**Контрольная работа 1. Калькулятор. Гришутина А.В. ЭФБО-17-24**

**Form1.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

using System.Linq;

using System.Collections.Generic;

using System.IO;

using Microsoft.VisualBasic;

namespace Calendar

{

public partial class Form1 : Form

{

//элементы управления

private MonthCalendar monthCalendar1;

private Label lblSelectedDate;

private Label lblDateChanged;

private Label lblSelectedCount;

private Button btnAddBoldDate;

private Button btnRemoveBoldDate;

private Button btnSetColors;

private Button btnResetColors;

private Button btnClearSelection;

private Button btnSelectRange;

private Button btnAddHoliday;

private DateTimePicker datePickerForBold;

private DateTimePicker datePickerStart;

private DateTimePicker datePickerEnd;

private CheckBox chkShowToday;

private CheckBox chkShowTodayCircle;

private CheckBox chkShowWeekNumbers;

private GroupBox groupBoxEvents;

private GroupBox groupBoxControls;

private GroupBox groupBoxSelection;

private GroupBox groupBoxAppearance;

private GroupBox groupBoxLeapYear;

private NumericUpDown numMaxSelection;

private ComboBox cmbFirstDayOfWeek;

private ListBox lstSelectedDates;

private TextBox txtHolidayName;

private Label label1;

private Label label2;

private Label label3;

private Label label4;

private Label label5;

private Label lblLeapYearInfo;

private Button btnCheckLeapYear;

private TextBox txtYearInput;

private Button btnGoToToday;

private Label lblCurrentTime;

private Button btnCopyDate;

private Label lblDaysUntil;

private Timer timer;

private StatusStrip statusStrip1;

private ToolStripStatusLabel toolStripStatusLabel1;

private ToolStripStatusLabel toolStripStatusLabel2;

private ToolStripStatusLabel toolStripStatusLabel3;

private Button btnAddMultiDayHoliday;

private NumericUpDown numHolidayDays;

private Label label6;

private Button btnSaveHolidays;

private Button btnLoadHolidays;

private Button btnShowHolidayFileInfo;

private Button btnRemoveHoliday;

private Button btnClearAllHolidays;

private NumericUpDown numCalendarMonths;

private Label label7;

private Button btnSetCalendarSize;

// всякие константы

private const int DEFAULT\_MAX\_SELECTION = 30;

private const int MAX\_DISPLAY\_DATES = 100; // чтобы не тормозило

// тут храним праздники

private Dictionary<DateTime, List<string>> holidays = new Dictionary<DateTime, List<string>>();

private string holidaysFilePath = Path.Combine(Application.StartupPath, "holidays.txt");

// проверяем високосный год или нет

private bool IsLeapYear(int year)

{

return (year % 4 == 0) && (year % 100 != 0 || year % 400 == 0);

}

//получает количество дней в феврале для указанного года

private int GetDaysInFebruary(int year)

{

return IsLeapYear(year) ? 29 : 28;

}

public Form1()

{

InitializeComponent();

SetupCalendar();

}

private void SetupCalendar()

{

try

{

monthCalendar1.MaxSelectionCount = DEFAULT\_MAX\_SELECTION;

monthCalendar1.ShowToday = true;

monthCalendar1.ShowTodayCircle = true;

monthCalendar1.ShowWeekNumbers = false;

monthCalendar1.MinDate = new DateTime(1900, 1, 1);

monthCalendar1.MaxDate = new DateTime(2200, 12, 31);

monthCalendar1.CalendarDimensions = new Size(2, 2);

//подключение событий

monthCalendar1.DateSelected += MonthCalendar1\_DateSelected;

monthCalendar1.DateChanged += MonthCalendar1\_DateChanged;

monthCalendar1.ShowToday = true;

//подключение событий кнопок

btnAddBoldDate.Click += BtnAddBoldDate\_Click;

btnRemoveBoldDate.Click += BtnRemoveBoldDate\_Click;

btnSetColors.Click += BtnSetColors\_Click;

btnResetColors.Click += BtnResetColors\_Click;

btnClearSelection.Click += BtnClearSelection\_Click;

btnSelectRange.Click += BtnSelectRange\_Click;

btnAddHoliday.Click += BtnAddHoliday\_Click;

btnCheckLeapYear.Click += BtnCheckLeapYear\_Click;

btnGoToToday.Click += BtnGoToToday\_Click;

btnCopyDate.Click += BtnCopyDate\_Click;

btnAddMultiDayHoliday.Click += BtnAddMultiDayHoliday\_Click;

btnSaveHolidays.Click += BtnSaveHolidays\_Click;

btnLoadHolidays.Click += BtnLoadHolidays\_Click;

btnShowHolidayFileInfo.Click += BtnShowHolidayFileInfo\_Click;

btnRemoveHoliday.Click += BtnRemoveHoliday\_Click;

btnClearAllHolidays.Click += BtnClearAllHolidays\_Click;

btnSetCalendarSize.Click += BtnSetCalendarSize\_Click;

chkShowToday.CheckedChanged += ChkShowToday\_CheckedChanged;

chkShowTodayCircle.CheckedChanged += ChkShowTodayCircle\_CheckedChanged;

chkShowWeekNumbers.CheckedChanged += ChkShowWeekNumbers\_CheckedChanged;

numMaxSelection.ValueChanged += NumMaxSelection\_ValueChanged;

cmbFirstDayOfWeek.SelectedIndexChanged += CmbFirstDayOfWeek\_SelectedIndexChanged;

// настраиваем

chkShowToday.Checked = monthCalendar1.ShowToday;

chkShowTodayCircle.Checked = monthCalendar1.ShowTodayCircle;

chkShowWeekNumbers.Checked = monthCalendar1.ShowWeekNumbers;

numMaxSelection.Value = monthCalendar1.MaxSelectionCount;

// заполняем список дней недели

PopulateFirstDayOfWeekComboBox();

// добавляем тестовые данные

AddSampleBoldDates();

LoadHolidaysFromFile(); // грузим сохраненные праздники

AddMissingSampleHolidays(); // добавляем только недостающие

// ставим текущий год

txtYearInput.Text = DateTime.Now.Year.ToString();

// запускаем таймер для времени

timer = new Timer();

timer.Interval = 1000;

timer.Tick += Timer\_Tick;

timer.Start();

// обновляем все что нужно

UpdateSelectedDatesDisplay();

UpdateLeapYearInfo();

UpdateCurrentTime();

UpdateDaysUntilSelectedDate();

UpdateStatusBar();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка инициализации календаря: {ex.Message}",

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

}

}

//заполнение ComboBox для дней недели

private void PopulateFirstDayOfWeekComboBox()

{

var daysOfWeek = new Dictionary<Day, string>

{

{ Day.Sunday, "Воскресенье" },

{ Day.Monday, "Понедельник" },

{ Day.Tuesday, "Вторник" },

{ Day.Wednesday, "Среда" },

{ Day.Thursday, "Четверг" },

{ Day.Friday, "Пятница" },

{ Day.Saturday, "Суббота" }

};

cmbFirstDayOfWeek.DisplayMember = "Value";

cmbFirstDayOfWeek.ValueMember = "Key";

cmbFirstDayOfWeek.DataSource = new BindingSource(daysOfWeek, null);

cmbFirstDayOfWeek.SelectedValue = monthCalendar1.FirstDayOfWeek;

}

private void AddSampleBoldDates()

{

try

{

DateTime today = DateTime.Today;

int currentYear = today.Year;

//выделяем первое и пятнадцатое число каждого месяца

monthCalendar1.MonthlyBoldedDates = new DateTime[]

{

new DateTime(currentYear, 1, 1),

new DateTime(currentYear, 1, 15)

};

//выделяем конкретные даты

monthCalendar1.BoldedDates = new DateTime[]

{

today.AddDays(1),

today.AddDays(7),

today.AddDays(14)

};

//ежегодные даты

var annuallyBoldedDates = new List<DateTime>

{

new DateTime(currentYear, 1, 1), // Новый год

new DateTime(currentYear, 3, 8), // 8 марта

new DateTime(currentYear, 5, 9), // 9 мая

new DateTime(currentYear, 12, 25), // Рождество

new DateTime(currentYear, 12, 31) // Новый год

};

// добавляем 29 февраля для високосных лет

if (IsLeapYear(currentYear))

{

annuallyBoldedDates.Add(new DateTime(currentYear, 2, 29));

}

monthCalendar1.AnnuallyBoldedDates = annuallyBoldedDates.ToArray();

monthCalendar1.UpdateBoldedDates();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка добавления выделенных дат: {ex.Message}",

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

}

}

// добавляем только недостающие предустановленные праздники

private void AddMissingSampleHolidays()

{

try

{

int currentYear = DateTime.Today.Year;

