**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) календаре выбирается дата. По итогам выбора даты некоторый интерфейсный элемент управления 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.

**Вариант 02.** 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 псевдослучайных чисел выполнять только по результату двойного клика по текущей дате в левом (или верхнем) календаре.

**Вариант 03.** Вывести в любой удобный интерфейсный элемент управления возраст человека, выбранного из списка, наступивший к указанной в левом (или верхнем) календаре дате по известной дате его рождения, записанной в текстовом файле. При выборе человека из списка его дата рождения отмечается в правом (или нижнем) календаре. Перечень людей, доступных для выбора через список, хранится в текстовом файле.

**Вариант 04.** На правом (или нижнем) календаре подсказать пользователю ближайший день бесплатного посещения указанного им музея. Рассматриваемый музей определяется директорией с соответствующим названием. Рассматривается только текущий календарный год.

**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.

**Вариант 06.** Вывести в любой удобный интерфейсный элемент управления возраст человека, наступивший к текущей дате по известной дате его рождения, указанной в правом (или нижнем) календаре.

**Вариант 07.** 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 автора программного обеспечения.

**Вариант 08.** 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 09.** 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 10.** 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 11.** 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.

**Вариант 12.**На левом (или верхнем) календаре выбирается текущая дата. Правый (или нижний) календарь подсказывает граничную дату, на которую открыта продажа билетов на поезд (ОАО «РЖД»). Рассмотреть варианты продаж билетов на 45, 60 и 90 суток, выбираемых через комбинированный список. В разделе с тестовыми примерами сопоставиться с данными официального сайта ОАО «РЖД».

**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.** На левом (или верхнем) календаре отмечается дата. После выбора даты в комбинированном списке, содержащем только две строки «I» и «II», указывается номер учебной недели (I / II), в строгом соответствии с действующим расписанием по РУТ (МИИТ), которой принадлежит указанная дата – это входные (исходные) данные. На правом (или нижнем) календаре программным обеспечением в ответ выделяются все даты учебного семестра, начиная с выбранной, которые относятся к предстоящим учебным неделям, соответствующим по номерам введённой неделе.

**Вариант 16.** Сделать правый (или нижний) календарь Пасхальным календарём на весь XXI век. Сведения о Пасхальных датах дублировать в текстовом файле.

**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.** 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 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.