

**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**»

«Календарь»

**Выполнили:** ст. гр. ТКИ-341

Бузаджи Н.М.

Самохвалова М.Д.

**Проверил:** к.т.н. Сафронов А.И.

**Москва – 2024 г**

1. **Цель работы**

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

## 2. Индивидуальная задача

* Заголовок главной формы должен содержать надпись вида: «Задание №4 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».
* Дата выполнения проставляется в момент, когда программа считается законченной и по ней можно готовить итоговый отчёт.
* Даты в списке должны быть упорядочены и выбор каждой даты из списка переводит собственный курсор календаря на такую же дату. Предусмотреть запись в файл интересующих дат, сформированных в результате работы программы (нечётные варианты).
* 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 вызов справки о работе каждого компонента программы путём размещения информации, считанной из файла, в компоненты *ToolTip* (всплывающая подсказка) подсвечиваемых элементов (нечётные варианты).

## Диаграммы классов, входящих в состав решения.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 1 – Диаграмма классов

## Сеть Петри

**Легенда:**

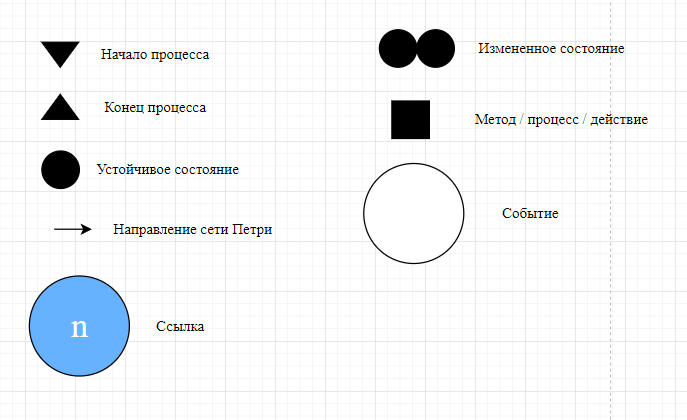


Рисунок 2 – Значение обозначений

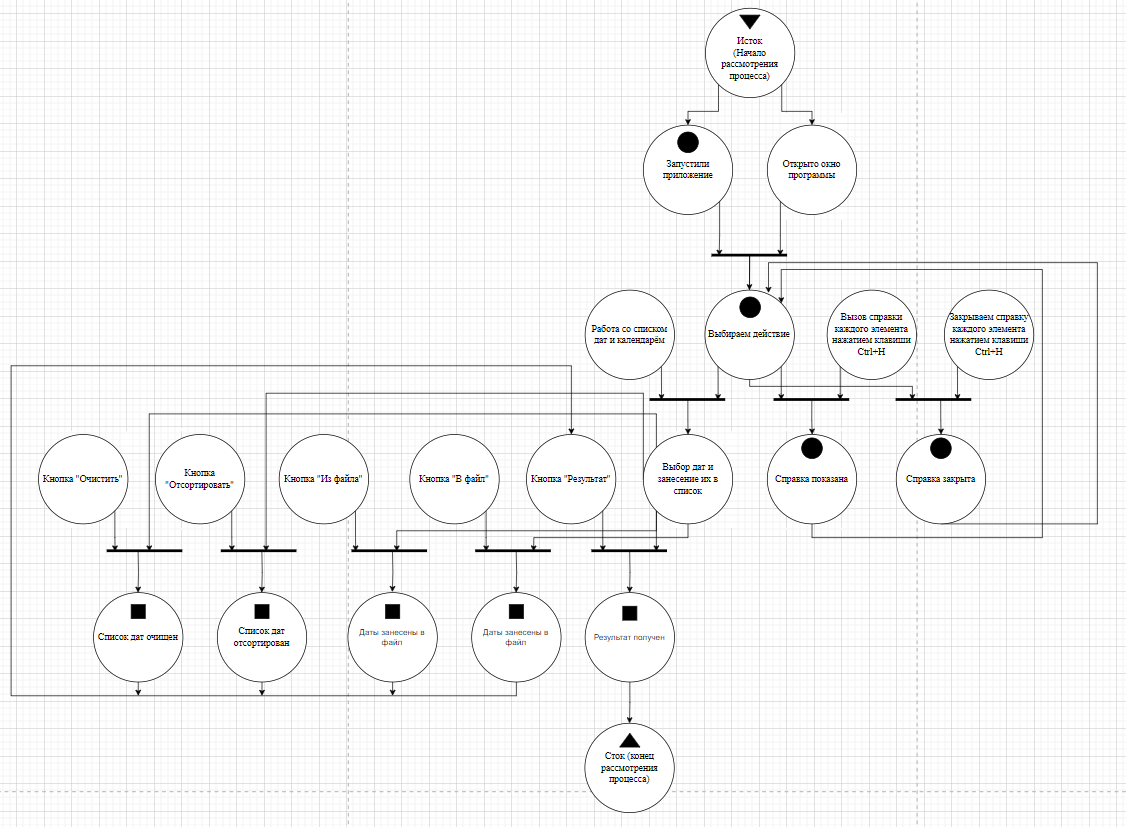


Рисунок 3 – Сеть Петри

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

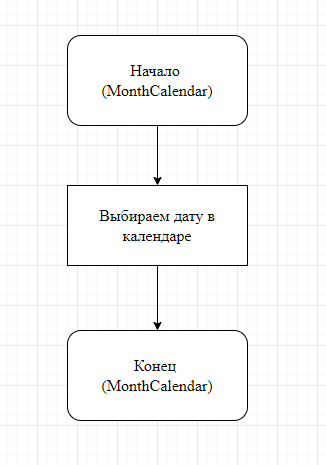


Рисунок 4 – Календарь

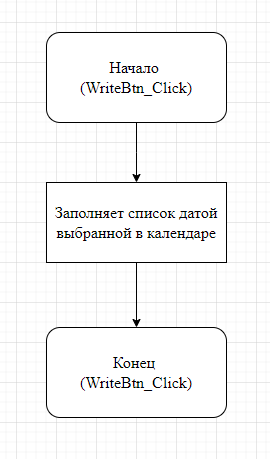


Рисунок 5 – Кнопка «В список»

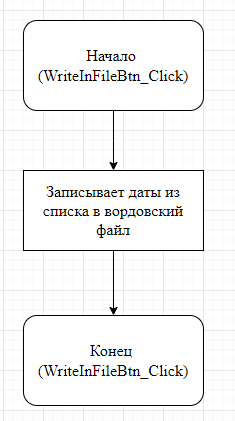


Рисунок 6 – Кнопка «В файл»

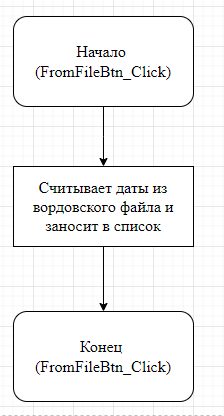


Рисунок 7 – Кнопка «Из файла»

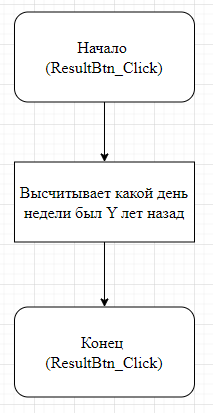


Рисунок 8 – Кнопка «Результат»

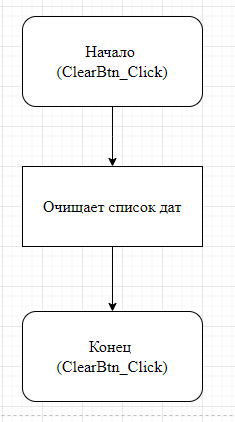
**

Рисунок 9 – Кнопка «Очистить»

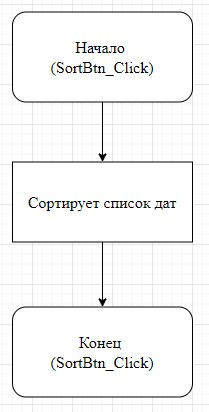


Рисунок 10 – Кнопка «Отсортировать»