// список предустановленных праздников

var sampleHolidays = new Dictionary<DateTime, List<string>>

{

{ new DateTime(currentYear, 1, 1), new List<string> { "Новый год" } },

{ new DateTime(currentYear, 3, 8), new List<string> { "Международный женский день" } },

{ new DateTime(currentYear, 5, 1), new List<string> { "День труда" } },

{ new DateTime(currentYear, 5, 9), new List<string> { "День Победы" } },

{ new DateTime(currentYear, 12, 25), new List<string> { "Рождество" } }

};

// добавляем специальный праздник для 29 февраля в високосные годы

if (IsLeapYear(currentYear))

{

sampleHolidays.Add(new DateTime(currentYear, 2, 29), new List<string> { "День високосного года" });

}

// добавляем только те праздники, которых еще нет

foreach (var holiday in sampleHolidays)

{

if (!holidays.ContainsKey(holiday.Key))

{

holidays[holiday.Key] = new List<string>(holiday.Value);

monthCalendar1.AddBoldedDate(holiday.Key);

}

}

monthCalendar1.UpdateBoldedDates();

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка добавления недостающих праздников: {ex.Message}");

}

}

// когда выбрали дату

private void MonthCalendar1\_DateSelected(object sender, DateRangeEventArgs e)

{

if (e == null) return;

UpdateSelectedDatesDisplay();

CheckForHolidays(e.Start, e.End);

}

//проверка на null

private void MonthCalendar1\_DateChanged(object sender, DateRangeEventArgs e)

{

if (e == null) return;

lblDateChanged.Text = $"Последнее изменение: {DateTime.Now.ToShortTimeString()}\n" +

$"Период: {e.Start.ToShortDateString()} - {e.End.ToShortDateString()}";

//обновляем информацию о високосном годе при изменении даты

UpdateLeapYearInfo();

}

private void UpdateSelectedDatesDisplay()

{

try

{

SelectionRange selection = monthCalendar1.SelectionRange;

DateTime start = selection.Start;

DateTime end = selection.End;

if (start == end)

{

lblSelectedDate.Text = $"Выбранная дата: {start.ToShortDateString()}";

lblSelectedCount.Text = "Количество дат: 1";

}

else

{

int days = (int)(end - start).TotalDays + 1;

lblSelectedDate.Text = $"Период: {start.ToShortDateString()} - {end.ToShortDateString()}";

lblSelectedCount.Text = $"Количество дней: {days}";

}

UpdateSelectedDatesList(start, end);

UpdateDaysUntilSelectedDate();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка обновления отображения: {ex.Message}",

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

}

}

private void UpdateLeapYearInfo()

{

try

{

int currentYear = monthCalendar1.SelectionStart.Year;

bool isLeap = IsLeapYear(currentYear);

int februaryDays = GetDaysInFebruary(currentYear);

string leapYearText = isLeap ? "Високосный год" : "Обычный год";

string leapYearColor = isLeap ? "зеленый" : "черный";

lblLeapYearInfo.Text = $"Текущий год: {currentYear}\n" +

$"Тип года: {leapYearText}\n" +

$"Дней в феврале: {februaryDays}\n" +

$"Всего дней в году: {(isLeap ? 366 : 365)}";

lblLeapYearInfo.ForeColor = isLeap ? Color.Green : Color.Black;

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка обновления информации о високосном годе: {ex.Message}",

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

}

}

//ограничение на количество отображаемых дат

private void UpdateSelectedDatesList(DateTime start, DateTime end)

{

try

{

lstSelectedDates.Items.Clear();

// проверка на слишком большой диапазон

TimeSpan span = end - start;

if (span.TotalDays > MAX\_DISPLAY\_DATES)

{

lstSelectedDates.Items.Add($"Слишком большой диапазон дат для отображения (>{MAX\_DISPLAY\_DATES} дней)");

lstSelectedDates.Items.Add($"Всего дней: {(int)span.TotalDays + 1}");

return;

}

DateTime current = start;

while (current <= end)

{

string dateStr = current.ToShortDateString();

string dayOfWeek = current.ToString("dddd");

int year = current.Year;

bool isLeap = IsLeapYear(year);

// проверяем, есть ли праздники в этот день

if (holidays.ContainsKey(current))

{

dateStr += $" ({dayOfWeek}) - {string.Join(", ", holidays[current])}";

}

else

{

dateStr += $" ({dayOfWeek})";

}

//информация о високосном годе для 29 февраля

if (current.Month == 2 && current.Day == 29)

{

dateStr += $" [29 февраля - високосный год!]";

}

else if (current.Month == 2 && current.Day == 28 && !isLeap)

{

dateStr += $" [Последний день февраля - обычный год]";

}

lstSelectedDates.Items.Add(dateStr);

current = current.AddDays(1);

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка обновления списка дат: {ex.Message}",

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

}

}

private void CheckForHolidays(DateTime start, DateTime end)

{

try

{

List<string> foundHolidays = new List<string>();

DateTime current = start;

while (current <= end)

{

if (holidays.ContainsKey(current))

{

foundHolidays.Add($"{current.ToShortDateString()} - {string.Join(", ", holidays[current])}");

}

current = current.AddDays(1);

}

if (foundHolidays.Count > 0)

{

MessageBox.Show($"В выбранном периоде есть праздники:\n{string.Join("\n", foundHolidays)}",

"Праздники найдены", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка проверки праздников: {ex.Message}",

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

}

}

//обработка исключений

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

{

try

{

DateTime selectedDate = datePickerForBold.Value.Date;

monthCalendar1.AddBoldedDate(selectedDate);

monthCalendar1.UpdateBoldedDates();

MessageBox.Show($"Дата {selectedDate.ToShortDateString()} добавлена к выделенным.",

"Дата добавлена", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка добавления выделенной даты: {ex.Message}",

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

}

}

//обработка исключений

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

{

try

{

DateTime selectedDate = datePickerForBold.Value.Date;

monthCalendar1.RemoveBoldedDate(selectedDate);

monthCalendar1.UpdateBoldedDates();

MessageBox.Show($"Дата {selectedDate.ToShortDateString()} удалена из выделенных.",

"Дата удалена", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка удаления выделенной даты: {ex.Message}",

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

}

}

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

{

try

{

monthCalendar1.TitleBackColor = Color.DarkBlue;

monthCalendar1.TitleForeColor = Color.White;

monthCalendar1.BackColor = Color.LightSteelBlue;

monthCalendar1.ForeColor = Color.DarkBlue;

monthCalendar1.TrailingForeColor = Color.Gray;

monthCalendar1.Refresh();

monthCalendar1.Invalidate();

MessageBox.Show("Цвета изменены! Если изменения не видны, попробуйте переключить месяцы.",

"Цвета обновлены", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка изменения цветов: {ex.Message}",

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

}

}

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

{

try

{

monthCalendar1.TitleBackColor = SystemColors.ActiveCaption;

monthCalendar1.TitleForeColor = SystemColors.ActiveCaptionText;

monthCalendar1.BackColor = SystemColors.Window;

monthCalendar1.ForeColor = SystemColors.WindowText;

monthCalendar1.TrailingForeColor = SystemColors.GrayText;

monthCalendar1.Refresh();

monthCalendar1.Invalidate();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка сброса цветов: {ex.Message}",

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

}

}

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

{

try

{

monthCalendar1.SetDate(DateTime.Today);

UpdateSelectedDatesDisplay();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка очистки выбора: {ex.Message}",

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

}

}

//проверка диапазона дат

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

{

try

{

DateTime start = datePickerStart.Value.Date;

DateTime end = datePickerEnd.Value.Date;

if (start > end)

{

MessageBox.Show("Дата начала не может быть больше даты окончания!",

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

return;

}

TimeSpan span = end - start;

int totalDays = (int)span.TotalDays + 1;

if (totalDays > monthCalendar1.MaxSelectionCount)

{

MessageBox.Show($"Выбранный период превышает максимальное количество дней ({monthCalendar1.MaxSelectionCount})!",

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

return;

}

monthCalendar1.SelectionStart = start;

monthCalendar1.SelectionEnd = end;

UpdateSelectedDatesDisplay();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка выбора диапазона: {ex.Message}",

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

}

}

//добавить праздник

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

{

try

{

DateTime date = datePickerForBold.Value.Date;

string name = txtHolidayName.Text.Trim();

if (string.IsNullOrEmpty(name))

{

MessageBox.Show("Введите название праздника!",

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

return;

}

if (holidays.ContainsKey(date))

{

holidays[date].Add(name);

}

else

{

holidays[date] = new List<string> { name };

monthCalendar1.AddBoldedDate(date);

}

monthCalendar1.UpdateBoldedDates();

txtHolidayName.Clear();

UpdateSelectedDatesDisplay();

UpdateStatusBar();

SaveHolidaysToFile();

MessageBox.Show($"Праздник '{name}' добавлен на {date.ToShortDateString()}",

"Праздник добавлен", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка добавления праздника: {ex.Message}",

