Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 15

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Работа с календарём»

Выполнил: ст. гр. ТУУ-111

Грачева Н.С.

Вариант №5

Проверил: к.т.н., доц. Сафронов А.И.

Москва – 2024 г.

## Цель работы

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 индивидуального задания), а также справочной информации для демонстрации в разделе «Помощь».

## Формулировка задачи

**Вариант 05.** На правом (или нижнем) календаре выполнить маркировку всех выходных дней указанного календарного года с учётом праздничных дней, принятых постановлением Правительства Российской Федерации на текущий календарный год.

## Составление диаграммы классов, входящих в состав решения.

Calendar

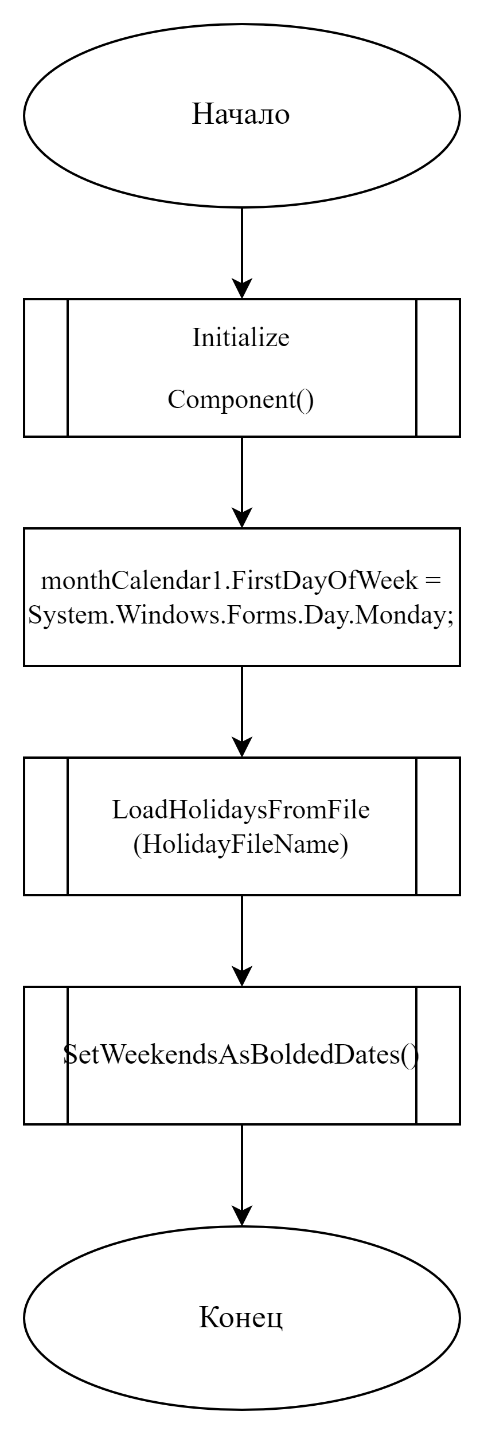
private void monthCalendar1\_DateChanged(object sender, DateRangeEventArgs e)

## Сети Петри запрограммированного технологического процесса.



Рисунок 1 – Сеть Петри

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



1.

2.

