

## ДЕНЬ 4. ОКОННЫЕ ПРИЛОЖЕНИЯ 1

### 4.1. 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	$\log_2(x*y)^2$

### 4.2. 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()
```

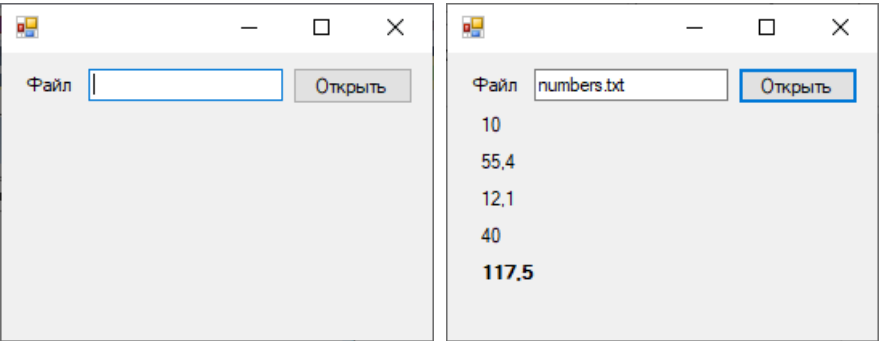
Для автоматического изменения размеров *TableLayoutPanel* настройте соответствующим образом свойства *AutoSize* и *AutoSizeMode*.

**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

### 4.3. 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 знаки препинания	11	Удалить все слова без гласных букв
2	Перевести первую букву каждого слова в верхний регистр	12	Удалить первое и последнее слова в строках
3	Удалить все слова размером больше, чем из 5 букв	13	Перемешать в случайном порядке все слова в строке
4	Перевернуть строки (абг деж -> жед гба)	14	Удалить все слова без согласных букв
5	Поменять местами слова в каждой паре слов: аб вг де жз -> вг аб жз де	15	Удалить все гласные
6	Перевернуть каждое слово в строке (абг деж -> гба жед)	16	Удалить все согласные
7	Удалить все слова, в которых меньше 5 согласных	17	Удалить все слова размером меньше, чем из 5 букв
8	Перевести все буквы первого слова каждой строке в верхний регистр	18	Удаление каждого второго слова
9	Перевести в верхний регистр каждую вторую букву слов: абв гдеж -> аБв гДеЖ	19	Удалить все слова, в которых больше 5 согласных
10	Перевести в верхний регистр все гласные	20	Перевести в верхний регистр все согласные

#### 4.4. Тест с различными типами вопросов

Вам необходимо разработать программу для тестирования знаний на заданную тему, посвященную прикладному программированию и языку C#. Тест должен состоять из 3-ех вопросов, формулировку которых вам нужно придумать самостоятельно. Каждый из вопросов должен иметь уникальный тип ответа:

1. С одним верным вариантом ответа
2. С несколькими верными ответами (2-3)
3. С текстовым вводом ответа

Программа должна выводить вопросы **в случайном порядке** (для этого используйте класс *Random*). После ответа на последний вопрос должен отображаться блок с результатом теста.

В один момент времени на форме должен отображаться только один текущий вопрос. Для скрытия неактивных вопросов, можно поместить каждый из них в элемент *Panel* и указать *False* в свойство *Visible*. На форме для каждого вопроса должно присутствовать:

- Номер вопроса (с точки зрения отвечающего)
- Текст вопроса
- Блок с ответом

В зависимости от типа вопроса блок с ответом должен иметь свой вид (см. картинки примера ниже)