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

}

}

private void ChkShowToday\_CheckedChanged(object sender, EventArgs e)

{

monthCalendar1.ShowToday = chkShowToday.Checked;

}

private void ChkShowTodayCircle\_CheckedChanged(object sender, EventArgs e)

{

monthCalendar1.ShowTodayCircle = chkShowTodayCircle.Checked;

}

private void ChkShowWeekNumbers\_CheckedChanged(object sender, EventArgs e)

{

monthCalendar1.ShowWeekNumbers = chkShowWeekNumbers.Checked;

}

private void NumMaxSelection\_ValueChanged(object sender, EventArgs e)

{

try

{

monthCalendar1.MaxSelectionCount = (int)numMaxSelection.Value;

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка изменения максимального количества выбранных дат: {ex.Message}",

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

}

}

//с какого дня начинается неделя

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

{

try

{

if (cmbFirstDayOfWeek.SelectedValue is Day selectedDay)

{

monthCalendar1.FirstDayOfWeek = selectedDay;

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка изменения первого дня недели: {ex.Message}",

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

}

}

//кнопка високосного года

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

{

try

{

if (int.TryParse(txtYearInput.Text, out int year))

{

if (year >= 1 && year <= 9999)

{

bool isLeap = IsLeapYear(year);

int februaryDays = GetDaysInFebruary(year);

int totalDays = isLeap ? 366 : 365;

string result = $"Год {year}:\n" +

$"• {(isLeap ? "Високосный" : "Обычный")} год\n" +

$"• Дней в феврале: {februaryDays}\n" +

$"• Всего дней в году: {totalDays}";

MessageBox.Show(result, "Информация о годе",

MessageBoxButtons.OK,

isLeap ? MessageBoxIcon.Information : MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Год должен быть в диапазоне от 1 до 9999!",

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

}

}

else

{

MessageBox.Show("Введите корректный год!",

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

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка проверки високосного года: {ex.Message}",

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

}

}

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

{

UpdateCurrentTime();

}

private void UpdateCurrentTime()

{

try

{

lblCurrentTime.Text = $"Текущее время: {DateTime.Now.ToString("HH:mm:ss")}";

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка обновления времени: {ex.Message}");

}

}

private void UpdateDaysUntilSelectedDate()

{

try

{

DateTime today = DateTime.Today;

DateTime selectedDate = monthCalendar1.SelectionStart.Date;

if (selectedDate == today)

{

lblDaysUntil.Text = "Сегодня!";

lblDaysUntil.ForeColor = Color.Blue;

}

else if (selectedDate > today)

{

int days = (selectedDate - today).Days;

lblDaysUntil.Text = $"Дней до выбранной даты: {days}";

lblDaysUntil.ForeColor = Color.DarkGreen;

}

else

{

int days = (today - selectedDate).Days;

lblDaysUntil.Text = $"Дней назад: {days}";

lblDaysUntil.ForeColor = Color.DarkOrange;

}

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка обновления дней до даты: {ex.Message}");

}

}

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

{

try

{

monthCalendar1.SetDate(DateTime.Today);

UpdateSelectedDatesDisplay();

UpdateDaysUntilSelectedDate();

MessageBox.Show("Переход к сегодняшней дате выполнен!",

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

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка перехода к сегодняшней дате: {ex.Message}",

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

}

}

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

{

try

{

string dateString = monthCalendar1.SelectionStart.ToString("dd.MM.yyyy");

Clipboard.SetText(dateString);

MessageBox.Show($"Дата '{dateString}' скопирована в буфер обмена!",

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

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка копирования даты: {ex.Message}",

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

}

}

// обновляем статус бар

private void UpdateStatusBar()

{

try

{

int totalHolidays = holidays.Values.SelectMany(list => list).Count();

toolStripStatusLabel1.Text = "Статус: Готов";

toolStripStatusLabel2.Text = $"Праздников: {totalHolidays}";

toolStripStatusLabel3.Text = $"Время: {DateTime.Now.ToString("HH:mm:ss")}";

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка обновления статус-бара: {ex.Message}");

}

}

// добавляем праздник на несколько дней

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

{

try

{

DateTime startDate = datePickerForBold.Value.Date;

string name = txtHolidayName.Text.Trim();

int days = (int)numHolidayDays.Value;

if (string.IsNullOrEmpty(name))

{

MessageBox.Show("Введите название праздника!",

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

return;

}

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

{

DateTime currentDate = startDate.AddDays(i);

if (holidays.ContainsKey(currentDate))

{

holidays[currentDate].Add($"{name} (день {i + 1})");

}

else

{

holidays[currentDate] = new List<string> { $"{name} (день {i + 1})" };

monthCalendar1.AddBoldedDate(currentDate);

}

}

monthCalendar1.UpdateBoldedDates();

txtHolidayName.Clear();

UpdateSelectedDatesDisplay();

UpdateStatusBar();

SaveHolidaysToFile();

MessageBox.Show($"Многодневный праздник '{name}' добавлен на {days} дней начиная с {startDate.ToShortDateString()}",

"Праздник добавлен", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка добавления многодневного праздника: {ex.Message}",

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

}

}

// сохраняем праздники в файлик

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

{

try

{

int totalHolidays = holidays.Values.SelectMany(list => list).Count();

if (totalHolidays == 0)

{

MessageBox.Show("Нет праздников для сохранения!",

"Предупреждение", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

using (StreamWriter writer = new StreamWriter(holidaysFilePath, false, System.Text.Encoding.UTF8))

{

writer.WriteLine($"# Файл праздников - сохранено {DateTime.Now:yyyy-MM-dd HH:mm:ss}");

writer.WriteLine($"# Формат: ДАТА|ПРАЗДНИК1,ПРАЗДНИК2");

foreach (var holiday in holidays.OrderBy(h => h.Key))

{

string holidaysList = string.Join(",", holiday.Value);

writer.WriteLine($"{holiday.Key:yyyy-MM-dd}|{holidaysList}");

}

}

MessageBox.Show($"Праздники сохранены в файл!\nПуть: {holidaysFilePath}\nКоличество праздников: {totalHolidays}",

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

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка сохранения праздников: {ex.Message}\nПуть: {holidaysFilePath}",

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

}

}

// грузим праздники из файла

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

{

try

{

LoadHolidaysFromFile();

UpdateSelectedDatesDisplay();

UpdateStatusBar();

MessageBox.Show("Праздники загружены из файла!",

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

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка загрузки праздников: {ex.Message}",

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

}

}

// метод для загрузки праздников

private void LoadHolidaysFromFile()

{

try

{

if (File.Exists(holidaysFilePath))

{

// полностью очищаем праздники и грузим из файла

holidays.Clear();

// убираем все выделенные даты

monthCalendar1.RemoveAllBoldedDates();

using (StreamReader reader = new StreamReader(holidaysFilePath, System.Text.Encoding.UTF8))

{

string line;

int lineNumber = 0;

while ((line = reader.ReadLine()) != null)

{

lineNumber++;

// пропускаем комментарии в файле

if (string.IsNullOrWhiteSpace(line) || line.StartsWith("#"))

continue;

string[] parts = line.Split('|');

if (parts.Length >= 2)

{

try

{

DateTime date = DateTime.Parse(parts[0]);

List<string> holidayNames = parts[1].Split(',').Where(h => !string.IsNullOrWhiteSpace(h)).ToList();

if (holidayNames.Count > 0)

{

holidays[date] = holidayNames;

monthCalendar1.AddBoldedDate(date);

}

}

catch (Exception parseEx)

{

System.Diagnostics.Debug.WriteLine($"Ошибка парсинга строки {lineNumber}: {parseEx.Message}");

}

}

}

}

monthCalendar1.UpdateBoldedDates();

System.Diagnostics.Debug.WriteLine($"Загружено {holidays.Count} дат с праздниками из файла");

}

else

{

System.Diagnostics.Debug.WriteLine($"Файл праздников не найден: {holidaysFilePath}");

}

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка загрузки праздников из файла: {ex.Message}");

}

}

// показываем инфу о файле с праздниками

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

{

try

{

string info = $"Информация о файле праздников:\n\n";

info += $"Путь к файлу: {holidaysFilePath}\n";

info += $"Файл существует: {(File.Exists(holidaysFilePath) ? "Да" : "Нет")}\n\n";

if (File.Exists(holidaysFilePath))

{

FileInfo fileInfo = new FileInfo(holidaysFilePath);

info += $"Размер файла: {fileInfo.Length} байт\n";

info += $"Дата создания: {fileInfo.CreationTime:yyyy-MM-dd HH:mm:ss}\n";

info += $"Дата изменения: {fileInfo.LastWriteTime:yyyy-MM-dd HH:mm:ss}\n\n";

using (StreamReader reader = new StreamReader(holidaysFilePath, System.Text.Encoding.UTF8))