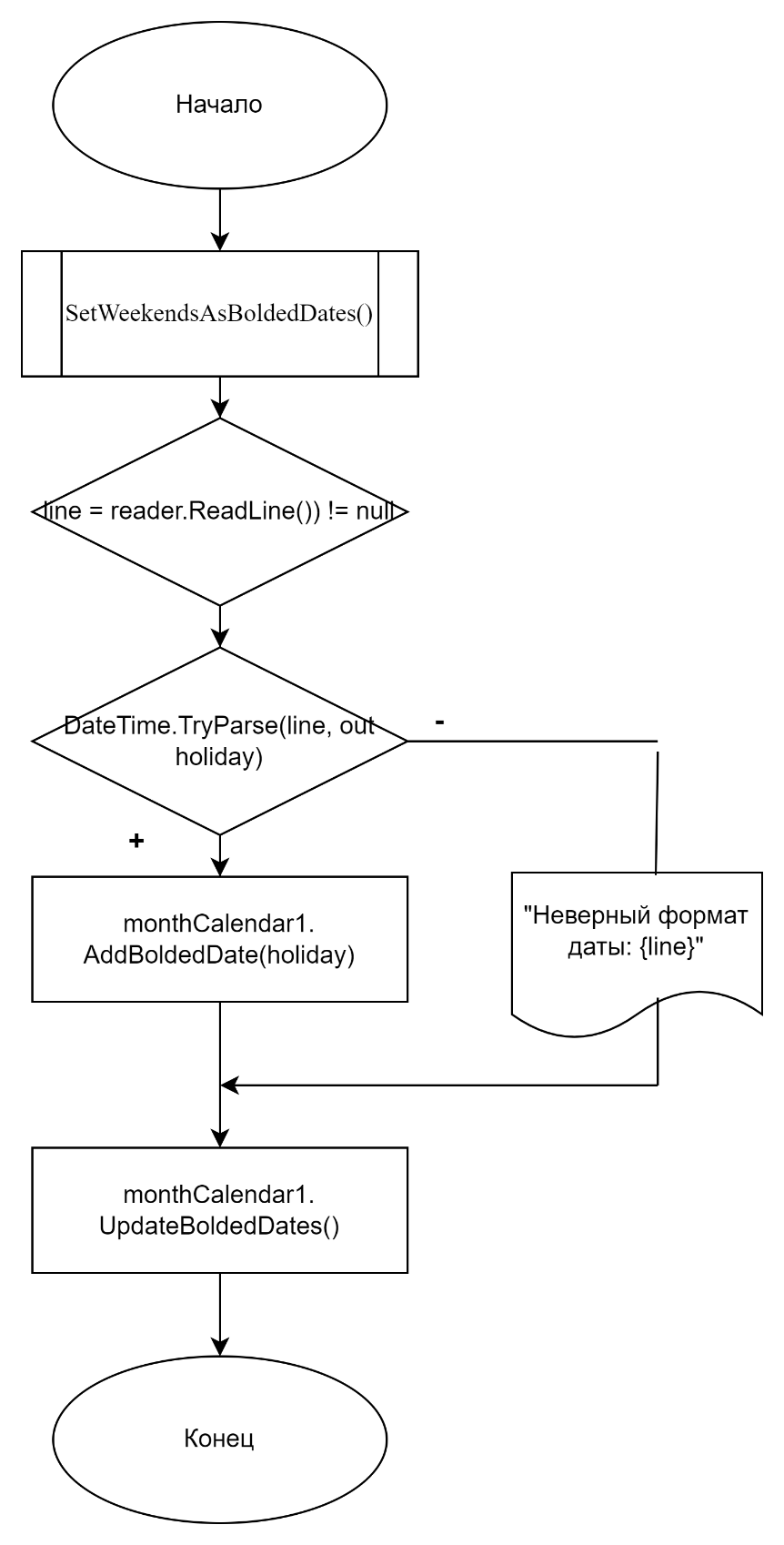
3.

4.

5.

6.

