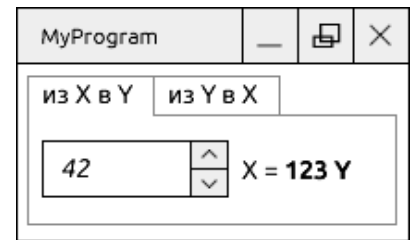


ДЕНЬ 5. ОКОННЫЕ ПРИЛОЖЕНИЯ 2

5.1. Создать программу-калькулятор пересчета единиц измерения

Создать форму для пересчета единиц измерения, состоящую из двух вкладок (*TabControl*), определяющих направление перевода. Каждая вкладка отвечает за свое направление перевода и содержит следующие элементы:

- поле ввода входного значения
- блок вывода результата

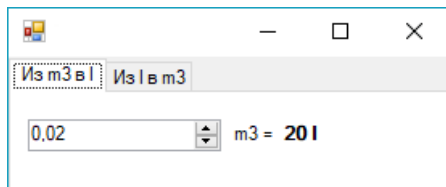


При вводе входного значения, результат пересчитывается автоматически, без дополнительных действий пользователя. Входное значение должно быть синхронизировано между вкладками, т.е. сохраняться при переключении.

В блоке вывода результат должен быть выделен жирным шрифтом.

Пример

Перевод куб. метров в литры ($1 \text{ m}^3 = 1000 \text{ l}$) и наоборот



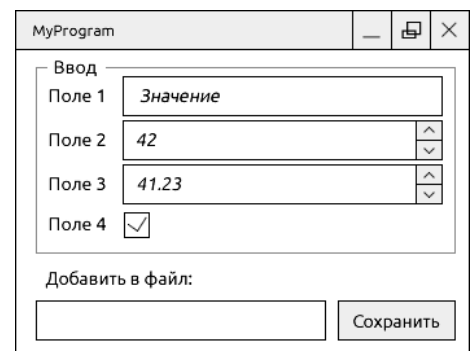
Варианты

№	Перевод	№	Перевод	№	Перевод	№	Перевод
1	1 \$ = 67.2 rub	6	1 h = 60 min	11	1 gal = 3.7854 L	16	1 ft = 0.3048 m
2	1 c = 100 kg	7	1 ha = 0.01 km ²	12	1 ozt = 31.103 g	17	1 y = 12 month
3	1 km = 1000 m	8	1 min = 60 s	13	1 s = 1000 ms	18	1 KB = 1024 B
4	1 d = 24 h	9	1 m = 100 cm	14	1 y = 365 d	19	1 d = 2.54 cm
5	1 eur = 77.1 rub	10	1 p = 0.4536 kg	15	1 kg = 1000 g	20	1 hf = 735.5 W

5.2. Создать форму ввода записей, добавляющихся в конец заданного CSV-файла

Создать форму содержащую:

- элементы управления для ввода полей записи:
 - *TextBox* – для ввода поля типа «строка»
 - *NumericUpDown* с шагом 1 - для типа «целое число»
 - *NumericUpDown* с шагом 0.01 – для типа «вещественное число»
 - *CheckBox* – для логического типа
- поле ввода имени файла
- кнопку «Сохранить»



При нажатии на кнопку «Сохранить» программа должна или создать новый CSV-файл, если файл, указанный в поле ввода имени файла, не существует, или добавить запись в конец уже существующего CSV-файла. Если поле ввода имени файла пусто или при

сохранении произошла ошибка записи, программа должна вывести сообщение о соответствующей ошибке во всплывающем окне.

