

## ДЕНЬ 3С. Оконные приложения

### 3С.1. Создать форму с вводом двух чисел, вычисляющую заданное выражение

Разработать форму, которая содержит следующие элементы:

- Элемент ввода числа X
- Элемент ввода числа Y
- Кнопку «Вычислить»
- Блок вывода результата

При нажатии на кнопку «Вычислить» программа должна вычислить выражение согласно варианту и вывести результат в блоке вывода результата. Если пользователь ввел некорректные значения и нажал кнопку «Вычислить», текст ошибки должен отобразиться в поле вывода результата.

#### Пример:

Вычислить  $X+Y$

#### Варианты

№	Выражение	№	Выражение	№	Выражение	№	Выражение
1	$\tan(x)*y$	6	$y/(x^2)$	11	$\tan(\cos(x))*y$	16	$y/(x^3)$
2	$\sin(x^2)*y$	7	$\log_2(x)\cos(y)$	12	$\sin(x)/\cos(y)$	17	$\sin(x)*y^2$
3	$(1+y)/(1+x)$	8	$\sin(x)*y$	13	$\tan(\sin(x))*y$	18	$\cos(x^2)+\sin(y)$
4	$2(x*y)^3$	9	$\cos(x)*y$	14	$\cos(x)*y^2$	19	$\tan(x^2)+y^2$
5	$\log_2(x)\sin(y)$	10	$\log_2(x^2)*y$	15	$\cos(x)/\sin(y)$	20	$\log_2(x*y)^2$

### 3С.2. Создать динамическую форму обработки чисел

Создать форму содержащую:

- Поле ввода имени файла
- Кнопку «Открыть»
- Блок отображения чисел (пустой *TableLayoutPanel*)

Изначально блок отображения чисел пуст. После того, как пользователь введет имя входного файла и нажмет кнопку «Открыть», необходимо отобразить числа из файла, а также результат их обработки в блоке отображения чисел.

Каждое число в блоке отображения чисел должно быть в отдельном элементе *Label*, которые динамически создаются и добавляются в *TableLayoutPanel*. Последнее число – результат обработки – должно отображаться жирным шрифтом

Для динамического создания *Label* используйте оператор *new*:

```
Label element = new Label();
element.Text = "My text"
```

Для добавления элемента в *TableLayoutPanel* используйте

```
<Экземпляр TableLayoutPanel>.Controls.Add(element, <Колонка>, <Строка>)
```

Для очистки *TableLayoutPanel* используйте

```
<Экземпляр TableLayoutPanel>.Controls.Clear()
```

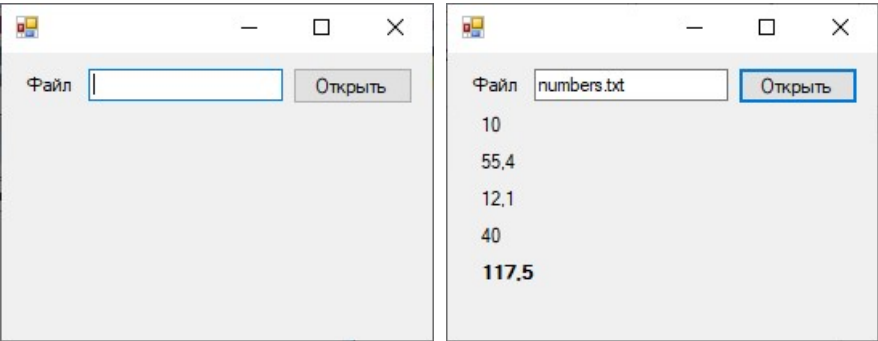
Для автоматического изменения размеров *TableLayoutPanel* настройте соответствующим образом свойства *AutoSize* и *AutoSizeMode*.

**Пример**

Просуммировать числа из файла

Входной файл *numbers.txt*:

```
10
55.4
12.1
40
```



**Варианты**

№	Операция	№	Операция
1	Перемножить дробные части всех чисел	11	Просуммировать все числа, которые меньше, чем предыдущее число
2	Просуммировать дробные части всех чисел	12	Вычислить среднее из квадратов чисел
3	Умножить числа на их порядковые номера и сложить результаты	13	Просуммировать все четные числа
4	У каждого второго числа поменять знак и просуммировать все числа	14	Просуммировать все нечетные числа
5	Просуммировать все числа, которые больше, чем предыдущее число	15	У каждого второго числа поменять знак и посчитать среднее

6	Перемножить все отрицательные числа	16	Разделить все числа на первое из чисел и вычислить сумму результатов
7	Умножить числа на их порядковые номера и вычислить среднее	17	Округлить числа и сложить результаты
8	Вычислить среднее значение	18	Перемножить все положительные числа
9	Просуммировать все числа, которые больше первого из чисел	19	Просуммировать квадраты всех чисел
10	Извлечь квадратный корень из суммы квадратов чисел	20	Просуммировать модули всех чисел

