

ДЕНЬ 3D. Оконные приложения

3D.1. Создать программу-калькулятор цены доставки

Создать форму для расчета стоимости доставки.

Форма должна состоять из следующих элементов:

- полей ввода координат (X, Y)
- трех групп элементов для ввода веса (w_i) и объема (v_i) посылок
- кнопки «Вычислить»

Все параметры – вещественные значения и должны вводиться через элемент *NumericUpDown*

Пользователь в форме может задать вес и объем до трех посылок. Если хотя бы одно из полей (вес или объем) посылки равно 0, то эта посылка не должна учитываться

При нажатии на кнопку «Вычислить» программа должна рассчитать стоимость доставки согласно формуле, указанной в варианте, округлить ее до 2 знаков после запятой и вывести ответ в всплывающем сообщении (*MessageBox*)

Обозначения: \sum - сумма элементов, \max – максимальный элемент, \min – минимальный элемент, $\sqrt{}$ – квадратный корень, $|z|$ - взятие модуля числа

Пример

Формула: $\sum w_i + \max(v_i) + x + y$

Варианты

№	Формула	№	Формула	№	Формула
1	$ x - y / \sqrt{x^2 + y^2} * \sum v_i$	9	$(x + y) * \sum v_i * 0.123$	17	$\sum w_i / \max(v_i) * x - y $
2	$\sqrt{x^2 + y^2} * \sum w_i * 0.1$	10	$ x - y * \max(v_i) * 0.2$	18	$\sqrt{x * y} * \max(v_i) * 0.7$

3	$(x + y)*\max(v_i^2+w_i^2)$	11	$ x-y * \min(v_i^2) / \max(w_i)$	19	$\sum w_i * \sum v_i + x-y $
4	$(x + y)*\sum v_i * \sum w_i$	12	$(x + y) * \sum v_i / \min(w_i)$	20	$\sqrt{x^2+y^2} + \sum w_i * \sum v_i$
5	$\sum w_i * \max(v_i) + x-y $	13	$\sqrt{x^2+y^2} * \sum v_i / 95$		
6	$\max(w_i) * \max(v_i) + x-y $	14	$\max(v_i/w_i)* x-y $		
7	$\sqrt{x*y} * \sum v_i * 0.165$	15	$\sum (v_i * w_i) * \sqrt{x^2+y^2}$		
8	$\sqrt{x^2+y^2} * \sum v_i * \sum w_i / 3$	16	$\sqrt{x^2+y^2} * \max(w_i) / 7$		

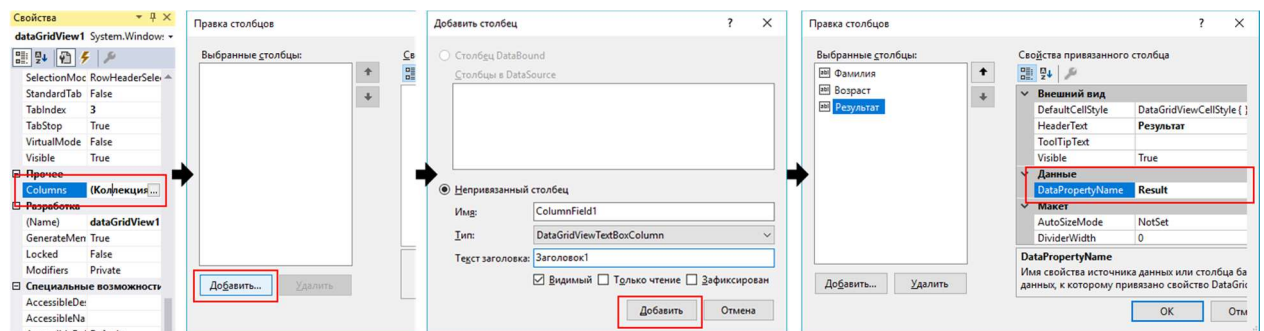
3D.2. Создать форму, отображающую список записей из CSV-файла

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

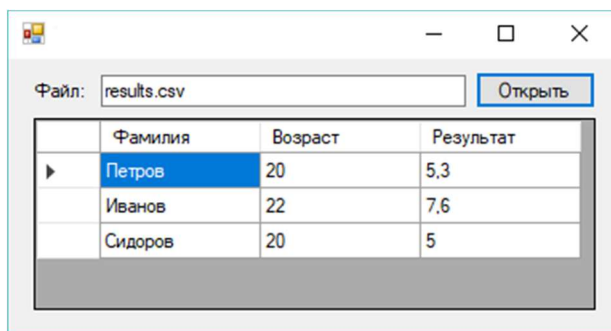
Для использования *DataGridView* необходимо создать вспомогательный класс, который будет хранить данные об одной записи. Поля записи в классе должны быть определены как **свойства** (см. описание отдельной записи в примере ниже). После добавления *DataGridView* на форму, необходимо задать колонки таблицы (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)