Пример

Запись: фамилия (строка), возраст (целое число), результат (вещественное число)

Варианты

№	Запись
1	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)
2	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)
3	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)
4	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)
5	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)
6	Фамилия (строка), рост (вещественное число), вес (вещественное число)
7	Фамилия (строка), должность (строка), оклад в руб (целое число)
8	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)
9	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)
10	Дисциплина (строка), номер курса (целое число), количество часов (целое число)
11	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)
12	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)
13	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)
14	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)
15	Название материала (строка), объем (вещественное число), вес (вещественное число)
16	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)
17	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)
18	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)
19	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)
20	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)

5.3. Создать форму, отображающую список записей из CSV-файла

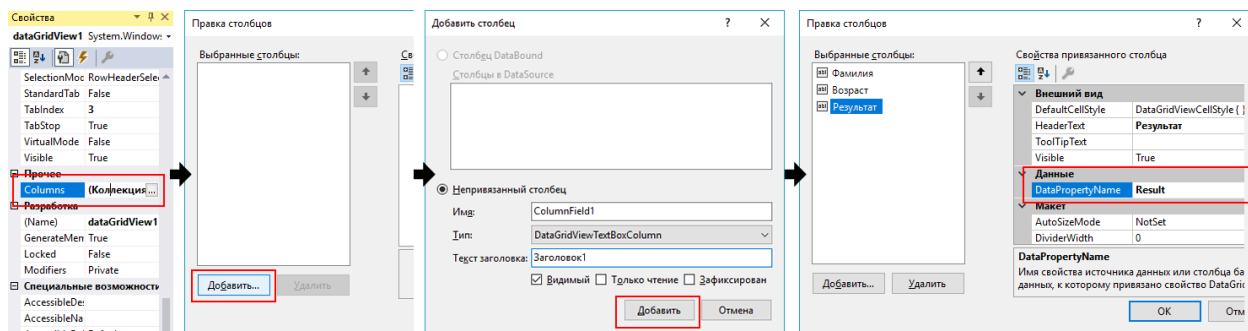
Создать форму, содержащую:

- поле ввода имени файла
- кнопку «Открыть»
- таблицу отображения данных (*DataGridView*)

При нажатии на кнопку «Открыть» программа должна открыть указанный CSV-файл и вывести все записи в таблицу отображения данных. Если файл не найден или его невозможно открыть, вывести сообщение об ошибке во всплывающем окне

Поле 1	Поле 2	Поле 3	Поле 4
Значение1	1	1,1	True
Значение2	2	2,2	False

Для использования *DataGridView* необходимо создать вспомогательный класс, который будет хранить данные об одной записи. Поля записи в классе должны быть определены как **свойства** (см. описание отдельной записи в примере ниже). После добавления *DataGridView* на форму, необходимо задать колонки таблицы (свойство *Columns*). В каждой колонке необходимо указать параметр *DataPropertyName* – имя свойства класса записи, значения которого должны отображаться в этой колонке.



После настройки колонок присвойте свойству *DataSource* у таблицы значение списка записей.

Пример

Запись: фамилия (строка), возраст (целое число), результат (вещественное число)

Фамилия	Возраст	Результат
Петров	20	5,3
Иванов	22	7,6
Сидоров	20	5

Описание отдельной записи:

```

public class Record
{
    public string Name { get; set; }
    public int Age { get; set; }
    public double Result { get; set; }
}

```

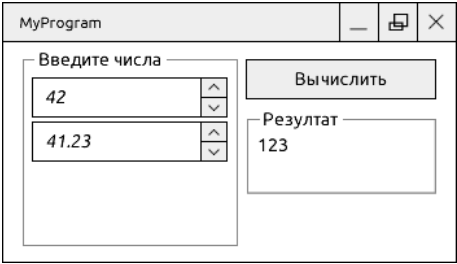
Варианты

№	Запись
1	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)
2	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)
3	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)
4	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)
5	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)
6	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)
7	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)
8	Название материала (строка), объем (вещественное число), вес (вещественное число)
9	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)
10	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)
11	Дисциплина (строка), номер курса (целое число), количество часов (целое число)
12	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)
13	Фамилия (строка), должность (строка), оклад в руб (целое число)
14	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)
15	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)
16	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)
17	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)
18	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)
19	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)
20	Фамилия (строка), рост (вещественное число), вес (вещественное число)

5.4. Создать программу, выполняющую заданное действие с набором чисел, вводимых в динамически создаваемые поля

Создать форму, содержащую следующие элементы:

- блок ввода чисел: *TableLayoutPanel*, в котором динамически добавляются элементы *NumericUpDown* со свойствами *Increment = 0.01* и *DecimalPlaces = 2*
- кнопка «Вычислить»
- блок с результатом

