

Дополнительные задания

Д.1. Создать форму, отображающую список записей из CSV-файла

Создать форму, содержащую:

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

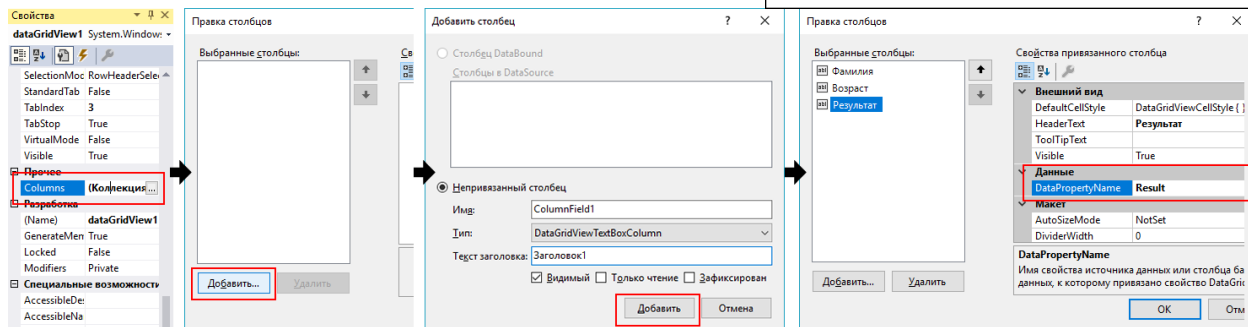
При нажатии на кнопку «Открыть» программа должна открыть указанный CSV-файл и вывести все записи в таблицу отображения данных. Если файл не найден или его невозможно открыть, вывести сообщение об ошибке во всплывающем окне

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.

MyProgram

Файл: Открыть

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



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

Пример

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

Файл: Открыть

	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)

Д.2. Создать программу просмотра отчета по нескольким CSV-файлам

Создать форму, содержащую следующие элементы (см. пример):

- кнопку «Открыть папку»
- дерево с данными (*TreeView* со свойством *Checkboxes = true*)
- блок вывода результата

Изначально блоки дерева данных и результата пустые.

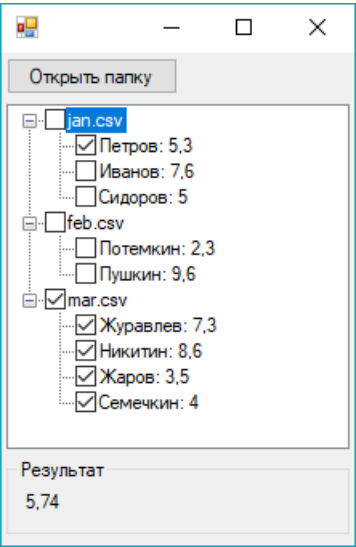
При нажатии на кнопку «Открыть папку» должен отобразиться стандартный диалог выбора папки (*FolderBrowserDialog*). После выбора папки программа должна считать все CSV-файлы в указанной директории и сформировать дерево с данными: узлы первого уровня – названия CSV-файлов, к которым прикреплены узлы второго уровня, соответствующие записям внутри файла. Название узла второго уровня формируется по шаблону:

<первое поле записи>: <поле, участвующее в отчете (см. вариант)>

При выборе записей в дереве данных автоматически должен рассчитываться отчет по вариантам и его результат записываться в блок результата. Выбор узла с названием файла эквивалентен выбору всех его записей

Пример

Запись: фамилия (строка), возраст (целое число), результат (вещественное число).
Вывести среднее значение результата.



Входной файл *jan.csv*: Входной файл *mar.csv*:

Петров, 20, 5.3 Журавлев, 18, 7.3
Иванов, 22, 7.6 Никитин, 33, 8.6
Сидоров, 20, 5 Жаров, 22, 3.5
 Семечкин, 55, 4

Входной файл *feb.csv*:

Потемкин, 19, 2.3
Пушкин, 32, 9.6

Варианты

№	Запись	Отчет
1	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)	Сумма разниц между суммами поступления и списания
2	Название материала (строка), объем (вещественное число), вес (вещественное число)	Сумма объемов
3	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)	Сумма разниц между планом выпуска деталей и фактическим выпуском
4	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)	Сумма разниц между числом товара на складе и числом зарезервированных товаров
5	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)	Среднее число сезонов
6	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)	Сумма значений пробега
7	Дисциплина (строка), номер курса (целое число), количество часов (целое число)	Среднее количество часов