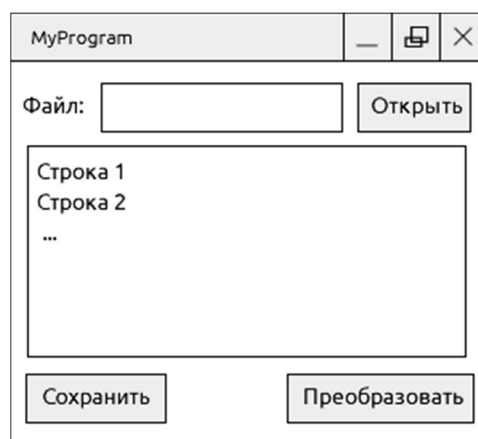
### 3С.3. Создать форму с выбором редактируемой строки в файле

Создать форму, содержащую:

- Поле ввода имени файла
- Кнопку «Открыть»
- Блок содержимого (*ListBox*)
- Кнопку «Сохранить»
- Кнопку «Преобразовать»

Изначально кнопки «Сохранить» и «Преобразовать» неактивны.

После ввода имени файла и нажатия на кнопку «Открыть» содержимое указанного файла отображается в блоке содержимого. Каждая строка файла выводится в виде отдельного элемента внутри *ListBox*

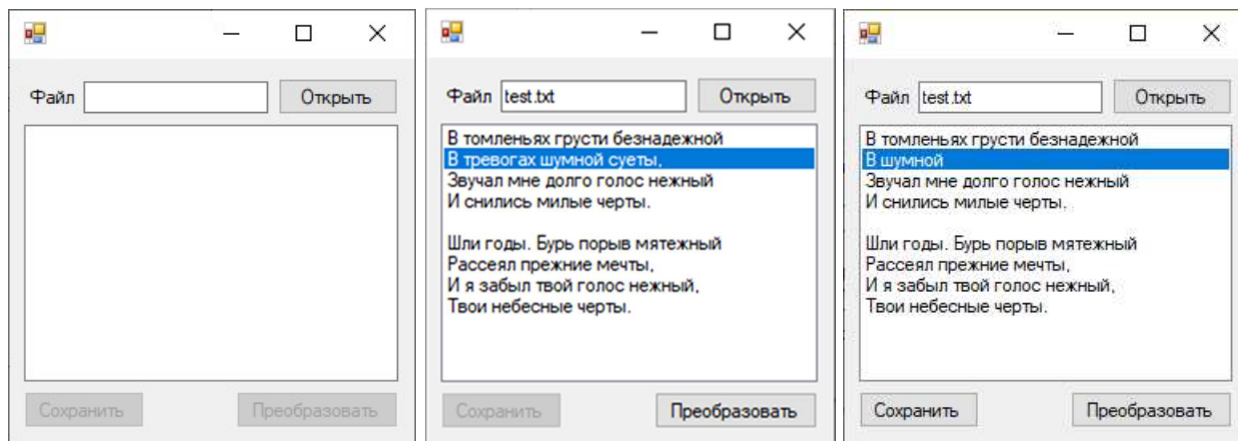


После выбора строки становится активной кнопка «Преобразовать», которая выполняет действие над строкой по варианту. После преобразования выделение должно остаться на той же самой строке.

После первого преобразования кнопка «Сохранить» становится активна (если перезагрузить файл, нажав «Открыть», то кнопка снова должна заблокироваться). При нажатии на эту кнопку преобразованный текст должен сохраниться в файл, указанный в строке ввода имени файла.

### Пример

Удаление каждого второго слова



### Варианты

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

### 3С.4. Создать программу для обработки списка случайных целых чисел

Создать форму состоящую из:

- блока случайных чисел  
(*FlowLayoutPanel*)
- блока команд

Блок чисел изначально наполняется

случайным количеством (от 5 до 10) случайных чисел (в диапазоне от -100 до 100).

Каждое число представляется в виде элемента *Label* и добавляется в *FlowLayoutPanel* с помощью:

```
<Экземпляр FlowLayoutPanel>.Controls.Add(<Элемент>)
```

Блок команд состоит из двух кнопок: команда обработки одного числа и команда обработки всех чисел.