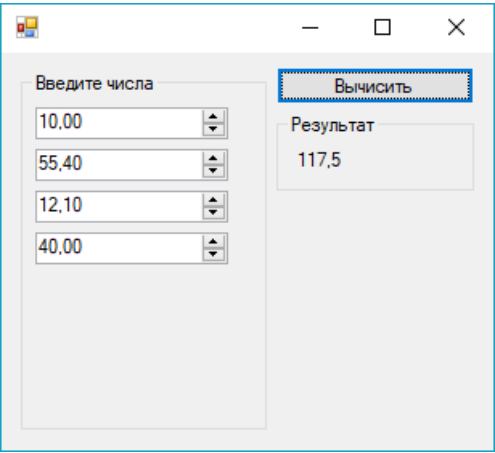


Изначально в блоке ввода чисел должен быть только один элемент *NumericUpDown*. Если фокус находится на этом элементе и пользователь нажимает клавишу «Enter», ниже должно появиться следующее поле ввода и фокус должен переместиться на него. Если фокус находится на одном из элементов блока ввода и пользователь нажимает клавишу «Delete», этот элемент должен удалиться (если он не последний) и фокус должен переместиться на элемент выше (или ниже, если это первый элемент).

При нажатии на кнопку «Вычислить» программа должна выполнить требуемую операцию со всеми числами из блока ввода и вывести полученное значение в блоке результата

Пример

Просуммировать числа



Варианты

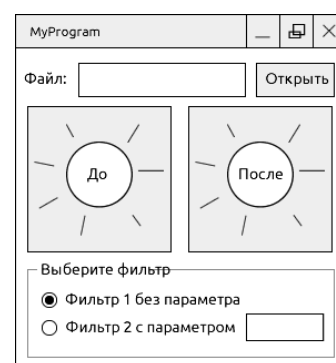
№	Операция	№	Операция
1	Перемножить все положительные числа	11	Просуммировать модули всех чисел
2	Просуммировать все нечетные числа	12	Вычислить среднее из квадратов чисел
3	Просуммировать все четные числа	13	Просуммировать квадраты всех чисел
4	Перемножить дробные части всех чисел	14	У каждого второго числа поменять знак и просуммировать все числа
5	Разделить все числа на первое из чисел и вычислить сумму результатов	15	У каждого второго числа поменять знак и посчитать среднее

6	Умножить числа на их порядковые номера и сложить результаты	16	Перемножить все отрицательные числа
7	Извлечь квадратный корень из суммы квадратов чисел	17	Вычислить среднее значение
8	Просуммировать дробные части всех чисел	18	Просуммировать все числа, которые больше первого из чисел
9	Умножить числа на их порядковые номера и вычислить среднее	19	Округлить числа и сложить результаты
10	Просуммировать все числа, которые меньше, чем предыдущее число	20	Просуммировать все числа, которые больше, чем предыдущее число

5.5. Создать форму применения фильтра к BMP-изображению

Создать форму, содержащую следующие элементы:

- поле ввода имени файла
- кнопка «Открыть»
- блок отображения входного изображения
- блок выбора фильтра (группа с радиокнопками)
- блок отображения результирующего изображения

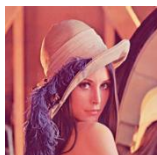


Блок выбора фильтра должен предлагать на выбор один из двух фильтров (согласно варианту). Некоторые фильтры требуют ввода дополнительного параметра. Изначально выбран первый фильтр. Одновременно активным может быть только один фильтр.


При нажатии на кнопку «Открыть» программа должна открыть указанный BMP-файл, отобразить его в блоке входного изображения, применить к нему текущий активный фильтр и вывести измененное изображение в блоке результирующего изображения. Если файл не найден или его невозможно открыть, вывести сообщение об ошибке во всплывающем окне.



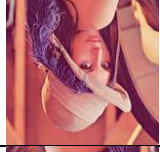
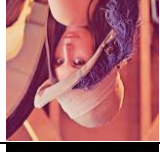


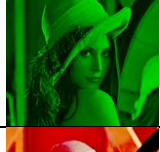




При смене активного фильтра, если входное изображение уже загружено, результирующее изображение должно обновиться автоматически.