{

string line;

int lineCount = 0;

info += "Содержимое файла (первые 10 строк):\n";

while ((line = reader.ReadLine()) != null && lineCount < 10)

{

if (!string.IsNullOrWhiteSpace(line) && !line.StartsWith("#"))

{

info += $"{line}\n";

lineCount++;

}

}

if (lineCount == 0)

{

info += "Файл пустой или содержит только комментарии\n";

}

}

}

info += $"\nТекущие праздники в памяти:\n";

int totalHolidays = holidays.Values.SelectMany(list => list).Count();

info += $"Количество дат с праздниками: {holidays.Count}\n";

info += $"Общее количество праздников: {totalHolidays}\n";

if (holidays.Count > 0)

{

info += "\nПримеры праздников:\n";

int count = 0;

foreach (var holiday in holidays.Take(5))

{

info += $"{holiday.Key:yyyy-MM-dd}: {string.Join(", ", holiday.Value)}\n";

count++;

}

if (holidays.Count > 5)

{

info += $"... и еще {holidays.Count - 5} дат\n";

}

}

MessageBox.Show(info, "Информация о файле праздников", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка получения информации о файле: {ex.Message}",

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

}

}

// удаляем праздник с выбранной даты

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

{

try

{

DateTime date = datePickerForBold.Value.Date;

if (!holidays.ContainsKey(date))

{

MessageBox.Show("На выбранную дату нет праздников для удаления!",

"Предупреждение", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

var dateHolidays = holidays[date];

if (dateHolidays.Count == 1)

{

string removedHoliday = dateHolidays[0];

holidays.Remove(date);

monthCalendar1.RemoveBoldedDate(date);

monthCalendar1.UpdateBoldedDates();

MessageBox.Show($"Праздник '{removedHoliday}' удален с {date.ToShortDateString()}",

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

}

else

{

string holidayList = string.Join("\n", dateHolidays.Select((h, i) => $"{i + 1}. {h}"));

string input = Interaction.InputBox(

$"Выберите праздник для удаления:\n\n{holidayList}\n\nВведите номер праздника:",

"Удаление праздника", "1");

if (int.TryParse(input, out int index) && index > 0 && index <= dateHolidays.Count)

{

string removedHoliday = dateHolidays[index - 1];

dateHolidays.RemoveAt(index - 1);

if (dateHolidays.Count == 0)

{

holidays.Remove(date);

monthCalendar1.RemoveBoldedDate(date);

}

monthCalendar1.UpdateBoldedDates();

MessageBox.Show($"Праздник '{removedHoliday}' удален с {date.ToShortDateString()}",

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

}

else

{

MessageBox.Show("Неверный номер праздника!",

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

return;

}

}

RebuildBoldedDates();

UpdateSelectedDatesDisplay();

UpdateStatusBar();

SaveHolidaysToFile();

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка удаления праздника: {ex.Message}",

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

}

}

// сохраняем праздники в файл

private void SaveHolidaysToFile()

{

try

{

int totalHolidays = holidays.Values.SelectMany(list => list).Count();

using (StreamWriter writer = new StreamWriter(holidaysFilePath, false, System.Text.Encoding.UTF8))

{

writer.WriteLine($"# Файл праздников - сохранено {DateTime.Now:yyyy-MM-dd HH:mm:ss}");

writer.WriteLine($"# Формат: ДАТА|ПРАЗДНИК1,ПРАЗДНИК2");

foreach (var holiday in holidays.OrderBy(h => h.Key))

{

string holidaysList = string.Join(",", holiday.Value);

writer.WriteLine($"{holiday.Key:yyyy-MM-dd}|{holidaysList}");

}

}

System.Diagnostics.Debug.WriteLine($"Автоматически сохранено {totalHolidays} праздников в файл");

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка автоматического сохранения: {ex.Message}");

}

}

// очищаем все праздники

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

{

try

{

int totalHolidays = holidays.Values.SelectMany(list => list).Count();

if (totalHolidays == 0)

{

MessageBox.Show("Нет праздников для удаления!",

"Предупреждение", MessageBoxButtons.OK, MessageBoxIcon.Warning);

return;

}

var result = MessageBox.Show($"Вы уверены, что хотите удалить все {totalHolidays} праздников?",

"Подтверждение удаления", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

holidays.Clear();

RebuildBoldedDates();

UpdateSelectedDatesDisplay();

UpdateStatusBar();

SaveHolidaysToFile();

MessageBox.Show("Все праздники удалены!",

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

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка очистки праздников: {ex.Message}",

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

}

}

// полностью перестраиваем выделенные даты

private void RebuildBoldedDates()

{

try

{

// убираем все выделения

monthCalendar1.RemoveAllBoldedDates();

// добавляем только даты с праздниками

foreach (var holiday in holidays)

{

monthCalendar1.AddBoldedDate(holiday.Key);

}

monthCalendar1.UpdateBoldedDates();

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка перестройки выделенных дат: {ex.Message}");

}

}

// меняем сколько месяцев показывать

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

{

try

{

int months = (int)numCalendarMonths.Value;

Size calendarSize;

switch (months)

{

case 1:

calendarSize = new Size(1, 1);

break;

case 2:

calendarSize = new Size(2, 1);

break;

case 3:

calendarSize = new Size(3, 1);

break;

case 4:

calendarSize = new Size(2, 2);

break;

case 5:

calendarSize = new Size(3, 2);

break;

case 6:

calendarSize = new Size(3, 2);

break;

default:

calendarSize = new Size(2, 2);

break;

}

monthCalendar1.CalendarDimensions = calendarSize;

MessageBox.Show($"Календарь настроен на отображение {months} месяцев",

"Настройки применены", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка настройки календаря: {ex.Message}",

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

}

}

private void AdjustFormSize(Size calendarSize)

{

try

{

int monthWidth = 200;

int monthHeight = 180;

int calendarPadding = 50;

int calendarWidth = calendarSize.Width \* monthWidth + calendarPadding;

int calendarHeight = calendarSize.Height \* monthHeight + calendarPadding;

int rightPanelWidth = 450;

int bottomPanelHeight = 450;

int statusBarHeight = 26;

int requiredWidth = calendarWidth + rightPanelWidth + 50;

int requiredHeight = calendarHeight + bottomPanelHeight + statusBarHeight + 50;

int formWidth = Math.Max(requiredWidth, 1200);

int formHeight = Math.Max(requiredHeight, 800);

formWidth = Math.Min(formWidth, 1800);

formHeight = Math.Min(formHeight, 1000);

this.ClientSize = new Size(formWidth, formHeight);

int rightPanelX = calendarWidth + 20;

if (rightPanelX < 733) rightPanelX = 733;

groupBoxAppearance.Location = new Point(rightPanelX, groupBoxAppearance.Location.Y);

int bottomPanelY = calendarHeight + 50;

groupBoxControls.Location = new Point(groupBoxControls.Location.X, bottomPanelY);

groupBoxLeapYear.Location = new Point(groupBoxLeapYear.Location.X, bottomPanelY + groupBoxControls.Height + 20);

UpdateStatusBar();

System.Diagnostics.Debug.WriteLine($"Календарь: {calendarWidth}x{calendarHeight}, Форма: {formWidth}x{formHeight}");

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка изменения размера формы: {ex.Message}");

}

}

private void Form1\_Resize(object sender, EventArgs e)

{

try

{

if (this.WindowState == FormWindowState.Normal)

{

AdjustElementPositions();

}

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка обработки изменения размера: {ex.Message}");

}

}

//корректировка позиций элементов

private void AdjustElementPositions()

{

try

{

Size calendarSize = monthCalendar1.CalendarDimensions;

int monthWidth = 200;

int monthHeight = 180;

int calendarPadding = 50;

int calendarWidth = calendarSize.Width \* monthWidth + calendarPadding;

int calendarHeight = calendarSize.Height \* monthHeight + calendarPadding;

int rightPanelX = calendarWidth + 20;

if (rightPanelX < 733) rightPanelX = 733;

if (groupBoxAppearance.Location.X != rightPanelX)

{

groupBoxAppearance.Location = new Point(rightPanelX, groupBoxAppearance.Location.Y);

}

int bottomPanelY = calendarHeight + 50;

if (groupBoxControls.Location.Y != bottomPanelY)

{

groupBoxControls.Location = new Point(groupBoxControls.Location.X, bottomPanelY);

}

int leapYearY = bottomPanelY + groupBoxControls.Height + 20;

if (groupBoxLeapYear.Location.Y != leapYearY)

{

groupBoxLeapYear.Location = new Point(groupBoxLeapYear.Location.X, leapYearY);

}

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка корректировки позиций: {ex.Message}");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

try

{

System.Diagnostics.Debug.WriteLine("Форма загружена успешно");

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine($"Ошибка при загрузке формы: {ex.Message}");

}

}

}

}

**Form1.Designer.cs**

using System.Drawing;

using System.Windows.Forms;

namespace Calendar

{

partial class Form1

{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing)