Кнопка команды обработки одного числа изначально выключена. Для того, чтобы воспользоваться этой командой, необходимо выбрать число, нажав на него кнопкой мыши. Выбранное число должно выделяться жирным. Если пользователь нажмет на другое число, то выделение должно перейти на него, т.е. одновременно может быть выделено только одно число. После выполнения команды обработки, выделение снимается. Кроме этого, командой также можно воспользоваться **дважды кликнув** на число, которое требуется обработать (событие *DoubleClick*)

Команда обработки всех чисел должна выдавать результат в стандартном всплывающем окне: *MessageBox.Show(...)*

### Список возможных команд обработки одного числа

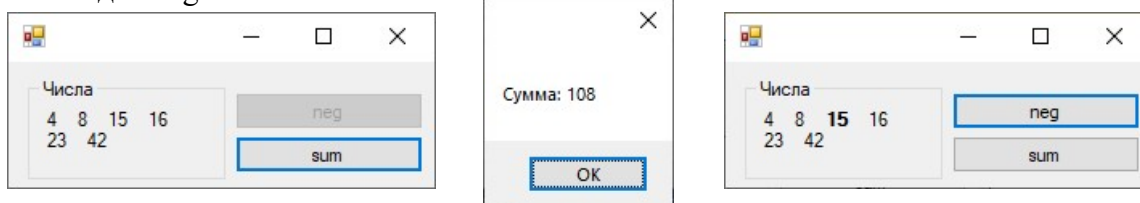
- *neg* – меняет знак у числа
- *delete* – удаляет число из списка
- *copy* – добавляет в конец списка копию числа
- *twice* – увеличивает число вдвое
- *rand* – меняют значение числа на другое случайное в диапазоне от -100 до 100

### Список возможных команд обработки всех чисел:

- *sum* – выводит сумму чисел
- *sumeven* – выводит сумму четных чисел
- *sumodd* – выводит сумму нечетных чисел
- *max* – выводит максимальное из чисел
- *min* – выводит минимальное из чисел
- *avg* – выводит среднее из чисел
- *disp* – выводит разницу между максимальным и минимальным числами

### Пример

Команды: *neg* и *sum*



### Варианты

№	Одно число	Все числа	№	Одно число	Все числа
1	delete	min	11	neg	sumodd
2	rand	sumeven	12	copy	sumodd
3	twice	avg	13	neg	disp
4	neg	min	14	delete	sum
5	neg	avg	15	twice	sum
6	twice	avg	16	twice	sumeven
7	rand	sumodd	17	delete	min
8	rand	sum	18	copy	disp
9	copy	sumeven	19	rand	max
10	delete	max	20	copy	max

### 3С.5. Создать динамическую форму фильтрации данных

Создать форму, предназначенную для фильтрации множества структур и содержащую следующие блоки:

- Блок фильтров
- Блок отображения данных (*ListBox*)
- Блок статуса (*StatusStrip*)

Блок фильтров содержит переключатели (*CheckBox*), какие фильтры в данный момент активны и отображаются (см. пример ниже). Изначально ни один фильтр не активен и, соответственно, ни один фильтр не отображается. Для скрытия фильтров можно поместить каждый из них в элемент *Panel* и указать *False* в свойство *Visible*. При активации фильтра он сразу начинает действовать и любые изменения в значении фильтра так же сразу обновляют отображаемый список структур.

Важно, чтобы при изменении количества фильтров, **блок отображения данных соответствующим образом растягивался или сжимался**, занимая все свободное место. Для того этого настройте соответствующим образом свойства *AutoSize*, *AutoSizeMode*, *Dock* у элементов

Набор фильтров зависит от состава структуры (см. вариант). Каждое поле – отдельный фильтр:

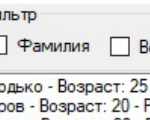
- Если поле *строковое*, то фильтр отображается в виде элемента *TextBox*. Если значение в *TextBox* не вписано, то фильтр не применяется. Если значение есть, то осуществляется поиск по подстроке (т.е. останутся только те записи, в которых в соответствующем поле присутствует вписанная подстрока)
- Если поле *целое*, то фильтр отображается в виде двух элементов *NumericUpDown* (от и до). В списке должны остаться только записи, у которых соответствующее поле попадает в указанный диапазон
- Если поле *вещественное*, то фильтр отображается в виде двух элементов *NumericUpDown* (от и до) с шагом 0.01 (свойство *Increment*). В списке должны остаться только записи, у которых соответствующее поле попадает в указанный диапазон
- Если поле *логическое*, то фильтр отображается в виде элемент *CheckBox*. В списке должны остаться только записи, у которых соответствующее поле равно значению фильтра

Блок отображения данных показывает список структур, которые попадают под фильтр. Общий перечень структур статичен и при запуске программы всегда одинаковый. При выводе структур в списке необходимо показать значения всех их полей.

Блок статуса показывает количество отображенных записей в формате: «Найдено: <число>»

### Пример

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



Фильтр

☐ Фамилия ☐ Возраст ☐ Результат

Походько - Возраст: 25 - Результат: 6  
Петров - Возраст: 20 - Результат: 5,3  
Иванов - Возраст: 22 - Результат: 7,6  
Сидоров - Возраст: 20 - Результат: 5  
Семенов - Возраст: 50 - Результат: 40

Найдено: 5

## Варианты