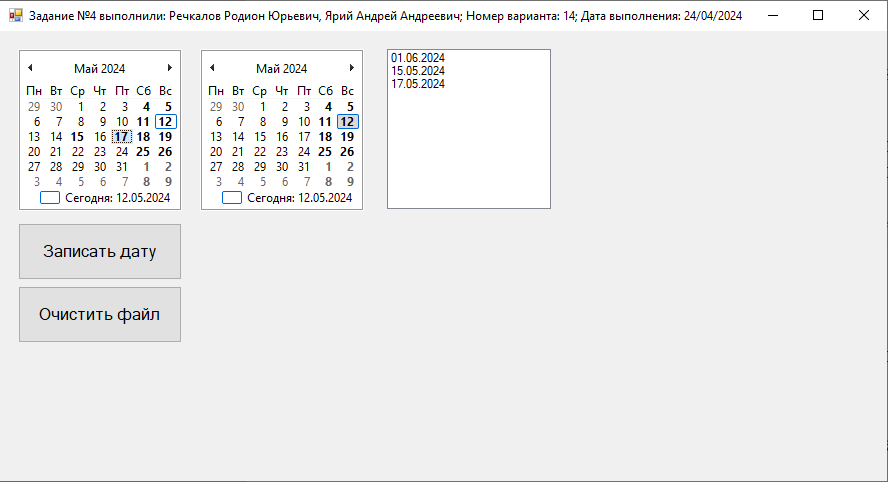


Рисунок 7 – Комбинированный список с записанными датами

4. На форме должны быть размещены два календаря вне зависимости от того, будут ли оба использованы для реализации основного и индивидуального заданий. В случае, когда календарь не используется, необходимо заблокировать его доступность.

См. Рисунок 4

5. Предусмотреть по нажатии сочетания клавиш «*Ctrl+H*» вызов справки о работе с программой из файла в статический по размеру ярлык.

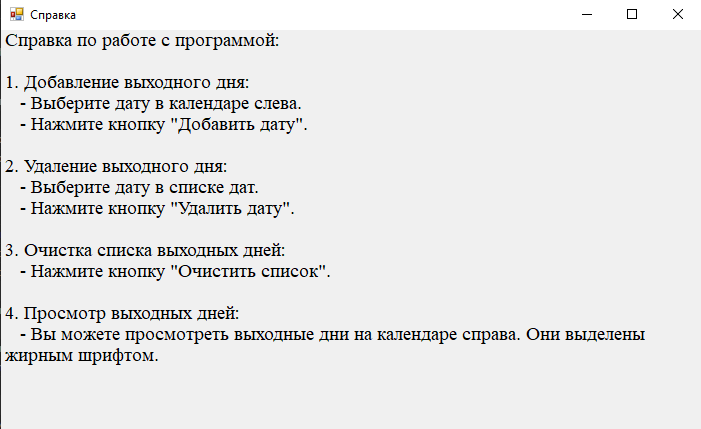


Рисунок 8 – Справка по программе

6.На правом (или нижнем) календаре выполнить выделение всех выходных дней указанного календарного года без учёта праздничных дней. Сведения о выходных днях допустимо хранить в текстовых файлах.

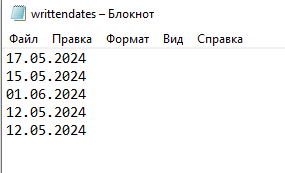


Рисунок 9 – Текстовый файл с записанными датами

# **Расчёт тестовых примеров с использованием составленного программного обеспечения.**

Тестовые примеры:

Проверка кнопки «открыть на весь экран»