{

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

{

components.Dispose();

}

base.Dispose(disposing);

}

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));

this.monthCalendar1 = new System.Windows.Forms.MonthCalendar();

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

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

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

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

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

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

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

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

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

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

this.datePickerForBold = new System.Windows.Forms.DateTimePicker();

this.datePickerStart = new System.Windows.Forms.DateTimePicker();

this.datePickerEnd = new System.Windows.Forms.DateTimePicker();

this.chkShowToday = new System.Windows.Forms.CheckBox();

this.chkShowTodayCircle = new System.Windows.Forms.CheckBox();

this.chkShowWeekNumbers = new System.Windows.Forms.CheckBox();

this.groupBoxEvents = new System.Windows.Forms.GroupBox();

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

this.groupBoxControls = new System.Windows.Forms.GroupBox();

this.label3 = new System.Windows.Forms.Label();

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

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

this.numHolidayDays = new System.Windows.Forms.NumericUpDown();

this.label6 = new System.Windows.Forms.Label();

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

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

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

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

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

this.groupBoxSelection = new System.Windows.Forms.GroupBox();

this.label1 = new System.Windows.Forms.Label();

this.label2 = new System.Windows.Forms.Label();

this.label4 = new System.Windows.Forms.Label();

this.numMaxSelection = new System.Windows.Forms.NumericUpDown();

this.groupBoxAppearance = new System.Windows.Forms.GroupBox();

this.label5 = new System.Windows.Forms.Label();

this.cmbFirstDayOfWeek = new System.Windows.Forms.ComboBox();

this.numCalendarMonths = new System.Windows.Forms.NumericUpDown();

this.label7 = new System.Windows.Forms.Label();

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

this.groupBoxLeapYear = new System.Windows.Forms.GroupBox();

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

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

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

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

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

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

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

this.statusStrip1 = new System.Windows.Forms.StatusStrip();

this.toolStripStatusLabel1 = new System.Windows.Forms.ToolStripStatusLabel();

this.toolStripStatusLabel2 = new System.Windows.Forms.ToolStripStatusLabel();

this.toolStripStatusLabel3 = new System.Windows.Forms.ToolStripStatusLabel();

this.groupBoxEvents.SuspendLayout();

this.groupBoxControls.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numHolidayDays)).BeginInit();

this.groupBoxSelection.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numMaxSelection)).BeginInit();

this.groupBoxAppearance.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numCalendarMonths)).BeginInit();

this.groupBoxLeapYear.SuspendLayout();

this.statusStrip1.SuspendLayout();

this.SuspendLayout();

//

// monthCalendar1

//

this.monthCalendar1.CalendarDimensions = new System.Drawing.Size(2, 2);

this.monthCalendar1.Location = new System.Drawing.Point(13, 25);

this.monthCalendar1.Margin = new System.Windows.Forms.Padding(12, 11, 12, 11);

this.monthCalendar1.Name = "monthCalendar1";

this.monthCalendar1.TabIndex = 0;

//

// lblSelectedDate

//

this.lblSelectedDate.Location = new System.Drawing.Point(13, 31);

this.lblSelectedDate.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.lblSelectedDate.Name = "lblSelectedDate";

this.lblSelectedDate.Size = new System.Drawing.Size(373, 25);

this.lblSelectedDate.TabIndex = 0;

this.lblSelectedDate.Text = "Выбранная дата: не выбрана";

//

// lblDateChanged

//

this.lblDateChanged.Location = new System.Drawing.Point(13, 80);

this.lblDateChanged.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.lblDateChanged.Name = "lblDateChanged";

this.lblDateChanged.Size = new System.Drawing.Size(373, 49);

this.lblDateChanged.TabIndex = 1;

this.lblDateChanged.Text = "Последнее изменение: не было";

//

// lblSelectedCount

//

this.lblSelectedCount.Location = new System.Drawing.Point(13, 55);

this.lblSelectedCount.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.lblSelectedCount.Name = "lblSelectedCount";

this.lblSelectedCount.Size = new System.Drawing.Size(373, 25);

this.lblSelectedCount.TabIndex = 2;

this.lblSelectedCount.Text = "Количество дат: 0";

//

// btnAddBoldDate

//

this.btnAddBoldDate.Location = new System.Drawing.Point(227, 49);

this.btnAddBoldDate.Margin = new System.Windows.Forms.Padding(4);

this.btnAddBoldDate.Name = "btnAddBoldDate";

this.btnAddBoldDate.Size = new System.Drawing.Size(133, 31);

this.btnAddBoldDate.TabIndex = 2;

this.btnAddBoldDate.Text = "Выделить";

this.btnAddBoldDate.UseVisualStyleBackColor = true;

//

// btnRemoveBoldDate

//

this.btnRemoveBoldDate.Location = new System.Drawing.Point(373, 49);

this.btnRemoveBoldDate.Margin = new System.Windows.Forms.Padding(4);

this.btnRemoveBoldDate.Name = "btnRemoveBoldDate";

this.btnRemoveBoldDate.Size = new System.Drawing.Size(133, 31);

this.btnRemoveBoldDate.TabIndex = 3;

this.btnRemoveBoldDate.Text = "Убрать";

this.btnRemoveBoldDate.UseVisualStyleBackColor = true;

//

// btnSetColors

//

this.btnSetColors.Location = new System.Drawing.Point(13, 31);

this.btnSetColors.Margin = new System.Windows.Forms.Padding(4);

this.btnSetColors.Name = "btnSetColors";

this.btnSetColors.Size = new System.Drawing.Size(173, 31);

this.btnSetColors.TabIndex = 0;

this.btnSetColors.Text = "Изменить цвета";

this.btnSetColors.UseVisualStyleBackColor = true;

//

// btnResetColors

//

this.btnResetColors.Location = new System.Drawing.Point(200, 31);

this.btnResetColors.Margin = new System.Windows.Forms.Padding(4);

this.btnResetColors.Name = "btnResetColors";

this.btnResetColors.Size = new System.Drawing.Size(173, 31);

this.btnResetColors.TabIndex = 1;

this.btnResetColors.Text = "Сбросить цвета";

this.btnResetColors.UseVisualStyleBackColor = true;

//

// btnClearSelection

//

this.btnClearSelection.Location = new System.Drawing.Point(200, 86);

this.btnClearSelection.Margin = new System.Windows.Forms.Padding(4);

this.btnClearSelection.Name = "btnClearSelection";

this.btnClearSelection.Size = new System.Drawing.Size(173, 31);

this.btnClearSelection.TabIndex = 5;

this.btnClearSelection.Text = "Очистить выбор";

this.btnClearSelection.UseVisualStyleBackColor = true;

//

// btnSelectRange

//

this.btnSelectRange.Location = new System.Drawing.Point(13, 86);

this.btnSelectRange.Margin = new System.Windows.Forms.Padding(4);

this.btnSelectRange.Name = "btnSelectRange";

this.btnSelectRange.Size = new System.Drawing.Size(173, 31);

this.btnSelectRange.TabIndex = 4;

this.btnSelectRange.Text = "Выбрать период";

this.btnSelectRange.UseVisualStyleBackColor = true;

//

// btnAddHoliday

//

this.btnAddHoliday.Location = new System.Drawing.Point(227, 92);

this.btnAddHoliday.Margin = new System.Windows.Forms.Padding(4);

this.btnAddHoliday.Name = "btnAddHoliday";

this.btnAddHoliday.Size = new System.Drawing.Size(173, 31);

this.btnAddHoliday.TabIndex = 5;

this.btnAddHoliday.Text = "Добавить праздник";

this.btnAddHoliday.UseVisualStyleBackColor = true;

//

// datePickerForBold

//

this.datePickerForBold.Location = new System.Drawing.Point(13, 49);

this.datePickerForBold.Margin = new System.Windows.Forms.Padding(4);

this.datePickerForBold.Name = "datePickerForBold";

this.datePickerForBold.Size = new System.Drawing.Size(199, 22);

this.datePickerForBold.TabIndex = 1;

//

// datePickerStart

//

this.datePickerStart.Location = new System.Drawing.Point(13, 49);

this.datePickerStart.Margin = new System.Windows.Forms.Padding(4);

this.datePickerStart.Name = "datePickerStart";

this.datePickerStart.Size = new System.Drawing.Size(172, 22);

this.datePickerStart.TabIndex = 1;

//

// datePickerEnd

//

this.datePickerEnd.Location = new System.Drawing.Point(200, 49);

this.datePickerEnd.Margin = new System.Windows.Forms.Padding(4);

this.datePickerEnd.Name = "datePickerEnd";

this.datePickerEnd.Size = new System.Drawing.Size(172, 22);

this.datePickerEnd.TabIndex = 3;

