**Задание Ш01.** В среде объектно-ориентированного программирования *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 в рамках графического пользовательского интерфейса раздел справочной информации, содержащий инструкцию пользователя к разработанному программному обеспечению. Организовать постраничный вывод справочной информации в экранную форму. Выводимые сведения необходимо хранить в текстовых файлах.

По итогам выполнения работы сдаются строго два файла:

- отчёт, выполненный в текстовом редакторе *Microsoft Office Word* (*\*.doc* или *\*.docx*);

- файл с архивом всего решения (*Solution*), написанного на языке *Visual C#* в среде *Microsoft Visual Studio* в режиме *Windows Forms Application* (*\*.zip* или *\*.rar*).

Отправленные поодиночке файлы проверке не подлежат. При отсутствии одного из упомянутых файлов зачёт по заданию не выставляется.

Раздел «Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению» наполнить демонстрацией скриншотами выполнения каждого из пунктов раздела «Требования к оформлению» из данной постановки задачи.

**Требования к именам файлов:**

**Общий вид формата имени файла:** «*Дата. Задание. Фамилия.mcdx*»

**Формат записи даты:** «*ГГГГММДД*», где *ГГГГ* – четыре цифры текущего года, *ММ* – две цифры текущего месяца, *ДД* – две цифры текущего дня.

**Формат записи задания:** «Задание *NNk*», где *NN* – две цифры номера задания, *k* – обозначение «о», если файл содержит общую часть; обозначение «и», если файл содержит индивидуальную часть; обозначение «ои», если файл содержит как общую, так и индивидуальную части.

**Если устранить замечания по работе удаётся в тот же день:** после фамилии ставится пробел и в круглых скобках записывается номер попытки исправления.

**Примеры правильных имён файлов, которые сдаются на проверку впервые:**

«*20190411. Задание 01ш. Иванов.docx*»

«*20190411. Задание 01ш. Иванов.rar*»

«*20190411. Задание 01ш. Иванов.vsdx*»

**Внимание!** Не забудьте выполнить автоматическую нумерацию страниц в отчёте.

Отчёт по выполненной работе должен содержать:

0. Титульный лист.

1. Формулировку цели работы.

2. Описание задачи согласно выданному варианту.

3. Составление диаграммы классов, входящих в состав решения.

4. Составление Сети Петри запрограммированного технологического процесса.

5. Составление блок-схем алгоритмов методов в составе решения.

6. Подбор и расчёт тестовых примеров.

7. Листинг кода составленного программного обеспечения.

8. Графический пользовательский интерфейс программного обеспечения и его описание.

9. Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

10. Расчёт тестовых примеров с использованием составленного программного обеспечения.

11. Формулировку вывода о проделанной работе.

Рекомендации к отчёту, доказывающие самостоятельность выполнения работы и упрощающие процедуру проверки отчёта преподавателем:

1. Выполнение дополнительных скриншотов для случаев, когда текстовое описание проделанных действий становится громоздким или трудным к восприятию.

2. Нумерация рисунков (если есть) с подписями, содержащими названия рисунков, например, «Рисунок 1 – Блок-схема алгоритма программы».

**Цель работы (одна из возможных формулировок)**: «Освоение навыков настройки элемента «Календарь на месяц» (*MonthCalendar*)».

**Требования к оформлению**:

1. Заголовок главной формы должен содержать надпись вида: «Задание №42 выполнил: [Фамилия И.О. автора]; Номер варианта: [Номер]; Дата выполнения: [дд/мм/гггг]».

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

3. Предусмотреть комбинированный список (чётные варианты), 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. Предусмотреть запись в файл интересующих дат, сформированных в результате работы программы (нечётные варианты).

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.

5. Нечётным вариантам необходимо отключить строку состояния с кнопками «*Minimize*», «*Maximize*», «*Close*». Взамен создать на форме авторские кнопки «Свернуть», «Развернуть», «Закрыть».

6. Предусмотреть по нажатии сочетания клавиш «*Ctrl+H*» вызов справки о работе с программой из файла в статический по размеру ярлык (чётные варианты). Повторное нажатие сочетания клавиш должно очищать ярлык. Предусмотреть вызов справки о работе каждого компонента программы путём размещения информации, считанной из файла, в компоненты *ToolTip* (всплывающая подсказка) подсвечиваемых элементов (нечётные варианты).

**Полезные сведения для выполнения задания:**

1. Для обеспечения возможности чтения файлов к проекту подключается модуль входа-выхода (*using System.IO*). В его состав входят необходимые классы *StreamReader* (для чтения) и *StreamWriter* (для записи).

2. Следует помнить, что по умолчанию считываемая/записываемая кодировка, воспринимаемая средой *Microsoft Visual Studio*, – это *UTF-8*, а информация, записанная в файлах *\*.txt*, как правило, хранится в кодировке *Windows-1251*. В связи с этим чтение потока данных следует выполнять с принудительным указанием кодировки: *StreamReader s = new StreamReader(inputPath, Encoding.GetEncoding(1251))*. Аналогичное замечание относится и к записи в файл *s = new StreamWriter(outputPath, false, Encoding.GetEncoding(1251))*.