8	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)	Среднее значение гонорара
9	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)	Среднее значение номера этажа
10	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)	Сумма балансов
11	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)	Среднее число выполненных заданий
12	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)	Среднее число уникальных посетителей
13	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)	Сумма весов
14	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)	Среднее значение среднего балла
15	Фамилия (строка), должность (строка), оклад в руб (целое число)	Сумма окладов
16	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)	Среднее значение всех оценок
17	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)	Среднее значение средней цены
18	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)	Среднее число слов
19	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)	Сумма сумм заказов
20	Фамилия (строка), рост (вещественное число), вес (вещественное число)	Среднее значение роста

Д.3. 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 отобразиться 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).

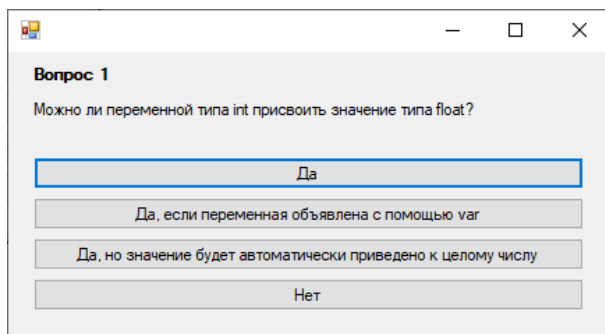
Для вопроса с текстовым вводом ответа блок ответа должен быть представлен в виде элемента *TextBox* с кнопкой отправки ответа. Если строка ответа пуста, то кнопка отправки ответа должна быть неактивной. После ввода ответа и нажатия на кнопку отправки ответ фиксируется и открывается следующий вопрос (или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

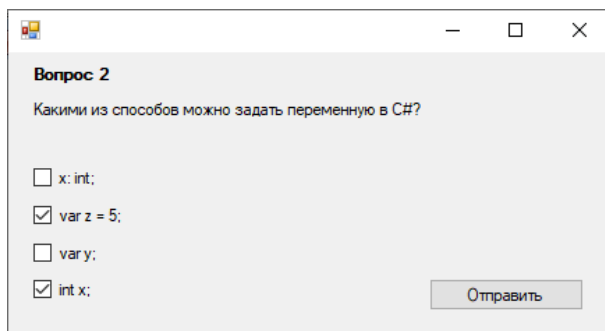
Пример

Тема: переменные

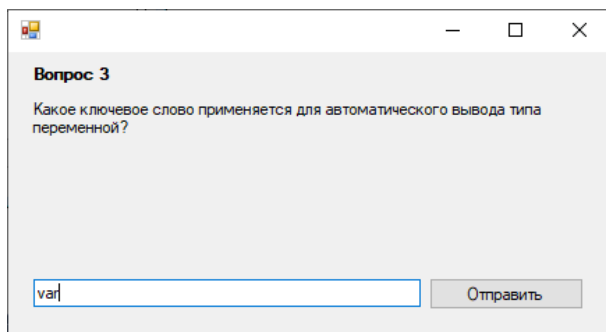
Вопрос с одним верным вариантом ответа



Вопрос с несколькими верными вариантами ответа



Вопрос с текстовым вводом ответа



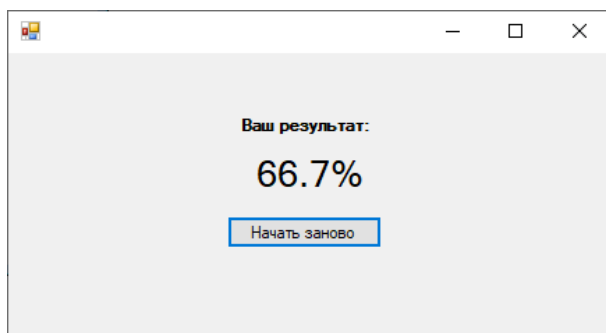
Вопрос 3

Какое ключевое слово применяется для автоматического вывода типа переменной?

var

Отправить

Блок результата:



Ваш результат:

66.7%

Начать заново

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

Д.4. Создать программу с окном настройки, генерирующую набор чисел и выполняющую заданную операцию

Создать программу, состоящую из двух форм: основное окно и модальное окно настроек.