//

// chkShowToday

//

this.chkShowToday.AutoSize = true;

this.chkShowToday.Location = new System.Drawing.Point(13, 74);

this.chkShowToday.Margin = new System.Windows.Forms.Padding(4);

this.chkShowToday.Name = "chkShowToday";

this.chkShowToday.Size = new System.Drawing.Size(235, 20);

this.chkShowToday.TabIndex = 2;

this.chkShowToday.Text = "Показывать сегодняшнюю дату";

this.chkShowToday.UseVisualStyleBackColor = true;

//

// chkShowTodayCircle

//

this.chkShowTodayCircle.AutoSize = true;

this.chkShowTodayCircle.Location = new System.Drawing.Point(13, 98);

this.chkShowTodayCircle.Margin = new System.Windows.Forms.Padding(4);

this.chkShowTodayCircle.Name = "chkShowTodayCircle";

this.chkShowTodayCircle.Size = new System.Drawing.Size(219, 20);

this.chkShowTodayCircle.TabIndex = 3;

this.chkShowTodayCircle.Text = "Обводить сегодняшнюю дату";

this.chkShowTodayCircle.UseVisualStyleBackColor = true;

//

// chkShowWeekNumbers

//

this.chkShowWeekNumbers.AutoSize = true;

this.chkShowWeekNumbers.Location = new System.Drawing.Point(13, 123);

this.chkShowWeekNumbers.Margin = new System.Windows.Forms.Padding(4);

this.chkShowWeekNumbers.Name = "chkShowWeekNumbers";

this.chkShowWeekNumbers.Size = new System.Drawing.Size(211, 20);

this.chkShowWeekNumbers.TabIndex = 4;

this.chkShowWeekNumbers.Text = "Показывать номера недель";

this.chkShowWeekNumbers.UseVisualStyleBackColor = true;

//

// groupBoxEvents

//

this.groupBoxEvents.Controls.Add(this.lblSelectedDate);

this.groupBoxEvents.Controls.Add(this.lblDateChanged);

this.groupBoxEvents.Controls.Add(this.lblSelectedCount);

this.groupBoxEvents.Controls.Add(this.lstSelectedDates);

this.groupBoxEvents.Location = new System.Drawing.Point(733, 25);

this.groupBoxEvents.Margin = new System.Windows.Forms.Padding(4);

this.groupBoxEvents.Name = "groupBoxEvents";

this.groupBoxEvents.Padding = new System.Windows.Forms.Padding(4);

this.groupBoxEvents.Size = new System.Drawing.Size(400, 308);

this.groupBoxEvents.TabIndex = 1;

this.groupBoxEvents.TabStop = false;

this.groupBoxEvents.Text = "Выбранные даты";

//

// lstSelectedDates

//

this.lstSelectedDates.ItemHeight = 16;

this.lstSelectedDates.Location = new System.Drawing.Point(13, 135);

this.lstSelectedDates.Margin = new System.Windows.Forms.Padding(4);

this.lstSelectedDates.Name = "lstSelectedDates";

this.lstSelectedDates.Size = new System.Drawing.Size(372, 148);

this.lstSelectedDates.TabIndex = 3;

//

// groupBoxControls

//

this.groupBoxControls.Controls.Add(this.label3);

this.groupBoxControls.Controls.Add(this.datePickerForBold);

this.groupBoxControls.Controls.Add(this.btnAddBoldDate);

this.groupBoxControls.Controls.Add(this.btnRemoveBoldDate);

this.groupBoxControls.Controls.Add(this.txtHolidayName);

this.groupBoxControls.Controls.Add(this.btnAddHoliday);

this.groupBoxControls.Controls.Add(this.btnAddMultiDayHoliday);

this.groupBoxControls.Controls.Add(this.numHolidayDays);

this.groupBoxControls.Controls.Add(this.label6);

this.groupBoxControls.Controls.Add(this.btnSaveHolidays);

this.groupBoxControls.Controls.Add(this.btnLoadHolidays);

this.groupBoxControls.Controls.Add(this.btnShowHolidayFileInfo);

this.groupBoxControls.Controls.Add(this.btnRemoveHoliday);

this.groupBoxControls.Controls.Add(this.btnClearAllHolidays);

this.groupBoxControls.Location = new System.Drawing.Point(13, 431);

this.groupBoxControls.Margin = new System.Windows.Forms.Padding(4);

this.groupBoxControls.Name = "groupBoxControls";

this.groupBoxControls.Padding = new System.Windows.Forms.Padding(4);

this.groupBoxControls.Size = new System.Drawing.Size(707, 240);

this.groupBoxControls.TabIndex = 3;

this.groupBoxControls.TabStop = false;

this.groupBoxControls.Text = "Управление выделенными датами и праздниками";

//

// label3

//

this.label3.Location = new System.Drawing.Point(13, 31);

this.label3.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label3.Name = "label3";

this.label3.Size = new System.Drawing.Size(133, 18);

this.label3.TabIndex = 0;

this.label3.Text = "Выберите дату:";

//

// txtHolidayName

//

this.txtHolidayName.Location = new System.Drawing.Point(13, 92);

this.txtHolidayName.Margin = new System.Windows.Forms.Padding(4);

this.txtHolidayName.Name = "txtHolidayName";

this.txtHolidayName.Size = new System.Drawing.Size(199, 22);

this.txtHolidayName.TabIndex = 4;

//

// btnAddMultiDayHoliday

//

this.btnAddMultiDayHoliday.Location = new System.Drawing.Point(227, 130);

this.btnAddMultiDayHoliday.Margin = new System.Windows.Forms.Padding(4);

this.btnAddMultiDayHoliday.Name = "btnAddMultiDayHoliday";

this.btnAddMultiDayHoliday.Size = new System.Drawing.Size(173, 31);

this.btnAddMultiDayHoliday.TabIndex = 6;

this.btnAddMultiDayHoliday.Text = "Многодневный праздник";

this.btnAddMultiDayHoliday.UseVisualStyleBackColor = true;

//

// numHolidayDays

//

this.numHolidayDays.Location = new System.Drawing.Point(13, 130);

this.numHolidayDays.Margin = new System.Windows.Forms.Padding(4);

this.numHolidayDays.Maximum = new decimal(new int[] {

30,

0,

0,

0});

this.numHolidayDays.Minimum = new decimal(new int[] {

2,

0,

0,

0});

this.numHolidayDays.Name = "numHolidayDays";

this.numHolidayDays.Size = new System.Drawing.Size(80, 22);

this.numHolidayDays.TabIndex = 7;

this.numHolidayDays.Value = new decimal(new int[] {

3,

0,

0,

0});

//

// label6

//

this.label6.AutoSize = true;

this.label6.Location = new System.Drawing.Point(100, 134);

this.label6.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label6.Name = "label6";

this.label6.Size = new System.Drawing.Size(89, 16);

this.label6.TabIndex = 8;

this.label6.Text = "дней подряд";

//

// btnSaveHolidays

//

this.btnSaveHolidays.Location = new System.Drawing.Point(427, 130);

this.btnSaveHolidays.Margin = new System.Windows.Forms.Padding(4);

this.btnSaveHolidays.Name = "btnSaveHolidays";

this.btnSaveHolidays.Size = new System.Drawing.Size(133, 31);

this.btnSaveHolidays.TabIndex = 9;

this.btnSaveHolidays.Text = "Сохранить";

this.btnSaveHolidays.UseVisualStyleBackColor = true;

//

// btnLoadHolidays

//

this.btnLoadHolidays.Location = new System.Drawing.Point(573, 130);

this.btnLoadHolidays.Margin = new System.Windows.Forms.Padding(4);

this.btnLoadHolidays.Name = "btnLoadHolidays";

this.btnLoadHolidays.Size = new System.Drawing.Size(133, 31);

this.btnLoadHolidays.TabIndex = 10;

this.btnLoadHolidays.Text = "Загрузить";

this.btnLoadHolidays.UseVisualStyleBackColor = true;

//

// btnShowHolidayFileInfo

//

this.btnShowHolidayFileInfo.Location = new System.Drawing.Point(13, 170);

this.btnShowHolidayFileInfo.Margin = new System.Windows.Forms.Padding(4);

this.btnShowHolidayFileInfo.Name = "btnShowHolidayFileInfo";

this.btnShowHolidayFileInfo.Size = new System.Drawing.Size(200, 31);

this.btnShowHolidayFileInfo.TabIndex = 11;

this.btnShowHolidayFileInfo.Text = "Информация о файле";

this.btnShowHolidayFileInfo.UseVisualStyleBackColor = true;

//

// btnRemoveHoliday

//

this.btnRemoveHoliday.Location = new System.Drawing.Point(227, 170);

this.btnRemoveHoliday.Margin = new System.Windows.Forms.Padding(4);

this.btnRemoveHoliday.Name = "btnRemoveHoliday";

this.btnRemoveHoliday.Size = new System.Drawing.Size(200, 31);

this.btnRemoveHoliday.TabIndex = 12;

this.btnRemoveHoliday.Text = "Удалить праздник";

this.btnRemoveHoliday.UseVisualStyleBackColor = true;