3. Данные, считанные из файла, будут представлять собой строку текста, которую требуется перевести к типу «Дата/Время» посредством вызова метода *Convert.ToDateTime*([Строка по формату]);

4. Для выделения даты её необходимо добавить в коллекцию выделенных дат календаря на месяц [переменная календаря].*AddBoldedDate*([Дата]);

5. Перед циклом заполнения выделенных дат, коллекцию выделенных дат необходимо очистить методом [переменная календаря].*RemoveAllBoldedDates()*;

6. После цикла заполнения выделенных дат, выделенные даты необходимо обновить методом [переменная календаря].*UpdateBoldedDates()*;

7. Если на календаре выбирается конкретная дата, а не диапазон дат, то полные сведения об этой дате можно получить из [переменная календаря].*SelectionStart*.

**Варианты индивидуального задания:**

1. На правом (или нижнем) календаре выбирается дата. По итогам выбора даты некоторый интерфейсный элемент должен подсказать номер соответствующей недели в учебном семестре. Рассуждение по семестрам вести с учётом двух опорных дат: 1 сентября и 7 февраля. Выполнить проверку факта, что если опорная дата выпала на выходные дни, то первой неделей считать ближайший будний день предстоящей недели.

2. На правом (или нижнем) календаре отметить дату одного из основных праздников, принятых в Российской Федерации, выпавшего случайным образом. Смена даты является сигналом к выдаче подсказки (через любой удобный интерфейсный элемент) с названием выпавшего праздника. Обращение к датчику случайных чисел выполнять только по результату двойного клика по текущей дате в левом (или верхнем) календаре.

3. Вывести в любой удобный интерфейсный элемент возраст человека, выбранного из списка, наступивший к указанной в левом (или верхнем) календаре дате по известной дате его рождения, записанной в файле. При выборе человека его дата рождения отмечается в правом (или нижнем) календаре. Перечень людей, доступных для выбора через список, хранится в файле.

4. На правом (или нижнем) календаре подсказать пользователю ближайший день бесплатного посещения указанного им музея. Рассматриваемый музей определяется директорией с соответствующим названием. Рассматривается только текущий год.

5. На правом (или нижнем) календаре выполнить выделение всех выходных дней указанного года с учётом праздничных дней, принятых постановлением Правительства РФ на текущий год.

6. Вывести в любой удобный интерфейсный элемент возраст человека, наступивший к текущей дате по известной дате его рождения, указанной в правом (или нижнем) календаре.

7. Определить, каким днём недели было указанное в левом (или верхнем) календаре число в *Y* году. *Y* вводится через любой удобный интерфейсный элемент на усмотрение автора.

8. Выполнить расчёт количества дней, месяцев и лет между датой, указанной в левом (или верхнем) календаре и датой, указанной в правом (или нижнем). Вывести результат расчёта в любой интерфейсный элемент, а также в файл. Файл должен хранить предысторию выполненных расчётов. Очистка файла, если она необходима, должна производиться по нажатии на дополнительно предусмотренную кнопку.

9. Определить, каким днём недели было указанное в левом (или верхнем) календаре число *Y* лет назад. *Y* вводится через любой удобный интерфейсный элемент на усмотрение автора.

10. Выбор даты в левом (или верхнем) календаре сопровождать выводом недостающих сведений по ней (сезон, квартал) в любом (любых) удобном интерфейсном элементе на усмотрение автора.

11. На правом (или нижнем) календаре указать дату, которая настанет через *N* дней относительно выбранной на левом (или верхнем) календаре даты. *N* вводится через любой удобный интерфейсный элемент на усмотрение автора.

12. На левом (или верхнем) календаре выбирается текущая дата. Правый (или нижний) календарь подсказывает граничную дату, на которую открыта продажа билетов на поезд. Рассмотреть варианты продаж на 45, 60 и 90 суток, выбираемых через комбинированный список.

13. На левом (или верхнем) календаре выбирается дата. По итогам выбора даты некоторый интерфейсный элемент должен подсказать номер соответствующей недели в учебном семестре. Рассуждение по семестрам вести с учётом двух опорных дат: 1 сентября и 7 февраля. Пренебречь проверкой на выходные дни, выпадающие на эти даты.

14. На правом (или нижнем) календаре выполнить выделение всех выходных дней указанного года без учёта праздничных дней.

15. На левом (или верхнем) календаре отмечается дата. После в комбинированном списке, содержащем строки «I» и «II», указывается номер учебной недели (I / II), в строгом соответствии с действующим расписанием по МИИТу, которой принадлежит указанная дата – это входные данные.

На правом (или нижнем) календаре в ответ выделяются все даты учебного семестра, начиная с выбранной, которые относятся к предстоящим учебным неделям, соответствующим по номерам введённой неделе.

16. Сделать правый (или нижний) календарь Пасхальным календарём на весь XXI век.

17. На правом (или нижнем) календаре выделить основные праздники, принятые в Российской Федерации. Выбор одной из выделенных дат является сигналом к выдаче подсказки (через любой удобный интерфейсный элемент) с названием выбранного праздника.

18. На правом (или нижнем) календаре отметить даты бесплатного посещения музеев за указанный год. Рассматриваемый музей определяется директорией с соответствующим названием. Год определяется субдиректорией в директории каждого музея. Внутри располагается файл с интересующими датами.

19. На левом (или верхнем) календаре выбирается желаемая дата покупки билета на поезд. Правый (или нижний) календарь подсказывает дату старта продаж билетов на выбранную дату. Предусмотреть варианты продаж на 45, 60 и 90 суток. Выбор варианта продаж реализовать с использованием любого (на выбор автора) интерфейсного элемента.

20. На левом (или верхнем) календаре выбирается дата. По итогам выбора даты некоторый интерфейсный элемент (на выбор автора) должен подсказать номер соответствующей недели в году.