**Задание 15.**

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».

Предусмотреть в рамках графического пользовательского интерфейса раздел со справочной информацией («Помощь»), 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\*:**

«*20240411. Задание 15и. Иванов.docx*»

«*20240411. Задание 15и. Иванов.rar*»

«*20240411. Задание 15и. Иванов.vsdx*»

**Примеры правильных имён файлов, которые сдаются на проверку повторно в тот же день\*:**

«*20240411. Задание 15и. Иванов (1).docx*»

«*20240411. Задание 15и. Иванов (1).rar*»

«*20240411. Задание 15и. Иванов (1).vsdx*»

\* Правила распространяются только при обмене файлами через электронную почту. При работе на платформе *GitHub* устранение замечаний проводится в одних и тех же файлах без необходимости их переименования.

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

6. Подбор тестовых примеров.

7. Листинг (код) 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 – 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 №15 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.

**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:**

**Вариант 01.** На правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре выбирается дата. По итогам выбора даты некоторый интерфейсный элемент управления (выбирается разработчиком программного обеспечения) должен подсказать номер соответствующей недели в учебном семестре. Номер недели не в астрономическом году, а именно в учебном семестре. Рассуждение по семестрам вести относительно 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.

Переписать вручную в текстовый файл в удобном для разработчика программного обеспечения формате расписание двух любых учебных групп с сайта РУТ (МИИТ). На левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре отмечать жирным шрифтом даты, в которые у выбранной учебной группы проводится выбранная учебная дисциплина. Интерфейсные элементы управления для выбора учебных группы и дисциплины подбираются на усмотрение разработчика программного обеспечения, но из категории позволяющих хранить списочные структуры. Содержимое перечней для выбора учебных группы и дисциплины должно наполняться в строгом соответствии с содержимым текстового файла. Не вручную. При отсутствии текстового файла или при несоответствии его внутренней структуры ожиданиям оставлять перечни незаполненными и недоступными для пользователя. Так же поступить и с календарём, связанным с этой частью задания.

**Вариант 02.** Работа программного обеспечения начинается с момента нажатия на экранную кнопку «Старт». После этого на экранной форме отображаются все прочие предусмотренные интерфейсные элементы управления. Экранная кнопка «Старт» скрывается. На правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре отмечается жирным шрифтом дата одного из основных праздников, принятых в Российской Федерации, выпавшего случайным образом. Даты праздников собраны в текстовом файле без расширения в удобном для разработчика программного обеспечения формате. Автоматически выбирается первое число месяца, в котором отмечена праздничная дата. Если праздничная дата – первое число месяца, то выбирается последнее число месяца. Смена даты пользователем является сигналом для обработки события: если пользователь выбирает праздничную дату, то получает подсказку (через любой удобный интерфейсный элемент управления на усмотрение разработчика программного обеспечения) с названием праздника и краткой исторической справкой. Обращение к датчику псевдослучайных чисел, подбирающему другую праздничную дату из текстового файла, реализуется только по результату двойного щелчка левой кнопкой мыши по любой дате, кроме праздничной, в том же месяце, что и праздничная дата, но в левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре. Двойной щелчок по дате, совпадающей с праздничной, должен инициировать сброс интерфейсных элементов управление в исходное состояние (отображение экранной кнопки «Старт»).

**Вариант 03.** Вывести в любой удобный на усмотрение разработчика программного обеспечения интерфейсный элемент управления возраст человека, выбранного из списка, наступивший к моменту указанной в левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре дате по известному дню рождения человека, записанному в текстовом файле формата *CSV*. При выборе человека из списка его дата рождения отмечается в правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре жирным шрифтом. Перечень полных имён, фамилий и отчеств людей, доступных для выбора из списка (демонстрировать только Ф.И.О. без какой-либо дополнительной информации), хранится в том же текстовом файле формата *CSV*. По мере активации экранного логического признака с пометкой «Дни радости», расположенной возле него, в правом (или нижнем – на усмотрение разработчика программного обеспечения) отметить жирным шрифтом десятидневный период в окрестностях дня рождения человека таким образом, чтобы сама дата рождения оказалась в середине диапазона. Если в текстовом файле формата *CSV* для выбранного человека указан перечень желаемых им подарков, то при выборе такого человека из списка высвечивать перечень желаемых подарков в побуждающем сообщении.

**Вариант 04.** На правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре подсказать пользователю ближайший день бесплатного посещения указанного им музея в одном из удобных на усмотрение разработчика программного обеспечения интерфейсных элементов управления. Предусмотреть удобную для работы на усмотрение разработчика программного обеспечения структуру файлов и каталогов в решении (*Solution*). Рассматриваемый музей определяется каталогом с соответствующим названием, записанным кириллицей. Перечень каталогов, размещённых в структуре решения (*Solution*) определяет содержимое списка музеев. Заполнять перечень элементов наименованиями музеев вручную не требуется. Рассматривается только текущий календарный год. Каждый корректно наполненный каталог содержит только один текстовый файл с одинаковым именем «*free.days*». Если каталог музея содержит более одного файла, в числе которых обнаруживается «*free.days*», то такой музей должен отображаться в общем перечне с пометкой «\*» после всех музеев, каталоги которых оформлены правильно. Если каталог музея не содержит ни одного файла, то его не требуется включать в перечень подгружаемых в программное обеспечение. Если каталог музея содержит несколько файлов, но среди них нет «*free.days*», то такой музей так же не следует отображать в перечне.