//

// btnClearAllHolidays

//

this.btnClearAllHolidays.Location = new System.Drawing.Point(441, 170);

this.btnClearAllHolidays.Margin = new System.Windows.Forms.Padding(4);

this.btnClearAllHolidays.Name = "btnClearAllHolidays";

this.btnClearAllHolidays.Size = new System.Drawing.Size(200, 31);

this.btnClearAllHolidays.TabIndex = 13;

this.btnClearAllHolidays.Text = "Очистить все праздники";

this.btnClearAllHolidays.UseVisualStyleBackColor = true;

//

// groupBoxSelection

//

this.groupBoxSelection.Controls.Add(this.label1);

this.groupBoxSelection.Controls.Add(this.label2);

this.groupBoxSelection.Controls.Add(this.datePickerStart);

this.groupBoxSelection.Controls.Add(this.datePickerEnd);

this.groupBoxSelection.Controls.Add(this.btnSelectRange);

this.groupBoxSelection.Controls.Add(this.btnClearSelection);

this.groupBoxSelection.Controls.Add(this.label4);

this.groupBoxSelection.Controls.Add(this.numMaxSelection);

this.groupBoxSelection.Location = new System.Drawing.Point(733, 345);

this.groupBoxSelection.Margin = new System.Windows.Forms.Padding(4);

this.groupBoxSelection.Name = "groupBoxSelection";

this.groupBoxSelection.Padding = new System.Windows.Forms.Padding(4);

this.groupBoxSelection.Size = new System.Drawing.Size(400, 185);

this.groupBoxSelection.TabIndex = 2;

this.groupBoxSelection.TabStop = false;

this.groupBoxSelection.Text = "Управление выбором";

//

// label1

//

this.label1.Location = new System.Drawing.Point(13, 31);

this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(107, 18);

this.label1.TabIndex = 0;

this.label1.Text = "Дата начала:";

//

// label2

//

this.label2.Location = new System.Drawing.Point(200, 31);

this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(107, 18);

this.label2.TabIndex = 2;

this.label2.Text = "Дата конца:";

//

// label4

//

this.label4.Location = new System.Drawing.Point(13, 129);

this.label4.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label4.Name = "label4";

this.label4.Size = new System.Drawing.Size(200, 18);

this.label4.TabIndex = 6;

this.label4.Text = "Максимум дней для выбора:";

//

// numMaxSelection

//

this.numMaxSelection.Location = new System.Drawing.Point(213, 123);

this.numMaxSelection.Margin = new System.Windows.Forms.Padding(4);

this.numMaxSelection.Maximum = new decimal(new int[] {

365,

0,

0,

0});

this.numMaxSelection.Minimum = new decimal(new int[] {

1,

0,

0,

0});

this.numMaxSelection.Name = "numMaxSelection";

this.numMaxSelection.Size = new System.Drawing.Size(80, 22);

this.numMaxSelection.TabIndex = 7;

this.numMaxSelection.Value = new decimal(new int[] {

30,

0,

0,

0});

//

// groupBoxAppearance

//

this.groupBoxAppearance.Controls.Add(this.btnSetColors);

this.groupBoxAppearance.Controls.Add(this.btnResetColors);

this.groupBoxAppearance.Controls.Add(this.chkShowToday);

this.groupBoxAppearance.Controls.Add(this.chkShowTodayCircle);

this.groupBoxAppearance.Controls.Add(this.chkShowWeekNumbers);

this.groupBoxAppearance.Controls.Add(this.label5);

this.groupBoxAppearance.Controls.Add(this.cmbFirstDayOfWeek);

this.groupBoxAppearance.Controls.Add(this.numCalendarMonths);

this.groupBoxAppearance.Controls.Add(this.label7);

this.groupBoxAppearance.Controls.Add(this.btnSetCalendarSize);

this.groupBoxAppearance.Location = new System.Drawing.Point(733, 542);

this.groupBoxAppearance.Margin = new System.Windows.Forms.Padding(4);

this.groupBoxAppearance.Name = "groupBoxAppearance";

this.groupBoxAppearance.Padding = new System.Windows.Forms.Padding(4);

this.groupBoxAppearance.Size = new System.Drawing.Size(400, 317);

this.groupBoxAppearance.TabIndex = 4;

this.groupBoxAppearance.TabStop = false;

this.groupBoxAppearance.Text = "Внешний вид";

//

// label5

//

this.label5.Location = new System.Drawing.Point(13, 154);

this.label5.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label5.Name = "label5";

this.label5.Size = new System.Drawing.Size(160, 18);

this.label5.TabIndex = 5;

this.label5.Text = "Первый день недели:";

//

// cmbFirstDayOfWeek

//

this.cmbFirstDayOfWeek.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;

this.cmbFirstDayOfWeek.Location = new System.Drawing.Point(13, 178);

this.cmbFirstDayOfWeek.Margin = new System.Windows.Forms.Padding(4);

this.cmbFirstDayOfWeek.Name = "cmbFirstDayOfWeek";

this.cmbFirstDayOfWeek.Size = new System.Drawing.Size(199, 24);

this.cmbFirstDayOfWeek.TabIndex = 6;

//

// numCalendarMonths

//

this.numCalendarMonths.Location = new System.Drawing.Point(13, 210);

this.numCalendarMonths.Margin = new System.Windows.Forms.Padding(4);

this.numCalendarMonths.Maximum = new decimal(new int[] {

6,

0,

0,

0});

this.numCalendarMonths.Minimum = new decimal(new int[] {

1,

0,

0,

0});

this.numCalendarMonths.Name = "numCalendarMonths";

this.numCalendarMonths.Size = new System.Drawing.Size(80, 22);

this.numCalendarMonths.TabIndex = 7;

this.numCalendarMonths.Value = new decimal(new int[] {

4,

0,

0,

0});

//

// label7

//

this.label7.AutoSize = true;

this.label7.Location = new System.Drawing.Point(100, 214);

this.label7.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label7.Name = "label7";

this.label7.Size = new System.Drawing.Size(147, 16);

this.label7.TabIndex = 8;

this.label7.Text = "месяцев в календаре";

//

// btnSetCalendarSize

//

this.btnSetCalendarSize.Location = new System.Drawing.Point(13, 240);

this.btnSetCalendarSize.Margin = new System.Windows.Forms.Padding(4);

this.btnSetCalendarSize.Name = "btnSetCalendarSize";

this.btnSetCalendarSize.Size = new System.Drawing.Size(199, 31);

this.btnSetCalendarSize.TabIndex = 9;

this.btnSetCalendarSize.Text = "Применить";

this.btnSetCalendarSize.UseVisualStyleBackColor = true;

//

// groupBoxLeapYear

//

this.groupBoxLeapYear.Controls.Add(this.lblLeapYearInfo);

this.groupBoxLeapYear.Controls.Add(this.txtYearInput);

this.groupBoxLeapYear.Controls.Add(this.btnCheckLeapYear);

this.groupBoxLeapYear.Controls.Add(this.btnGoToToday);

this.groupBoxLeapYear.Controls.Add(this.lblCurrentTime);

this.groupBoxLeapYear.Controls.Add(this.btnCopyDate);

this.groupBoxLeapYear.Controls.Add(this.lblDaysUntil);

this.groupBoxLeapYear.Location = new System.Drawing.Point(13, 679);

this.groupBoxLeapYear.Margin = new System.Windows.Forms.Padding(4);

this.groupBoxLeapYear.Name = "groupBoxLeapYear";

this.groupBoxLeapYear.Padding = new System.Windows.Forms.Padding(4);

this.groupBoxLeapYear.Size = new System.Drawing.Size(712, 180);

this.groupBoxLeapYear.TabIndex = 5;

this.groupBoxLeapYear.TabStop = false;

this.groupBoxLeapYear.Text = "Информация о високосном годе";

//

// lblLeapYearInfo

//

this.lblLeapYearInfo.Location = new System.Drawing.Point(13, 49);

this.lblLeapYearInfo.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.lblLeapYearInfo.Name = "lblLeapYearInfo";

this.lblLeapYearInfo.Size = new System.Drawing.Size(400, 80);

this.lblLeapYearInfo.TabIndex = 0;

this.lblLeapYearInfo.Text = "Информация о текущем годе";

//

// txtYearInput

//

this.txtYearInput.Location = new System.Drawing.Point(427, 31);

this.txtYearInput.Margin = new System.Windows.Forms.Padding(4);

this.txtYearInput.Name = "txtYearInput";

this.txtYearInput.Size = new System.Drawing.Size(132, 22);

this.txtYearInput.TabIndex = 1;

this.txtYearInput.Text = "2025";