Рисунок 2 – Блок-схема основной программы



4.1

4.2

4.3

4.4

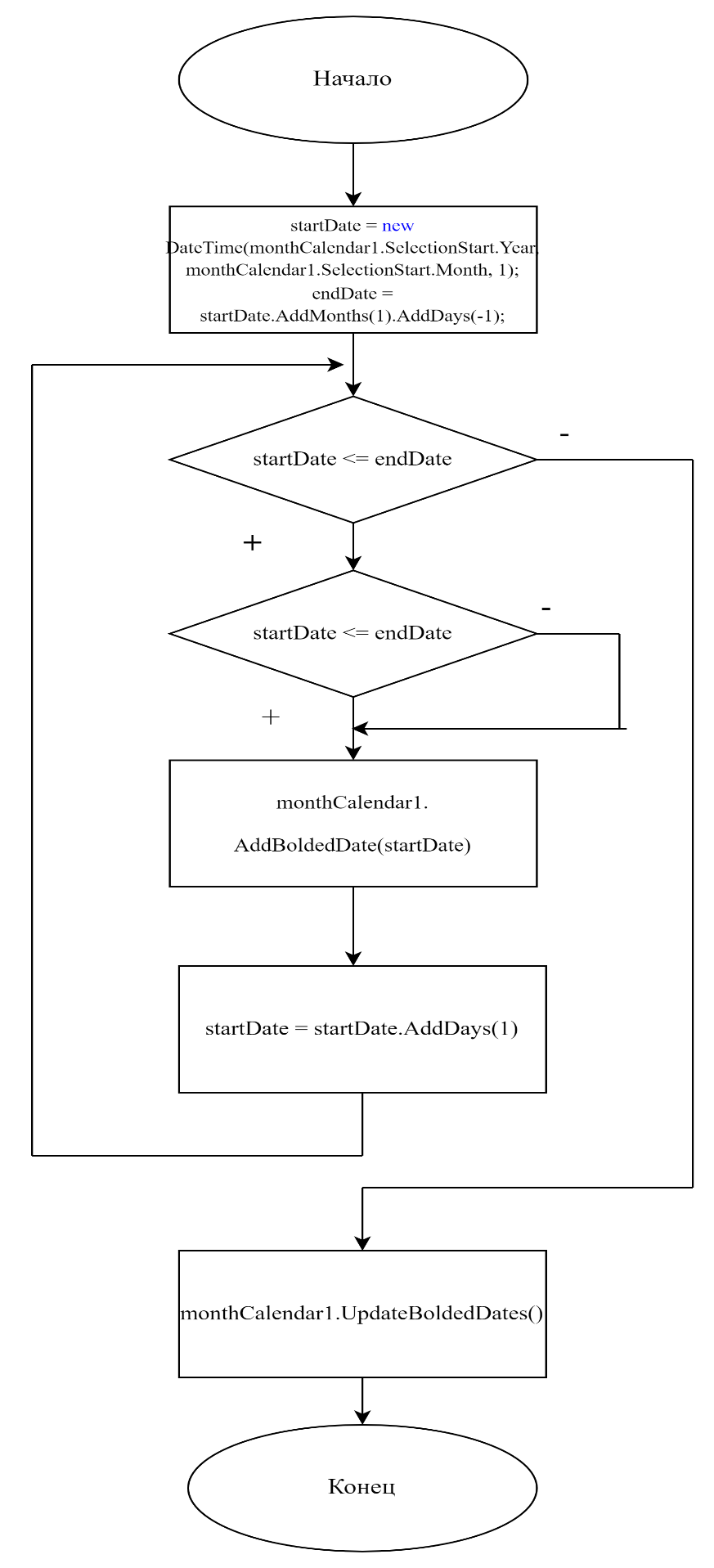
4.5

4.6

4.7

4.8

Рисунок 3 – Блок-схема метода LoadHolidaysFromFile(string fileName)



5.1

5.2

5.3

5.4

5.5

5.6

5.7

5.8

Рисунок 4 – Блок-схема метода SetWeekendsAsBoldedDates()

**6. Подбор тестовых примеров.**

Пример №1

Праздничные и выходные дни в мае

Пример №2

Праздничные и выходные дни в ноябре

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

using System;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace A\_N\_2

{

public partial class Calendar : Form

{

private const string HolidayFileName = "holidays.txt";

public Calendar()

{

InitializeComponent();

//первый день недели - понедельник

monthCalendar1.FirstDayOfWeek = System.Windows.Forms.Day.Monday;

LoadHolidaysFromFile(HolidayFileName);

//отмечаем выходные дни полужирным шрифтом

SetWeekendsAsBoldedDates();

}

private void LoadHolidaysFromFile(string fileName)

{

monthCalendar1.RemoveAllBoldedDates();

DateTime holiday;

//отмечаем выходные дни

SetWeekendsAsBoldedDates();

using (StreamReader reader = new StreamReader(fileName, Encoding.GetEncoding(1251)))

{

string line;

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

{

if (DateTime.TryParse(line, out holiday))

{

// Добавляем праздничный день к выделенным дат

monthCalendar1.AddBoldedDate(holiday);

}

else

{

MessageBox.Show($"Неверный формат даты: {line}", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

monthCalendar1.UpdateBoldedDates();

}

// Метод для выделения выходных дней (субботы и воскресенья)

private void SetWeekendsAsBoldedDates()

