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

Волобуев Е.Д.

Вариант №13

Проверил: к.т.н., доц. Сафронов А.И.

Москва – 2024 г.

1. **Формулировка цели 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».

1. **Описание задачи.**

**Вариант 13.** На левом (или верхнем) календаре выбирается дата. По итогам выбора этой даты некоторый интерфейсный элемент управления должен подсказать номер соответствующей недели в учебном семестре РУТ (МИИТ). Рассуждение по семестрам вести с учётом двух опорных дат: 1 сентября и 7 февраля. Пренебречь проверкой даты на выходные дни, выпадающие на эти даты. В скобках указывать номер этой недели в календарном году и контролировать совпадение по чётности/нечётности рассматриваемых недель.

1. **Составление диаграммы классов, входящих в состав решения.**

Kalendar

- private void KalendarDateTimePicker\_ValueChanged(object sender, EventArgs e)

- private int GetWeekNumber(DateTime selectedDate, DateTime semesterStart, DateTime semesterEnd)

- private int GetCalendarWeekNumber(DateTime date)

Рисунок 1 – Диаграмма классов, входящих в состав решения

1. **Сети Петри запрограммированного технологического процесса.**

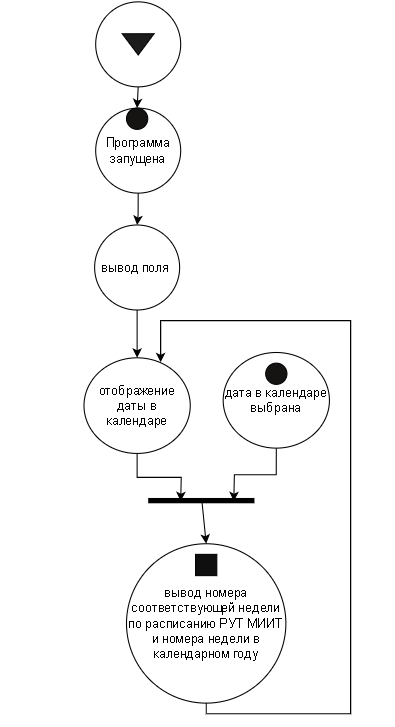


Рисунок 2 – Сеть Петри запрограммированного технологического процесса

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

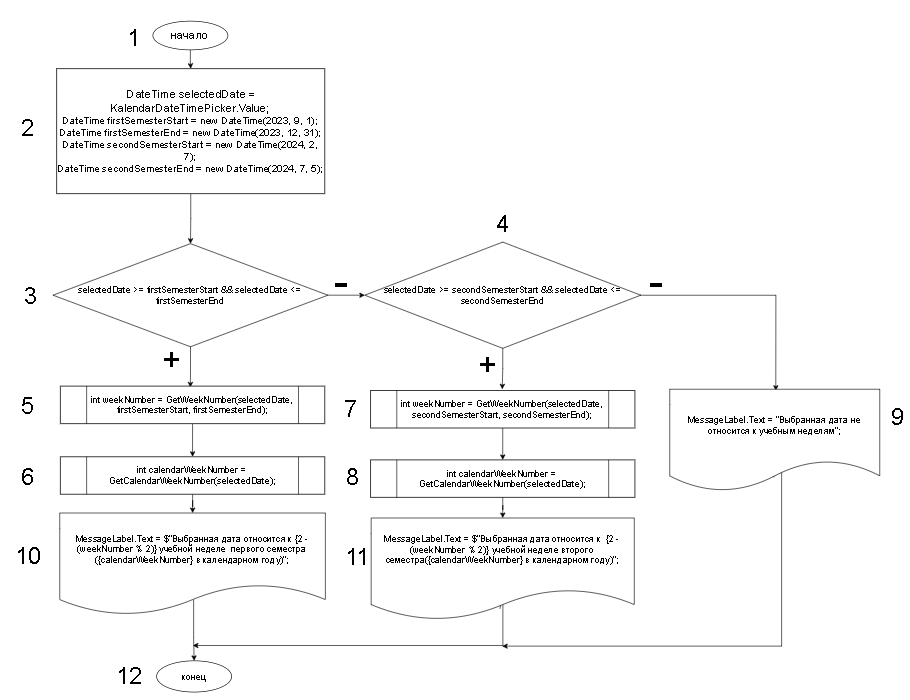


Рисунок 3 – Схема алгоритма метода KalendarDateTimePicker\_ValueChanged

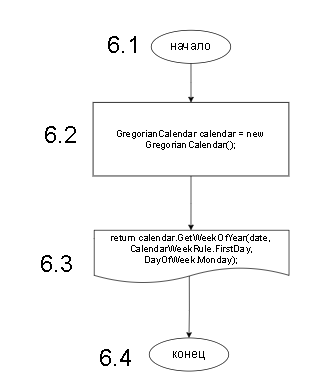


Рисунок 4 – Схема алгоритма метода GetCalendarWeekNumber

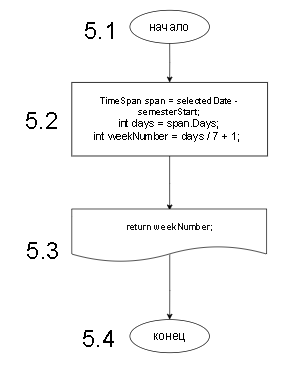


Рисунок 5 – Схема алгоритма метода GetWeekNumber

1. **Подбор тестовых примеров.**

Пример 1:

Выбирая дату 1 сентября 2023 года должно выводиться, что это первая неделя первого семестра в расписании РУТ МИИТ и 36 в календарном году

Пример 2:

Выбирая дату 29 февраля 2024 года должно выводиться, что это вторая неделя второго семестра в расписании РУТ МИИТ и 39 в календарном году

Пример 3:

Выбирая дату 1 февраля 2024 года должно выводиться, что дата не относится к учебным неделям