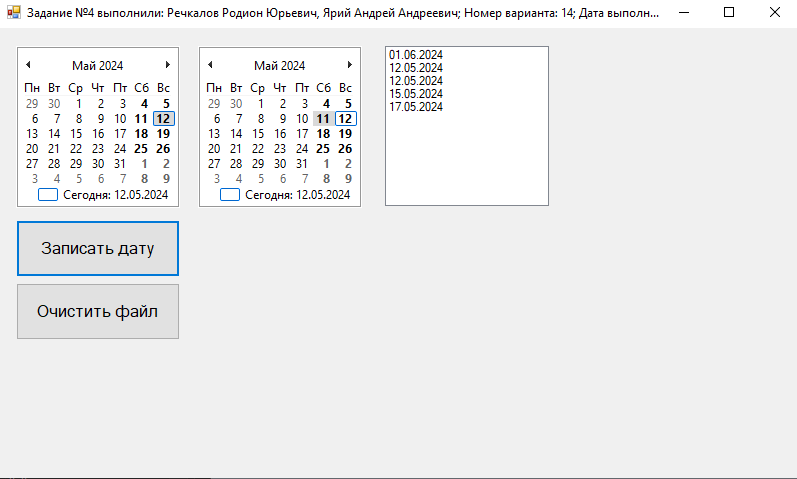


Рисунок 10 – Форма в обычном состоянии

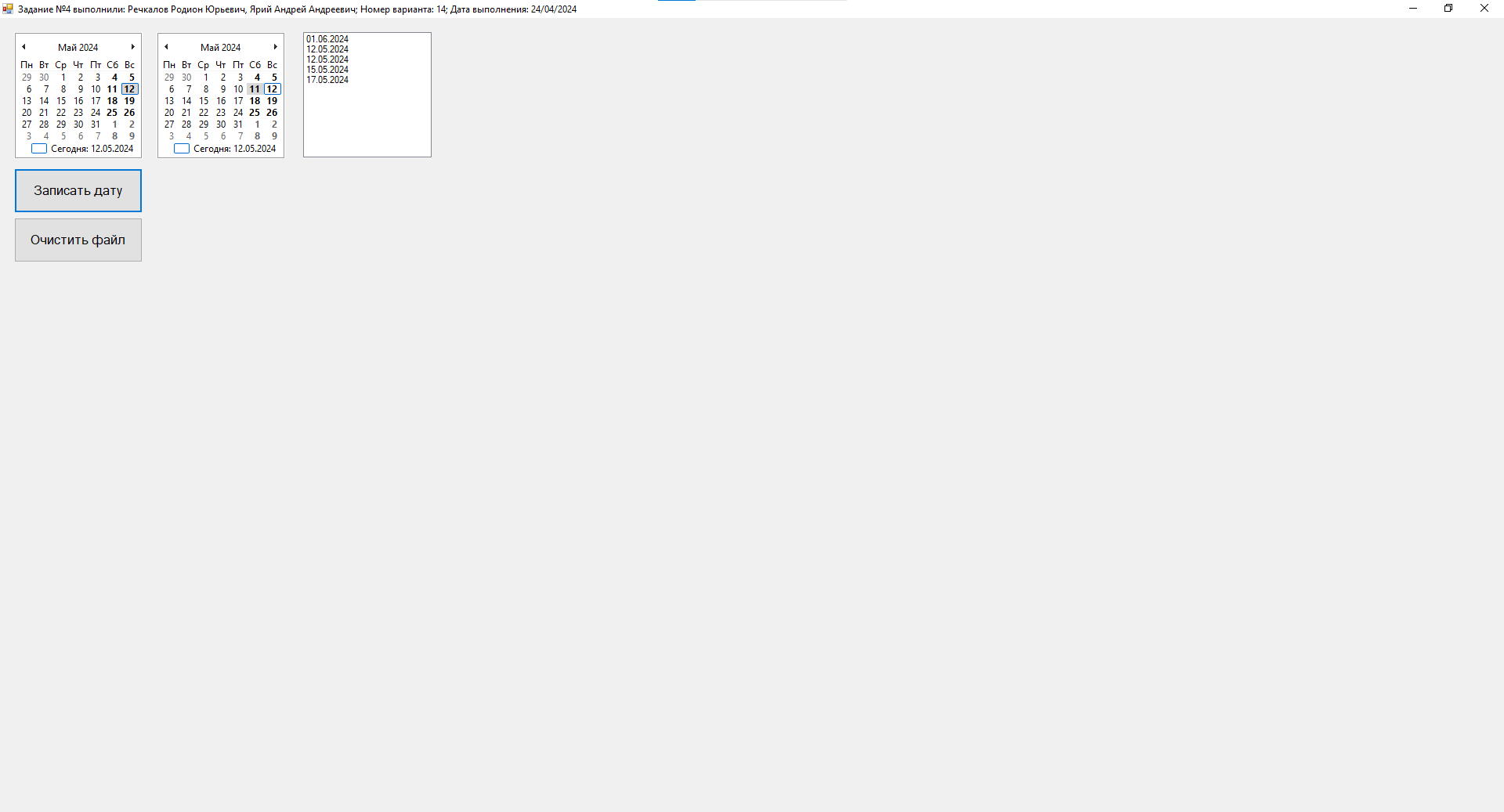


Рисунок 11 – Форма отрытая на весь экран

Проверка записи в файл

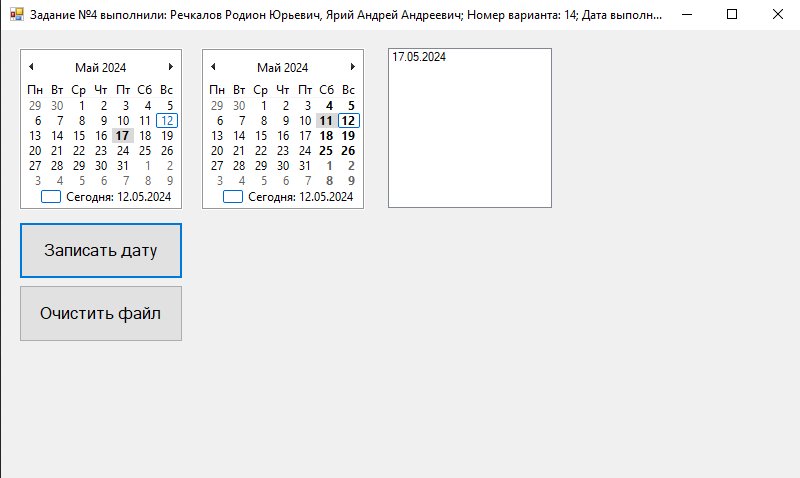


Рисунок 12 – В левом календаре была выбрана дата 17 мая 2024 года

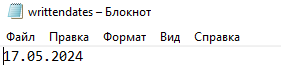


Рисунок 13 – В файле сохранилась выбранная дата

Проверка очистки файла

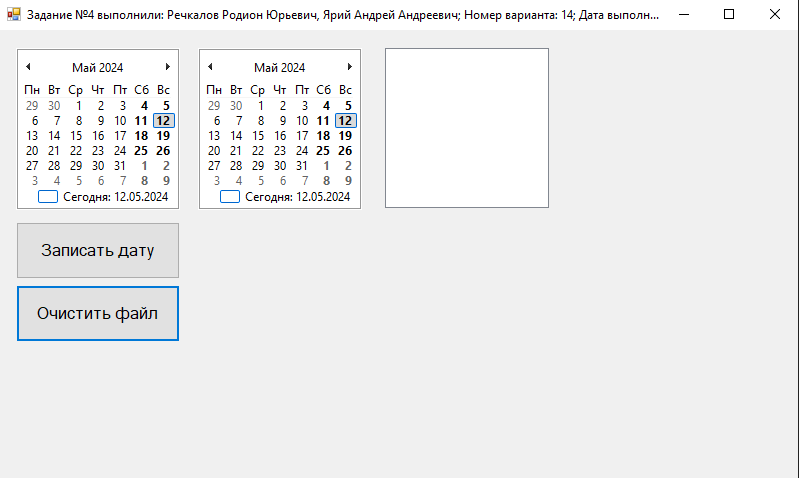


Рисунок 14 – Форма после очистки файла

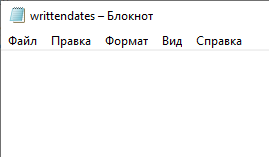


Рисунок 15 – Файл после очистки

# **Формулировка вывода о проделанной работе**

В ходе выполнения работы была достигнута цель по освоению навыков настройки элемента «Календарь на месяц» (MonthCalendar).

Разработанная программа в среде Microsoft Visual Studio с использованием языка Visual C# представляет собой удобное средство взаимодействия с текстовым файлом для отображения дат на календаре. В ходе работы была успешно реализована следующая функциональность:

1.Отображение дат из текстового файла на календаре:

2.Управление выбором дат:

3.Размещение и блокировка календарей:

4.Интеграция справочной информации:

5.Выделение выходных дней:

Работа продемонстрировала возможности и преимущества использования элемента «Календарь на месяц» для создания удобных и функциональных интерфейсов в Windows Forms Application. Полученные навыки и разработанная программа могут быть полезны для дальнейших проектов и приложений, требующих интерактивного отображения и управления датами.