Основное окно содержит следующие элементы (см. пример):

- элемент отображения прогресса генерации чисел (*ProgressBar*)
- список сгенерированных чисел (*ListBox*)
- кнопку «Настроить»
- блок результата

Модальное окно настроек содержит следующие элементы (см. пример):

- слайдер выбора периода (*TrackBar*)
- элемент, отображающий значение периода текстом
- кнопка «Отмена»
- кнопка «Ок»

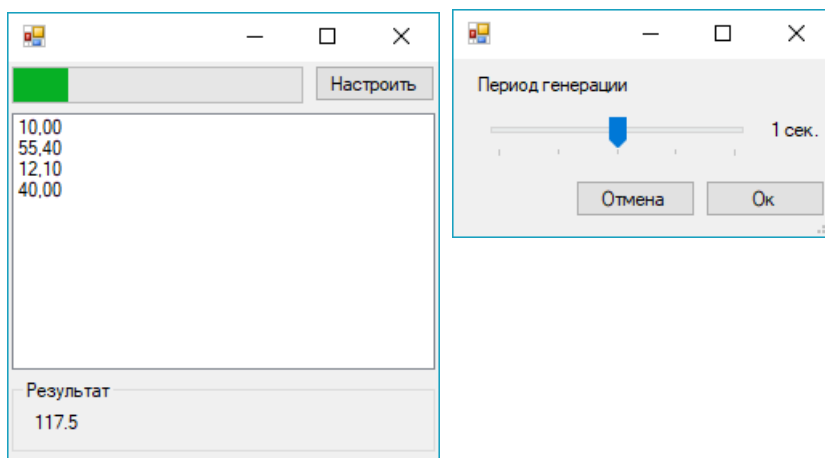
Изначально список сгенерированных чисел пуст. При старте программы должен быть запущен таймер, который при каждом срабатывании добавляет новое случайное число в список из диапазона от -100.00 до 100.00 (с двумя случайными цифрами после запятой). После каждого добавления числа программа должна применить заданную по вариантам операцию к списку чисел и отобразить вычисленное значение в блоке результата.

По мере добавления новых чисел должен обновляться элемент прогресса генерации. Он должен достигнуть 100%, когда будет сгенерировано 20 чисел. После этого работа таймера прекращается.

Период срабатывания таймера по умолчанию равен 1 секунде. При нажатии на кнопку «Настроить» должно отображаться **модальное** окно настроек, в котором с помощью слайдера выбора периода можно выбрать одно из следующих значений: 0.25 сек., 0.5 сек., 1 сек., 2 сек. или 4 сек. Изменение слайдера должно сопровождаться обновлением текста в элементе, отображающем значение периода текстом. При нажатии на кнопки «Ок» или «Отмена» модальное окно должно закрыться и, если была нажата кнопка «Ок», то должен обновиться период таймера добавления чисел.

Пример

Просуммировать числа



Варианты

№	Операция	№	Операция
1	Округлить числа и сложить результаты	11	Извлечь квадратный корень из суммы квадратов чисел
2	У каждого второго числа поменять знак и просуммировать все числа	12	Просуммировать все нечетные числа
3	Просуммировать все четные числа	13	Умножить числа на их порядковые номера и вычислить среднее
4	Просуммировать все числа, которые больше первого из чисел	14	Перемножить дробные части всех чисел
5	Просуммировать модули всех чисел	15	Умножить числа на их порядковые номера и сложить результаты

6	Разделить все числа на первое из чисел и вычислить сумму результатов	16	Просуммировать дробные части всех чисел
7	Перемножить все положительные числа	17	Просуммировать все числа, которые больше, чем предыдущее число
8	Вычислить среднее из квадратов чисел	18	Вычислить среднее значение
9	У каждого второго числа поменять знак и посчитать среднее	19	Перемножить все отрицательные числа
10	Просуммировать квадраты всех чисел	20	Просуммировать все числа, которые меньше, чем предыдущее число

Д.5. Создать программу отображения и редактирования данных CSV-файла, состоящую из трех форм

Разработать программу, состоящую из трех форм: окно приветствия, основное окно и окно редактирования/добавления записи.