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

using System;

using System.Globalization;

using System.Windows.Forms;

namespace Rabota\_15

{

public partial class Kalendar : Form

{

public Kalendar()

{

InitializeComponent();

}

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

{

DateTime selectedDate = KalendarDateTimePicker.Value;

// Определение учебных недель

// Первый семестр: с 1 сентября 2023 по 31 декабря 2023

DateTime firstSemesterStart = new DateTime(2023, 9, 1);

DateTime firstSemesterEnd = new DateTime(2023, 12, 31);

// Второй семестр: с 7 февраля 2024 по 5 июля 2024

DateTime secondSemesterStart = new DateTime(2024, 2, 7);

DateTime secondSemesterEnd = new DateTime(2024, 7, 5);

// Проверка, к какой учебной неделе относится выбранная дата

if (selectedDate >= firstSemesterStart && selectedDate <= firstSemesterEnd)

{

int weekNumber = GetWeekNumber(selectedDate, firstSemesterStart, firstSemesterEnd);

int calendarWeekNumber = GetCalendarWeekNumber(selectedDate);

MessageLabel.Text = $"Выбранная дата относится к {2 - (weekNumber % 2)} учебной неделе первого семестра ({calendarWeekNumber} в календарном году)";

}

else if (selectedDate >= secondSemesterStart && selectedDate <= secondSemesterEnd)

{

int weekNumber = GetWeekNumber(selectedDate, secondSemesterStart, secondSemesterEnd);

int calendarWeekNumber = GetCalendarWeekNumber(selectedDate);

MessageLabel.Text = $"Выбранная дата относится к {2 - (weekNumber % 2)} учебной неделе второго семестра({calendarWeekNumber} в календарном году)";

}

else

{

MessageLabel.Text = "Выбранная дата не относится к учебным неделям";

}

}

private int GetWeekNumber(DateTime selectedDate, DateTime semesterStart, DateTime semesterEnd)

{

TimeSpan span = selectedDate - semesterStart;

int days = span.Days;

int weekNumber = days / 7 + 1;

return weekNumber;

}

private int GetCalendarWeekNumber(DateTime date)

{

GregorianCalendar calendar = new GregorianCalendar();

return calendar.GetWeekOfYear(date, CalendarWeekRule.FirstDay, DayOfWeek.Monday);