**Для вопроса с одним верным вариантом ответа** варианты должны быть представлены в виде кнопок (*Button*). Порядок расположения вариантов ответов с каждым запуском должен случайным образом меняться. При нажатии на кнопку ответ фиксируется и открывается следующий вопрос (или блок результата)

**Для вопроса с несколькими верными ответами** варианты должны быть представлены в виде элементов *CheckBox* с отдельной кнопкой отправки ответа. Порядок расположения вариантов ответов с каждым запуском должен случайным образом меняться. Если ни один из вариантов ответа не выбран, кнопка отправки ответа должна быть неактивной. После выбора вариантов и нажатия на кнопку отправки ответ фиксируется и открывается следующий вопрос (или блок результата).

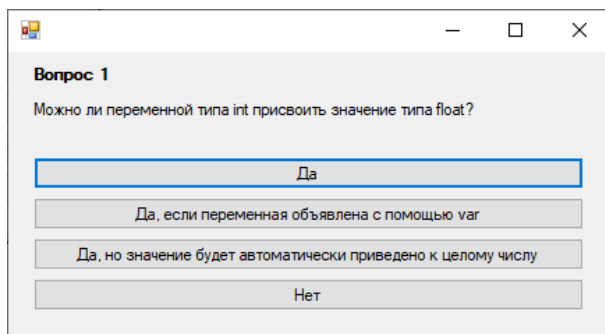
**Для вопроса с текстовым вводом ответа** блок ответа должен быть представлен в виде элемента *TextBox* с кнопкой отправки ответа. Если строка ответа пуста, то кнопка отправки ответа должна быть неактивной. После ввода ответа и нажатия на кнопку отправки ответ фиксируется и открывается следующий вопрос (или блок результата).

Блок результата должен содержать элемент *Label* для показа процента правильных ответов и кнопку «Начать заново». Ответ считается правильным, если он в точности совпадает с требуемым (т.е. если верных вариантов 2, а пользователь правильно выбрал только один, то этот вопрос ему не засчитывается). После нажатия на кнопку «Начать заново» тест должен повториться, **при этом все вопросы и ответы должны быть случайно перемешаны снова.**

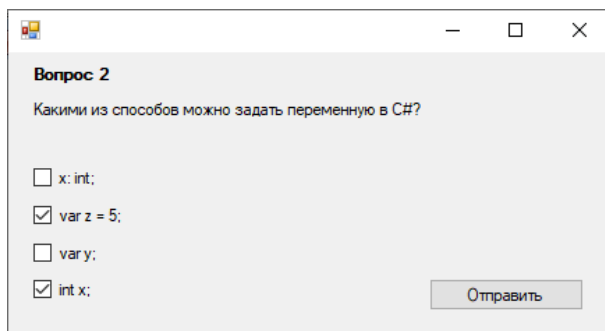
## Пример

Тема: переменные

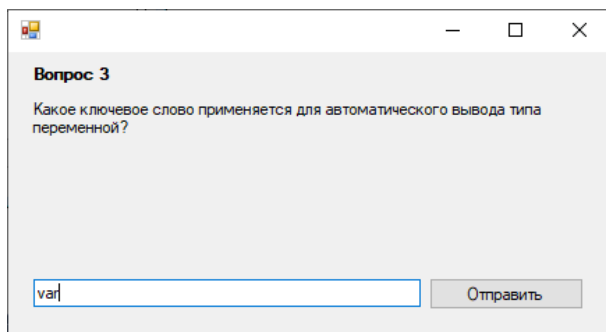
Вопрос с одним верным вариантом ответа



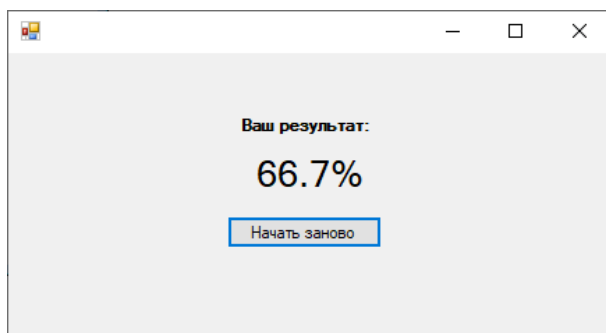
Вопрос с несколькими верными вариантами ответа



## Вопрос с текстовым вводом ответа



Блок результата:



### Варианты

№	Преобразование	№	Преобразование
1	Функции и методы в C#	11	UML-диаграммы
2	Операторы языка C#	12	Обработка файлов
3	Стандартная библиотека C#	13	Массивы в C#
4	Обобщения (generics) в C#	14	Типы данных C#
5	Объектно-ориентированное программирование	15	Ключевые слова C#
6	Этапы разработки ПО	16	Классы в C#
7	Механизм исключений	17	Тестирование
8	Коллекции в C#	18	Форматирование вывода в C#
9	Структуры в C#	19	Преобразование типов в C#
10	Отличия C# от C	20	Механизм событий

## 4.5. 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

Фильтр

☒ Фамилия ☐ Возраст ☒ Результат

Фамилия

Результат  -

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

Найдено: 3

## Варианты

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