Окно приветствия должно содержать следующие элементы (см. пример):

- кнопку «Открыть файл...»
- кнопку «Создать файл...»
- кнопку «Выход»

Основное окно должно содержать (см. пример)::

- блок списка записей (*ListBox*)
- кнопку «Редактировать»
- кнопку «Добавить запись»
- кнопку «Сохранить»

Окно редактирования/добавления записи должно содержать (см. пример):

- 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
- кнопку «Отмена»
- кнопку «Ок»

При запуске программы сначала отображается окно приветствия.

При нажатии на «Открыть файл...» должен отобразиться стандартный диалог открытия файла, в котором доступны для выбора только файлы с расширением .csv. После выбора файла форма приветствия закрывается, открывается основное окно, в котором в блоке списка записей отображаются данные CSV-файла

При нажатии на «Создать файл...» должен отобразиться стандартный диалог сохранения файла с расширением по умолчанию .csv. После выбора файла форма приветствия закрывается, открывается основное окно с пустым списком записей.

При нажатии на «Выход» программа завершает свое выполнение.

В основном окне кнопка «Редактировать» изначально неактивна. Она становится активной, как только будет выбран какой-либо из элементов списка записей. Записи отображаются в списке в виде текста путем объединения всех полей записи в одну строку через символ «|».

При нажатии на кнопки «Добавить запись» или «Редактировать» должно отображаться окно редактирования/добавления **не** в модальном режиме (т.е. можно одновременно редактировать несколько записей). При нажатии на кнопку «Редактировать», поля во всплывающем окне должны быть заполнены текущими данными записи.

Кнопка «Сохранить» должна сохранить текущий список записей в открытый или созданный CSV-файл.

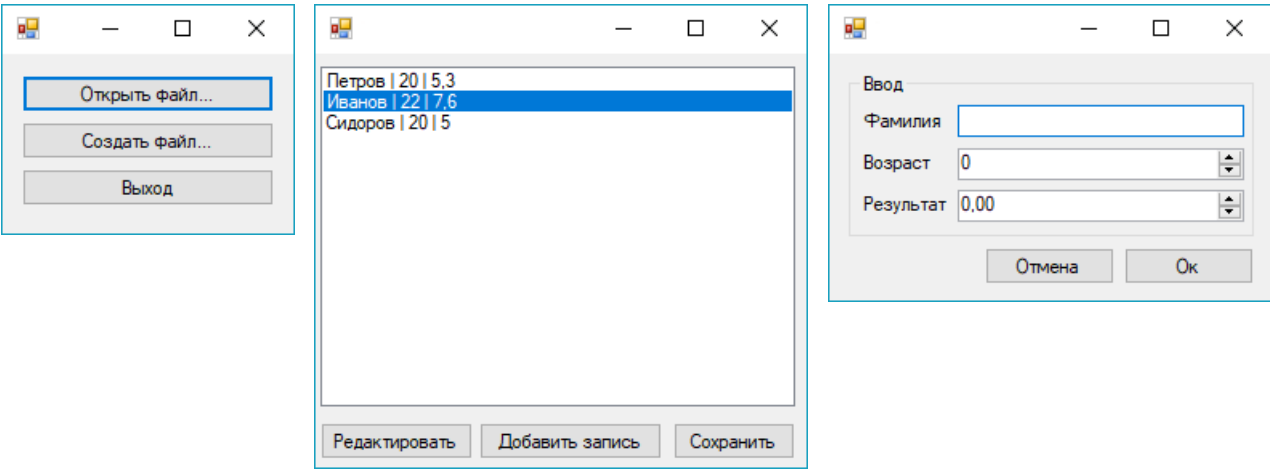
В окне редактирования/добавления кнопка «Отмена» приводит к закрытию окна без сохранения введенных данных; кнопка «Ок» приводит к закрытию окна с обновлением записи в основной форме (или добавлением в конец списка).

Пример

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

Входной файл:

Петров, 20, 5.3
Иванов, 22, 7.6
Сидоров, 20, 5



Варианты

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