}

}

}

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

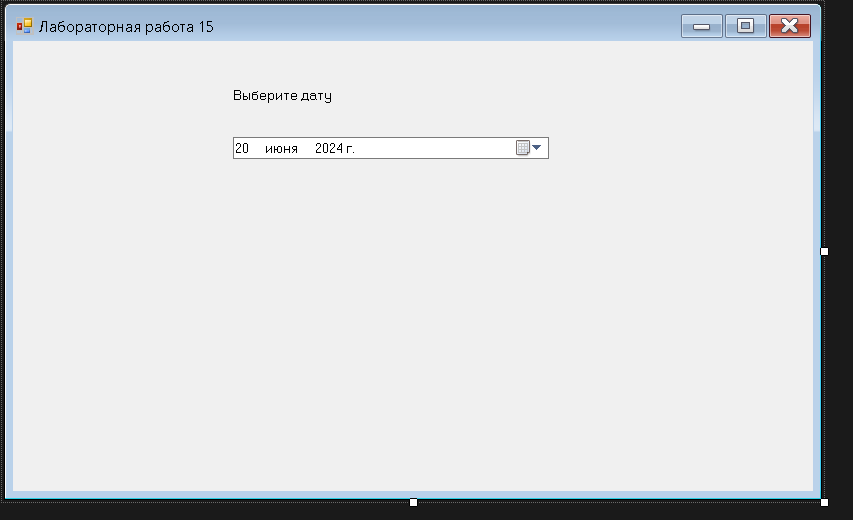


Рисунок 6 - Графический пользовательский интерфейс программного обеспечения

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

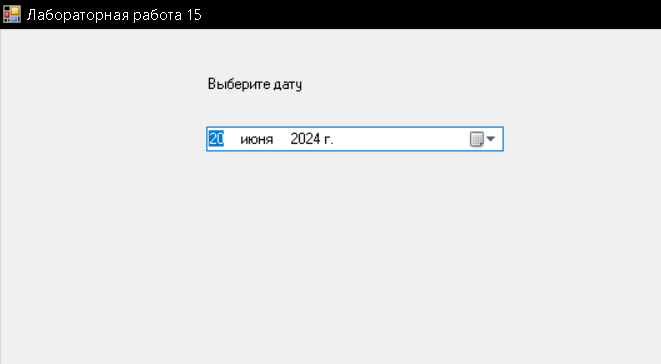


Рисунок 7 - Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

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

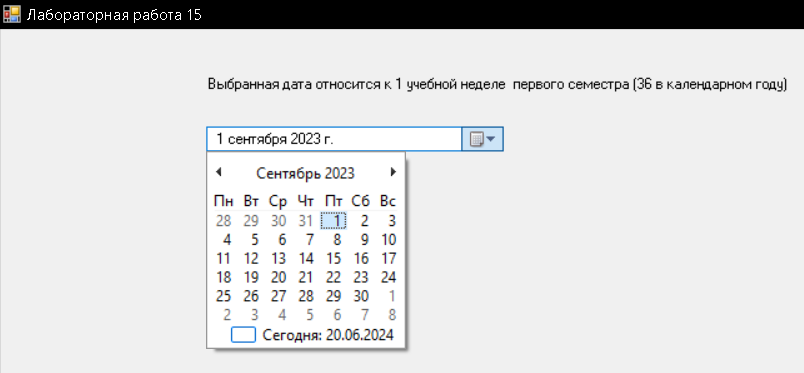


Рисунок 8 – Расчёт тестового примера 1

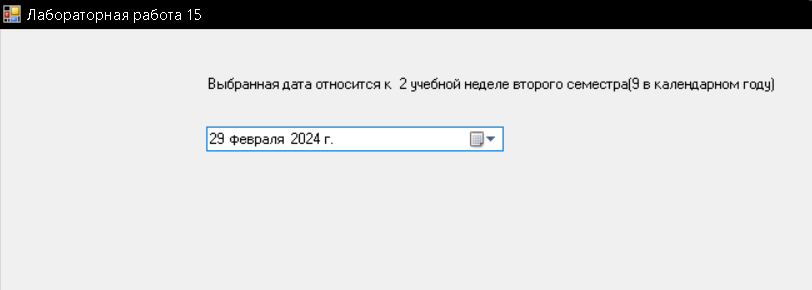


Рисунок 9 – Расчёт тестового примера 2

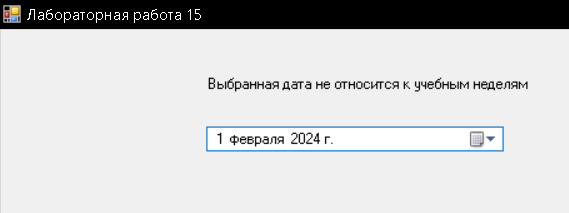


Рисунок 10 – Расчёт тестового примера 3

1. **Формулировку вывода о проделанной работе.**

Разработал программу в режиме *Windows Forms Application*, представляющую собой средство взаимодействия с инструментом календарь. Научился использовать элементы управления связанные с работой с календарём. Улучшил навыки работы в режиме *Windows Forms Application.*