**Вариант 05.** На правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре выполнить маркировку жирным шрифтом всех выходных дней текущего календарного года, сформировавшихся на основе сведений о праздничных днях, принятых постановлением Правительства Российской Федерации на текущий календарный год. Обычные выходные дни (субботы и воскресенья) маркировать жирным шрифтом не требуется. Если какие-либо даты из перечня праздничных дней, принятых постановлением Правительства Российской Федерации на текущий календарный год, накладываются на выходные дни (субботы и воскресенья), то такие выходные дни должны быть маркированы жирным шрифтом полностью: и субботы, и воскресенья. По нажатии на некоторую предусмотренную экранную кнопку, выдавать получившееся количество выходных дней, удовлетворяющих установленным ограничениям. На левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре отметить жирным шрифтом выходной день, который оказался на центральной позиции в сформированном под ограничения перечне.

**Вариант 06.** Вывести в любой удобный интерфейсный элемент управления стаж работы человека – сотрудника в некоторой компании, «накопившийся» к текущей дате по известной дате приёма человека на работу в организацию. Дата приёма человека в штат организации указывается в правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре. Стаж должен считаться с точностью до месяцев с округлением в меньшую сторону (если достижения дня приёма человека в штат организации в очередном месяце не случилось, то счётчик количества месяцев не увеличивается). В случае, если полностью корректно не указаны Ф.И.О. человека через специально предусмотренный для этих целей интерфейсный элемент управления (на усмотрение разработчика программного обеспечения), то не сообщать стаж работы сотрудника. Список сотрудников (без указания каких-либо дат) содержится в текстовом файле. Корректность ввода Ф.И.О. должна быть связана с обнаружением точного соответствия того же значения в одной из строк текстового файла.

**Вариант 07.** На левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре выбирается необходимая расчётная дата. Правый (или нижний – на усмотрение разработчика программного обеспечения) календарь подсказывает множеством маркированных жирным шрифтом дат диапазон, на который открыта продажа билетов на поезда (ОАО «РЖД»). Предлагать пользователю программного обеспечения варианты проведения расчётов для случаев продажи билетов на 45, 60 и 90 суток. В качестве тестовых примеров брать реальные расчёты с официального сайта ОАО «РЖД». Смену вариантов маркировки рабочего диапазона дат реализовать через любой удобный интерфейсный элемент управления на усмотрение разработчика программного обеспечения.

**Вариант 08.** На левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре отмечается опорная дата. После отметки даты в комбинированном списке, содержащем только две строки «I» и «II», указывается номер учебной недели («I» / «II») в строгом соответствии с действующим расписанием, опубликованным на официальном сайте РУТ (МИИТ), соответствующий отмеченной дате – это исходные данные. Обязательно указать в тестовых примерах последовательность определения исходных данных по официальному сайту РУТ (МИИТ). На правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре программным обеспечением в ответ выделяются жирным шрифтом все даты только одного учебного семестра, начиная с отмеченной опорной, которые относятся к предстоящим учебным неделям и соответствуют указанному номеру учебной недели.

**Вариант 09.** Сделать оба календаря Пасхальными календарями на весь XXI век для разных христианских религий. Сведения о Пасхальных датах не считать по специальной формуле, а найти в сети Интернет наперёд рассчитанные таблицы дат. Наиболее удачные находки разместить в разделе с тестовыми примерами. Найденные даты вписать вручную в текстовый файл в формате, удобном на усмотрение разработчика программного обеспечения. Даты, размещённые в файле, после его прочтения в программном обеспечении маркируются на календарях жирным шрифтом. Под каждым календарём разместить экранные логические признаки по количеству учитываемых христианских религий. Под каждым календарём активируется только одна «галочка», соответствующая интересующей христианской религии. Остальные «галочки» должны быть сброшены. Для двух календарей не может быть выставлена одинаковая христианская религия. При попытках оператора принудительно выставить одинаковые христианские религии вторичная по отношению к текущей «галочка» должна переместиться на следующую позицию (для последнего варианта следующая позиция – первая).