{

DateTime startDate = new DateTime(monthCalendar1.SelectionStart.Year, monthCalendar1.SelectionStart.Month, 1);

DateTime endDate = startDate.AddMonths(1).AddDays(-1); // Конец текущего месяца

while (startDate <= endDate)

{

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

{

monthCalendar1.AddBoldedDate(startDate);

}

startDate = startDate.AddDays(1);

}

monthCalendar1.UpdateBoldedDates();

}

private void monthCalendar1\_DateChanged(object sender, DateRangeEventArgs e)

{

//загружаем из файла и отмечаем праздничные дни

LoadHolidaysFromFile(HolidayFileName);

//метод для выделения выходных дней

SetWeekendsAsBoldedDates();

}

}

}

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

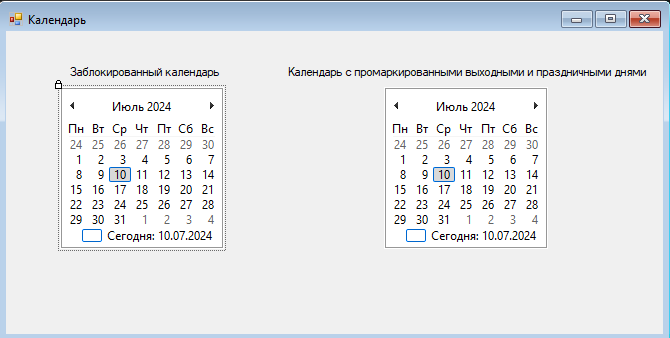


Рисунок 5 – Графический пользовательский интерфейс

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

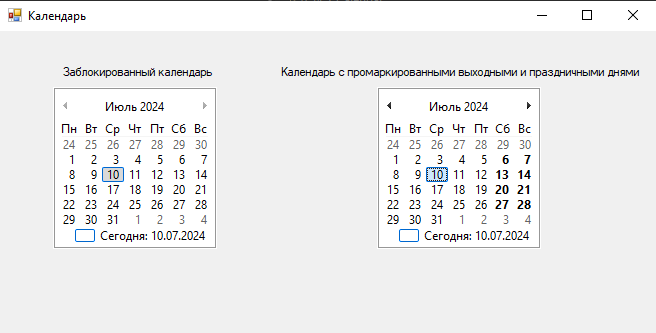


Рисунок 6 – Подтверждение соответствия графического интерфейса

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

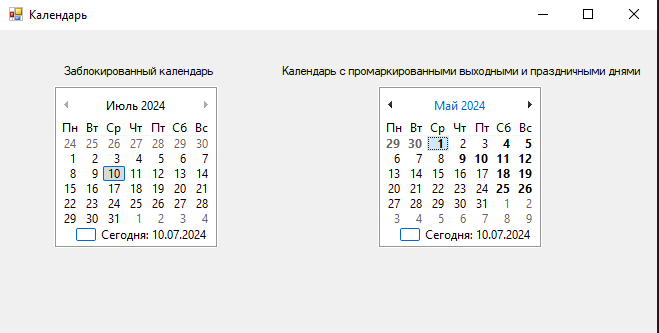


Рисунок 7 – Пример №1

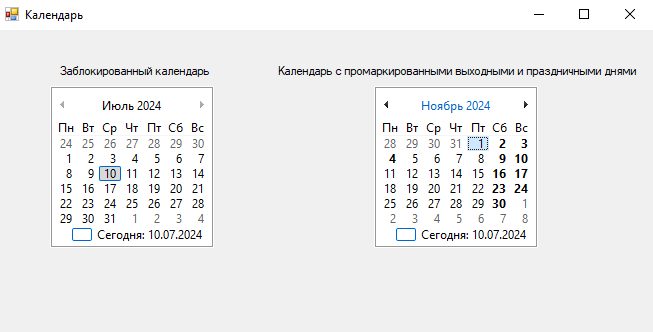


Рисунок 8 – Пример №2

**11. Вывод о проделанной работе**.

С помощью данной работы можно подробно ознакомиться с новым элементом управления MonthCalendar и поработать с ним, улучшить навыки работы в Windows Forms, подробнее изучив свойства элементов управления. Также, для данной работы нужно применить знания построения блок-схем и Сети Петри.