# Теоретическая часть

* Определение текущей строки и ячейки по координатам клика мыши в **DataGridView** – по материалам домашней работы
* Формирование строки **DataGridView** при помощи классов **DataGridViewRow**, **DataGrodViewCell** – по материалам домашней работы
* Использование компонента **BindingSource** для упрощения создания столбцов **DataGridView** в визуальном конструкторе
* Использование компонента **BindingSource** во время исполнения кода для заполнения **DataGridView** данными коллекции с использованием свойства **DataSource**
* Реализация CRUD-операций с доступом к коллекции данных через **BindingSource**, обновление отображения коллекции в **DataGridView**

# Практическая часть

**Задача 1.** Требуется спроектировать и разработать приложение-макет для продажи билетов на поезд. Это должно быть приложение Windows Forms с использованием изученных элементов интерфейса (**DataGridView**, **BindingSource**).

Разработайте классы для прикладной части приложения.

Класс Поезд, хранящий коллекцию объектов класса Вагон. Максимальное количество вагонов в поезде 20. Поезд характеризуется номером (цифры и буквы), пунктом отправления и пунктом назначения.

Класс Вагон характеризуется номером вагона, количеством купе, массивом купе, максимальной вместимостью купе (типом купе).

Класс Купе характеризуется типом (СВ, купе, плацкарт), количеством мест, списком мест.

Класс Место характеризуется номером и состоянием – продано или свободно (не продано).

В главной форме разместить **TreeView**, отображающий поезд. Корнем дерева является номер поезда, узлами – вагоны, подузлами – купе, листьями – места.

В **DataGridView** по клику в TreeView на вагон выводить текущее состояние мест выбранного вагона.

Учтите, что Вагоны всегда состоят из 9 купе, отличаются количеством мест в купе:

* СВ – 2 места
* Купейный – 4 места
* Плацкартный – 6 мест

При помощи меню, контекстных меню, кнопок панели инструментов реализуйте следующий функционал:

* добавление вагона (тип вагона задаем в меню или соответствующей кнопкой)
* удаление вагона
* покупка билета
* возврат билета
* сохранение данных поезда в файл (JSON)
* чтение данных поезда из файла (JSON)

И сохранение в файл и чтение из файла реализуйте при помощи диалогов выбора файлов и техники Drag'n'Drop.

В строку состояния главного окна выводить информацию:

1. Всего вагонов в поезде
2. Всего мест в поезде/продано билетов
3. СВ: вагонов/мест/продано билетов
4. купейный: вагонов/мест/продано билетов
5. плацкарт: вагонов/мест/продано билетов

В меню и контекстных меню по Вашему выбору включите команды выхода, вывода окна со сведениями о программе и ее разработчике.

# Дополнительно

Запись занятия можно скачать [**по этой ссылке**](https://cloud.mail.ru/public/8nqh/Ww2ENJtZR), материалы занятия в этом же архиве.