**Вариант 10.** Смоделировать с использованием интерфейсного элемента управления «календарь на месяц» работу системы бронирования билетов на киносеансы в микро-кинотеатре (10 посадочных мест). Каждый фильм показывается только три раза в день. Использовать два файла, в одном из которых хранить фильмы и диапазоны дат их показов, во втором – произведённые посетителями бронирования мест в различное время. Пользователь выбирает фильм из списка. В список попадают только те фильмы, которые содержатся в текстовом файле. Вручную вводить фильмы в список не требуется. В правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре жирным шрифтом маркируются даты показа из записанного в файле диапазона. Если все места на все доступные на день сеансы забронированы, то отменять маркировку этой даты жирным шрифтом. Все даты, на которые невозможно забронировать билеты, демонстрируются на левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре вне зависимости от выбора фильма.

**Вариант 11.** На метрополитене для проведения ремонтов и диагностических осмотров действуют следующие правила: технический осмотр второго набора работ (диагностика) проводится по мере наката электроподвижным составом 100 тыс. вагоно-километров пробега, технический осмотр третьего набора работ (текущий ремонт) – по мере наката 250 тыс. вагоно-километров пробега. Из расчёта, что один вагон в электроподвижном составе по линии за час проезжает, в среднем, 50 километров, метрополитен работает не более 20 часов в сутки, график движения поездов составляется на пару соседних сезонов (лето-осень или зиму-весну, то есть на полгода) спланировать на период действия графика движения поездов с использованием интерфейсного элемента управления «календарь на месяц» (*Month Calendar*) даты проведения ремонтно-диагностических мероприятий. На левом (или верхнем – на усмотрение разработчика программного обеспечения) календаре жирным шрифтом отображать даты проведения диагностик; в правом (или нижнем – на усмотрение разработчика программного обеспечения) календаре жирным шрифтом отображать даты проведения текущих ремонтов. Вагонов в составах на различных линиях может быть 6, 7 или 8. Предусмотреть интерфейсный элемент управления, позволяющий удобно выбирать количество вагонов, которые можно предварительно заполнить вручную. Предусмотреть интерфейсный элемент управления «выбор даты и времени» (*DateTime Picker*) для указания отправной даты вступления в действие графика движения поездов. От этой даты отсчитывается полугодовой период.

**Вариант 12.**На правом (или нижнем) календаре отметить даты бесплатного посещения музеев за указанный календарный год. Рассматриваемый музей определяется файлом с соответствующим названием. Год так же определяется по названию файла. Формат наименования файла продумать максимально удобным для парсинга.

**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.

**Вариант 14.** На правом (или нижнем) календаре выполнить маркировку всех выходных дней указанного календарного года без учёта праздничных дней. Сведения о выходных днях допустимо хранить в текстовых файлах, даже для случаев, когда выходные дни подчиняются классической логике (субботы и воскресенья).

**Вариант 15.** 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 управления, а также в текстовый файл. Файл должен хранить предысторию выполненных расчётов. Очистка текстового файла, если она необходима, должна производиться по нажатии на дополнительно предусмотренную экранную кнопку.

**Вариант 16.** Определить, каким днём недели было указанное в левом (или верхнем) календаре число *Y* лет назад. *Y* вводится через любой удобный интерфейсный элемент управления на усмотрение автора программного обеспечения.

**Вариант 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 управления) с названием выбранного праздника. Ежегодные праздники хранить в датах ежегодной маркировки, праздники с плавающей датой хранить в массиве обычной маркировке только на рассматриваемый календарный год.

**Вариант 18.** Определить, каким днём недели было указанное в левом (или верхнем) календаре число в *Y* году. *Y* вводится через любой удобный интерфейсный элемент управления на усмотрение разработчика программного обеспечения.

**Вариант 19.** На левом (или верхнем) календаре выбирается желаемая дата поездки на поезде (ОАО «РЖД») в другой город. Правый (или нижний) календарь подсказывает дату старта продаж билетов на выбранную дату. Предусмотреть варианты продаж билетов на 45, 60 и 90 суток. Выбор варианта продаж реализовать с использованием любого (на выбор автора программного обеспечения) интерфейсного элемента управления. Вместе с тем должен маркироваться диапазон дат между желаемой датой поездки и датой старта продаж билетов. В разделе с тестовыми примерами сопоставиться с данными официального сайта ОАО «РЖД».

**Вариант 20.** На левом (или верхнем) календаре выбирается дата. По итогам выбора даты некоторый интерфейсный элемент управления (на выбор автора программного обеспечения) должен подсказать номер соответствующей недели в календарном году. Дополнительно правый (или нижний) календарь должен указывать эту же дату в год рождения автора программного обеспечения, а дополнительный удобный интерфейсный элемент управления должен выводить полное наименование дня недели для этой даты.

**Вариант 21.** Выбор даты в левом (или верхнем) календаре сопровождать выводом недостающих сведений по ней (сезон, квартал) в любом (любых) удобном интерфейсном элементе (элементах) управления на усмотрение автора программного обеспечения.

**Вариант 22.** На правом (или нижнем) календаре указать дату, которая настанет через *N* дней относительно выбранной даты на левом (или верхнем) календаре. *N* вводится пользователем через любой удобный интерфейсный элемент управления на усмотрение автора программного обеспечения.