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)

3D.3. 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	Умножить числа на их порядковые номера и сложить результаты
2	Разделить все числа на первое из чисел и вычислить сумму результатов	12	Извлечь квадратный корень из суммы квадратов чисел
3	Просуммировать все четные числа	13	Просуммировать все нечетные числа
4	Просуммировать все числа, которые меньше, чем предыдущее число	14	Умножить числа на их порядковые номера и вычислить среднее

5	Просуммировать все числа, которые больше первого из чисел	15	У каждого второго числа поменять знак и посчитать среднее
6	Просуммировать квадраты всех чисел	16	Округлить числа и сложить результаты
7	Просуммировать дробные части всех чисел	17	У каждого второго числа поменять знак и просуммировать все числа
8	Вычислить среднее значение	18	Просуммировать модули всех чисел
9	Перемножить все отрицательные числа	19	Вычислить среднее из квадратов чисел
10	Просуммировать все числа, которые больше, чем предыдущее число	20	Перемножить дробные части всех чисел

3D.4. Создать программу для редактирования текста с диалогами открытия и сохранения файла

Создать форму, содержащую элементы (см. пример):

- меню (*MenuStrip*):
 - подменю «Файл»
 - кнопка «Открыть...»
 - кнопка «Сохранить»
 - кнопка «Сохранить как...»
 - кнопка «Выход»
 - кнопка «Преобразовать текст»
- многострочное поле ввода текста (*TextBox* со свойством *Multiline* = true)

При запуске программы кнопка «Сохранить» в меню должна быть неактивной

Кнопка в меню «Открыть» должна открывать стандартный диалог открытия файла, в котором доступны для выбора только файлы с расширением .txt. После выбора файла его содержимое помещается в поле ввода текста (старое значение очищается) и становится активной кнопка «Сохранить»

Кнопка «Сохранить» сохраняет текущий текст из поля ввода в последний из открытых файлов

Кнопка «Сохранить как...» должна открывать стандартный диалог сохранения файла с расширением по умолчанию .txt. После выбора файла текущий текст из поля ввода записывается в выбранный файл.

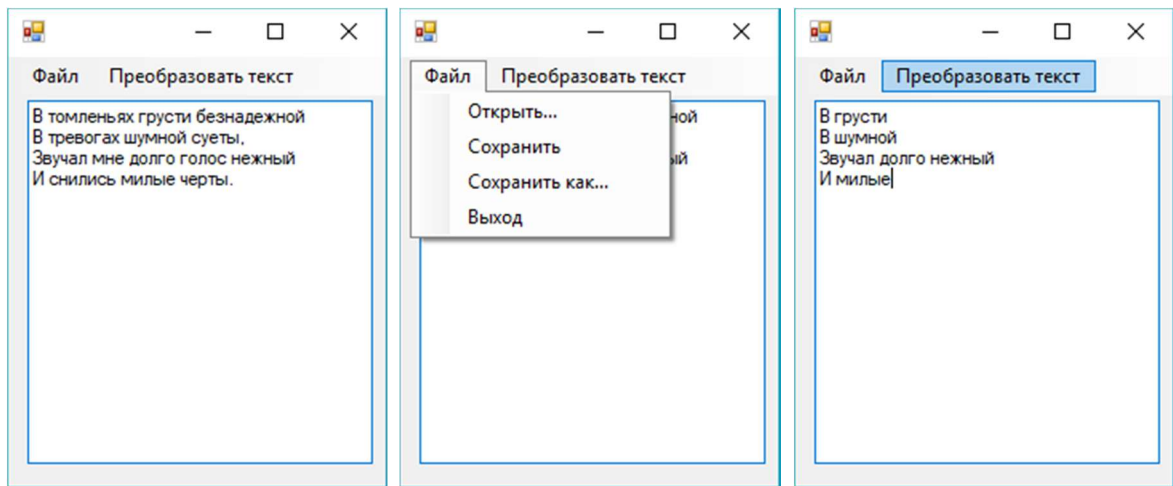
Кнопка «Выход» должна закрывать программу

Кнопка «Преобразовать текст» должна изменить текст в поле ввода согласно варианту

При закрытии формы, если были несохраненные изменения, программа должна попросить подтверждение выхода во всплывающем окне.