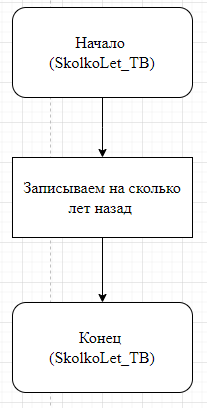


Рисунок 11 – Поле для ввода количества лет назад

## Тестовые примеры

1. Дата: 02.05.2024

Количество лет назад: 9

День недели: Суббота

1. Дата: 10.04.2024

Количество лет назад: 9

День недели: Пятница

1. Дата: 29.06.2024

Количество лет назад: 9

День недели: Понедельник

## Листинг (код) составленного программного обеспечения

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;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Reflection.Emit;

namespace WindowsFormsApp1

{

public partial class Form1 : Form

{

bool a;

public Form1()

{

InitializeComponent();

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

bool a = false;

//toolTip1.Active = a; // Видно/не видно подсказки на форме

Spravka.AutoPopDelay = 5000;

Spravka.InitialDelay = 1000;

Spravka.ReshowDelay = 500;

Spravka.ShowAlways = true;

Spravka.SetToolTip(this.Mc, "Календарь");

Spravka.SetToolTip(this.listDates, "Лист Дат");

Spravka.SetToolTip(this.WriteBtn, "Кнопка записи");

Spravka.SetToolTip(this.WriteInFileBtn, "Кнопка записи в файл");

Spravka.SetToolTip(this.ResultBtn, "Кнопка вывода результата");

Spravka.SetToolTip(this.ClearBtn, "Кнопка очистки");

Spravka.SetToolTip(this.FromFileBtn, "Кнопка считывания из файла");

Spravka.SetToolTip(this.SortBtn, "Кнопка сортировки");

Spravka.SetToolTip(this.CloseBtn, "Кнопка закрыть");

Spravka.SetToolTip(this.ExpandBtn, "Кнопка развернуть");

Spravka.SetToolTip(this.HideBtn, "Кнопка свернуть");

Spravka.SetToolTip(this.ResultLbl, "Поле вывода результата");

Spravka.SetToolTip(this.SkolkoLet\_TB, "Кнопка ввода количества лет назад");

}

IList<DateTime> colDates = new List<DateTime>();

string path = Application.StartupPath + "\\listDates.txt";

void Form\_KeyDown(object sender, KeyEventArgs e)

{

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

{

if (a == false)

{

Spravka.Active = true;

a = true;

}

else if (a == true)

{

Spravka.Active = false;

a = false;

}

}

}

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

{

Application.Exit();

}

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

{

if (a == true)

{

this.WindowState = FormWindowState.Normal;

a = false;

}

else

{

this.WindowState = FormWindowState.Maximized;

a = true;

}

}

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

{

this.WindowState = FormWindowState.Minimized;

}

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

{

listDates.Items.Add(Mc.SelectionRange.Start.ToString("dd.MM.yyyy"));

colDates.Add(Mc.SelectionRange.Start);

}

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

{

listDates.Items.Clear();

colDates.Clear();

}

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

{

StreamWriter sw = new StreamWriter(path);

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

sw.WriteLine(colDates[i].ToString("dd.MM.yyyy"));

sw.Close();

}

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

{

string tmpStr;

string[] dateAndTime, dateProps;

int day, mon, year;

StreamReader sr = new StreamReader(path);

colDates.Clear();

while ((tmpStr = sr.ReadLine()) != null)

{

dateAndTime = tmpStr.Split(' ');

dateProps = dateAndTime[0].Split('.');

int.TryParse(dateProps[0], out day);

int.TryParse(dateProps[1], out mon);

int.TryParse(dateProps[2], out year);

colDates.Add(new DateTime(year, mon, day));

}

fillList();

}

private void fillList()

{

listDates.Items.Clear();