//

// btnCheckLeapYear

//

this.btnCheckLeapYear.Location = new System.Drawing.Point(427, 68);

this.btnCheckLeapYear.Margin = new System.Windows.Forms.Padding(4);

this.btnCheckLeapYear.Name = "btnCheckLeapYear";

this.btnCheckLeapYear.Size = new System.Drawing.Size(133, 31);

this.btnCheckLeapYear.TabIndex = 2;

this.btnCheckLeapYear.Text = "Проверить год";

this.btnCheckLeapYear.UseVisualStyleBackColor = true;

//

// btnGoToToday

//

this.btnGoToToday.Location = new System.Drawing.Point(573, 68);

this.btnGoToToday.Margin = new System.Windows.Forms.Padding(4);

this.btnGoToToday.Name = "btnGoToToday";

this.btnGoToToday.Size = new System.Drawing.Size(133, 31);

this.btnGoToToday.TabIndex = 3;

this.btnGoToToday.Text = "Сегодня";

this.btnGoToToday.UseVisualStyleBackColor = true;

//

// lblCurrentTime

//

this.lblCurrentTime.AutoSize = true;

this.lblCurrentTime.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Bold);

this.lblCurrentTime.ForeColor = System.Drawing.Color.DarkBlue;

this.lblCurrentTime.Location = new System.Drawing.Point(13, 106);

this.lblCurrentTime.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.lblCurrentTime.Name = "lblCurrentTime";

this.lblCurrentTime.Size = new System.Drawing.Size(150, 20);

this.lblCurrentTime.TabIndex = 4;

this.lblCurrentTime.Text = "Текущее время";

//

// btnCopyDate

//

this.btnCopyDate.Location = new System.Drawing.Point(573, 31);

this.btnCopyDate.Margin = new System.Windows.Forms.Padding(4);

this.btnCopyDate.Name = "btnCopyDate";

this.btnCopyDate.Size = new System.Drawing.Size(133, 31);

this.btnCopyDate.TabIndex = 5;

this.btnCopyDate.Text = "Копировать дату";

this.btnCopyDate.UseVisualStyleBackColor = true;

//

// lblDaysUntil

//

this.lblDaysUntil.AutoSize = true;

this.lblDaysUntil.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F);

this.lblDaysUntil.ForeColor = System.Drawing.Color.DarkGreen;

this.lblDaysUntil.Location = new System.Drawing.Point(13, 130);

this.lblDaysUntil.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.lblDaysUntil.Name = "lblDaysUntil";

this.lblDaysUntil.Size = new System.Drawing.Size(186, 18);

this.lblDaysUntil.TabIndex = 6;

this.lblDaysUntil.Text = "Дней до выбранной даты";

//

// statusStrip1

//

this.statusStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);

this.statusStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.toolStripStatusLabel1,

this.toolStripStatusLabel2,

this.toolStripStatusLabel3});

this.statusStrip1.Location = new System.Drawing.Point(0, 863);

this.statusStrip1.Name = "statusStrip1";

this.statusStrip1.Size = new System.Drawing.Size(1242, 26);

this.statusStrip1.TabIndex = 8;

this.statusStrip1.Text = "statusStrip1";

//

// toolStripStatusLabel1

//

this.toolStripStatusLabel1.Name = "toolStripStatusLabel1";

this.toolStripStatusLabel1.Size = new System.Drawing.Size(98, 20);

this.toolStripStatusLabel1.Text = "Статус: Готов";

//

// toolStripStatusLabel2

//

this.toolStripStatusLabel2.Name = "toolStripStatusLabel2";

this.toolStripStatusLabel2.Size = new System.Drawing.Size(106, 20);

this.toolStripStatusLabel2.Text = "Тема: Светлая";

//

// toolStripStatusLabel3

//

this.toolStripStatusLabel3.Name = "toolStripStatusLabel3";

this.toolStripStatusLabel3.Size = new System.Drawing.Size(109, 20);

this.toolStripStatusLabel3.Text = "Праздников: 5";

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

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

this.ClientSize = new System.Drawing.Size(1242, 889);

this.Controls.Add(this.statusStrip1);

this.Controls.Add(this.monthCalendar1);

this.Controls.Add(this.groupBoxEvents);

this.Controls.Add(this.groupBoxSelection);

this.Controls.Add(this.groupBoxControls);

this.Controls.Add(this.groupBoxAppearance);

this.Controls.Add(this.groupBoxLeapYear);

this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));

this.Margin = new System.Windows.Forms.Padding(4);

this.Name = "Form1";

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

this.Text = "Календарь";

this.Load += new System.EventHandler(this.Form1\_Load);

this.groupBoxEvents.ResumeLayout(false);

this.groupBoxControls.ResumeLayout(false);

this.groupBoxControls.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.numHolidayDays)).EndInit();

this.groupBoxSelection.ResumeLayout(false);

((System.ComponentModel.ISupportInitialize)(this.numMaxSelection)).EndInit();

this.groupBoxAppearance.ResumeLayout(false);

this.groupBoxAppearance.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.numCalendarMonths)).EndInit();

this.groupBoxLeapYear.ResumeLayout(false);

this.groupBoxLeapYear.PerformLayout();

this.statusStrip1.ResumeLayout(false);

this.statusStrip1.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

//#endregion

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Calendar

{

internal static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

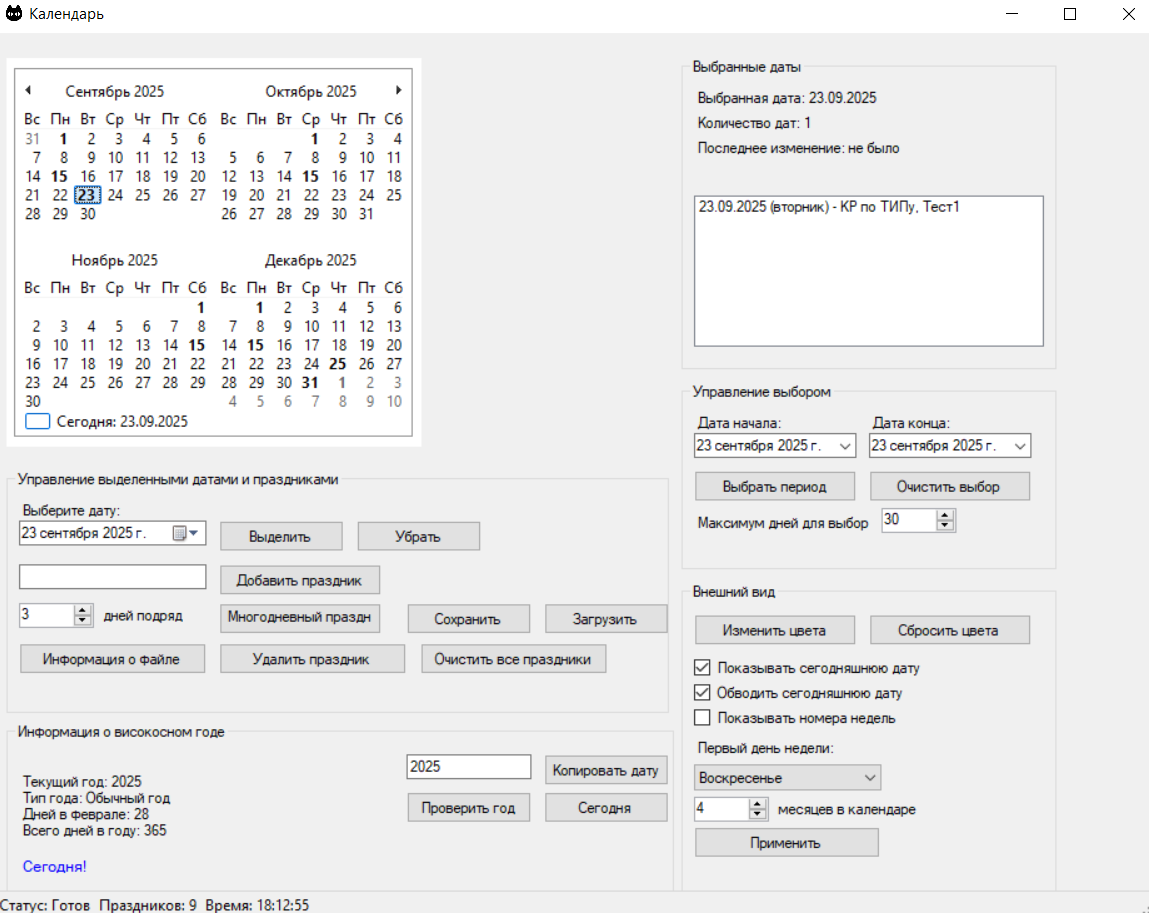
Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Интерфейс** 

**Рекомендации**

1. Добавить выделение выходных (сб и вс)
2. Сделать сохранение праздников в json
3. Добавить адаптивный интерфейс (при изменении окна и тп)

**Ссылка на гит**

<https://github.com/A24-0/Calendar_WF.git>