Пример входного изображения:



Список фильтров:

Код	Описание	Пример работы
Ф1	Перевод в градации серого (усреднение всех каналов цвета)	

Ф2	Бинаризация по порогу (задается параметр T , пиксели, яркость которых меньше T , становятся черными, больше T - белыми)	
Ф3	Отражение изображения по горизонтали	
Ф4	Отражение изображения по вертикали	
Ф5	Отражение изображения по горизонтали и по вертикали	
Ф6	Добавление черной рамки вокруг изображения (задается параметр W , определяющий размер рамки)	
Ф7	Зануление всех каналов, кроме красного	
Ф8	Зануление всех каналов, кроме синего	
Ф9	Зануление всех каналов, кроме зеленого	
Ф10	Усиление черного (задается параметр T , все пиксели меньше этого значения, становятся черными по каждому из каналов)	
Ф11	Усиление белого (задается параметр T , все пиксели больше этого значения, становятся белыми по каждому из каналов)	
Ф12	Инвертирование цветов (негатив)	
Ф13	Выделить в круг (задается параметр R - радиус круга. Все пиксели, находящиеся дальше от центра, чем R , становятся белыми)	

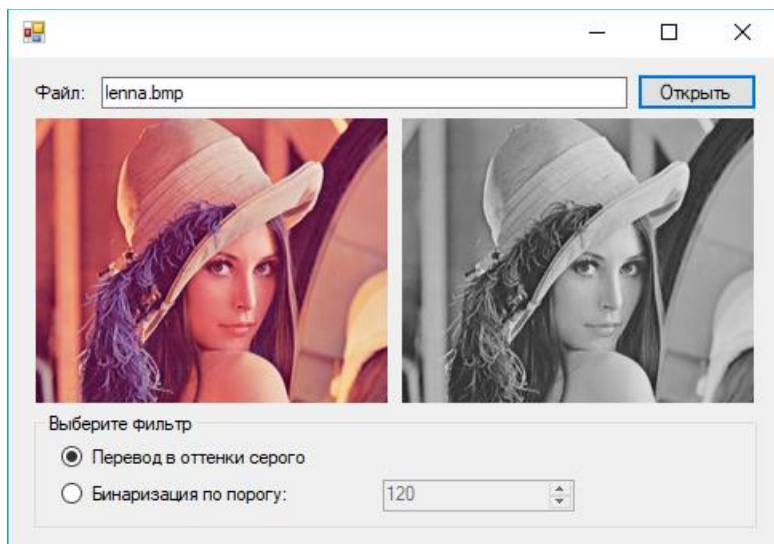
Ф14	Вырезать круг (задается параметр R - радиус круга. Все пиксели, находящиеся ближе от центра, чем R, становятся белыми)	
Ф15	Удвоение значения яркости	

Для открытия и работы с изображением используйте класс *System.Drawing.Bitmap*. Подключите в проекте ссылку на стандартную библиотеку *System.Drawing.dll*, если класс недоступен.

Для получения цвета пикселя используйте *Bitmap.GetPixel(Int32, Int32)*. Для установки значения пикселя – *Bitmap.SetPixel(Int32, Int32, System.Drawing.Color)*

Пример

Фильтры: перевод в оттенки серого (Ф1), бинаризация по порогу (Ф2)



Варианты

№	Фильтр		№	Фильтр		№	Фильтр		№	Фильтр	
	1	2		1	2		1	2		1	2
1	Ф10	Ф9	6	Ф1	Ф8	11	Ф14	Ф12	16	Ф6	Ф12
2	Ф2	Ф11	7	Ф13	Ф5	12	Ф13	Ф9	17	Ф15	Ф8
3	Ф14	Ф14	8	Ф2	Ф7	13	Ф10	Ф15	18	Ф13	Ф9
4	Ф11	Ф4	9	Ф4	Ф15	14	Ф1	Ф5	19	Ф10	Ф3
5	Ф3	Ф9	10	Ф6	Ф11	15	Ф7	Ф4	20	Ф12	Ф5