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

listDates.Items.Add(colDates[i].ToString("dd.MM.yyyy"));

}

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

{

// Получаем выбранный индекс из ListBox

int selectedIndex = listDates.SelectedIndex;

// Предположим, что у вас есть List<DateTime> с названием datesList, содержащий ваши даты

if (selectedIndex >= 0 && selectedIndex < colDates.Count)

{

// Получаем выбранную дату из списка

DateTime selectedDate = colDates[selectedIndex];

// Устанавливаем выбранную дату в MonthCalendar

Mc.SetDate(selectedDate);

}

}

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

{

List<(DateTime Date, int Index)> datesList = new List<(DateTime, int)>();

// Добавляем все даты из элементов ListBox в список вместе с их индексами

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

{

if (DateTime.TryParse(listDates.Items[i].ToString(), out DateTime date))

{

datesList.Add((date, i));

}

}

datesList.Sort((x, y) => DateTime.Compare(x.Date, y.Date));

listDates.Items.Clear();

foreach (var (date, index) in datesList)

{

listDates.Items.Add(date);

}

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

{

listDates.SetSelected(i, true); // Выбираем элемент по новому индексу

}

}

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

{

DateTime selectedDate = Mc.SelectionStart;

int yearsAgo = int.Parse(SkolkoLet\_TB.Text);

DateTime newDate = selectedDate.AddYears(-yearsAgo);

DayOfWeek dayOfWeek = newDate.DayOfWeek;

Mc.SetDate(newDate);

ResultLbl.Text = $"День недели {yearsAgo} лет назад от {selectedDate}: {dayOfWeek}";

}

}

}

## Графический пользовательский интерфейс программного обеспечения и его описание.

## Пользовательский интерфейс:

В список – кнопка занесения даты в список

В файл – кнопка занесения списка дат в документ

Результат – кнопка вычисления результата

Из файла – кнопка считывания списка дат из документа

Очистить – кнопка очищения списка дат

Отсортировать – кнопка сортировки списка дат

1 – календарь

2 – поле для списка дат

3 – поле для ввода количества лет назад

4 – поле для вывода результата

Закрыть – закрывает форму

Развернуть – меняет размер формы

Свернуть – сворачивает форму

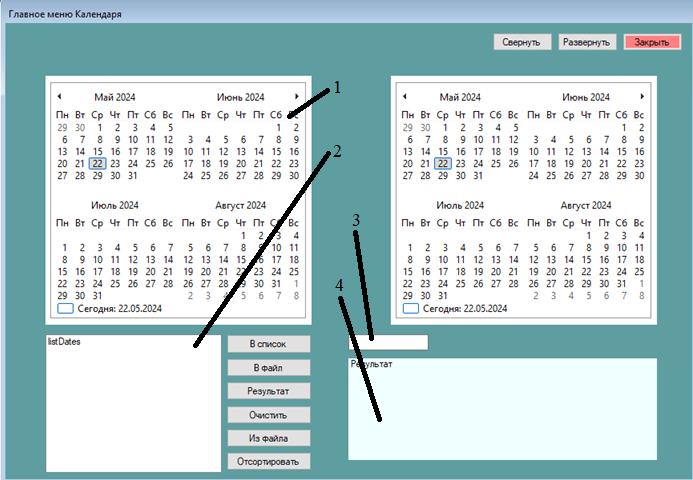


Рисунок 13 – Окно главного меню



Рисунок 14 – Справка определенного элемента

## Расчёт тестовых примеров с использованием составленного программного обеспечения.

## 

Рисунок 15 – Результат первого тестового примера

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 16 – Результат второго тестового примера

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 17 – Результат третьего тестового примера

1. **Вывод по работе**

В данной лабораторной работе были изучены разные взаимодействия с инструментом MonthCalendar. Мы научились заносить указанные даты в список, записывать данный список в документ Word, а также доставать данные из этого документа, сортировать список дат. Также был реализован алгоритм, который позволяет высчитать какой день был с определенной даты Y лет назад.