Пример

Удаление каждого второго слова



Варианты

№	Преобразование	№	Преобразование
1	Перевести первую букву каждого слова в верхний регистр	11	Перевести в верхний регистр все гласные
2	Перевернуть строки (абг деж -> жед гба)	12	Удалить все слова, в которых больше 5 согласных
3	Перевести все буквы первого слова каждой строке в верхний регистр	13	Перевернуть каждое слово в строке (абг деж -> гба жед)
4	Перевести в верхний регистр каждую вторую букву слов: абв гдеж -> аБв гДеЖ	14	Удалить все слова, в которых меньше 5 согласных
5	Поменять местами слова в каждой паре слов: аб вг де жз -> вг аб жз де	15	Удалить первое и последнее слова в строках
6	Удалить все слова размером больше, чем из 5 букв	16	Удалить все согласные
7	Удалить все слова без согласных букв	17	Удалить все гласные
8	Удалить знаки препинания	18	Перемешать в случайном порядке все слова в строке
9	Удалить все слова без гласных букв	19	Удалить все слова размером меньше, чем из 5 букв
10	Удаление каждого второго слова	20	Перевести в верхний регистр все согласные

3D.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.

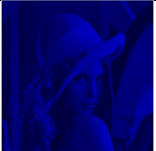
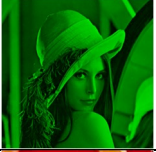

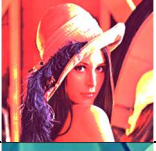

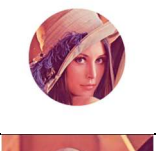
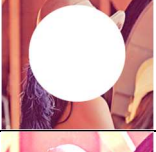

При смене активного фильтра, если входное изображение уже загружено, результирующее изображение должно обновиться автоматически.

Пример входного изображения:



Список фильтров:

Код	Описание	Пример работы
Ф1	Перевод в градации серого (усреднение всех каналов цвета)	
Ф2	Бинаризация по порогу (задается параметр Т, пиксели, яркость которых меньше Т, становятся черными, больше Т - 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	

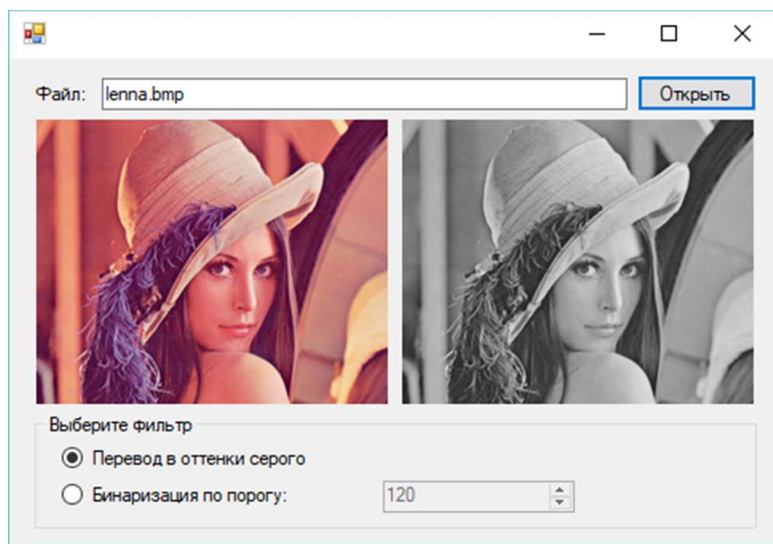
Ф8	Зануление всех каналов, кроме синего	
Ф9	Зануление всех каналов, кроме зеленого	
Ф10	Усиление черного (задается параметр Т, все пиксели меньше этого значения, становятся черными по каждому из каналов)	
Ф11	Усиление белого (задается параметр Т, все пиксели больше этого значения, становятся белыми по 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	Ф1	Ф8	6	Ф15	Ф8	11	Ф13	Ф9	16	Ф6	Ф12
2	Ф13	Ф9	7	Ф14	Ф14	12	Ф10	Ф3	17	Ф11	Ф4
3	Ф2	Ф11	8	Ф10	Ф9	13	Ф7	Ф4	18	Ф13	Ф5
4	Ф2	Ф7	9	Ф1	Ф5	14	Ф12	Ф5	19	Ф6	Ф11
5	Ф3	Ф9	10	Ф10	Ф15	15	Ф4	Ф15	20	Ф14	Ф12