# **Windows Forms.**

# **Теоретическая часть**

* Использование события **Validating** для проверки значения поля ввода
* Использование свойства **Tag** для хранения произвольных данных в элементе интерфейса
* Понятие об использовании переключателя **RadioButton**
* Контейнеры для переключателей – **GroupBox**, **Panel**

# **Практическая часть**

**Задача 1.** Разработайте приложение Windows Forms для вычислений параметров объемных геометрических фигур по выбору пользователя. Каждый расчет выполняется на собственной форме, открываемой в модальном режиме. Выбор формы – при помощи кнопочного меню.

По команде **О программе** кнопочного меню в отдельной форме выводить в Label сведения о приложении и разработчике (фамилия, имя, группа). При помощи таймера реализуйте закрытие формы «О программе» через 10 секунд после открытия.

В формах фигур разместить строки ввода для параметров фигур, чек-боксы для задания вычисляемых параметров фигуры, изображение фигуры, изображение материала фигуры. Изображения выводить в **Paintbox**, Вы без проблем найдете изображения в интернете.

Собственно вычисления выполнять при клике на кнопку "Вычислить". Необходимо контролировать правильность ввода параметров, использовать **ErrorProvider**. Используйте модели для вычислений.

Что именно вычислять для каждой фигуры:

* площадь поверхности
* объем
* масса

Типы фигур для расчетов:

* усеченный конус
* сфера
* цилиндр
* прямоугольный параллелепипед

Виды материалов, из которых создаются фигуры (расширения номенклатуры материалов не предвидится 😊):

* сталь (нержавеющая)
* медь
* водяной лед
* гранит

# **Дополнительно**

Запись занятия можно скачать [**по этой ссылке**](https://cloud.mail.ru/public/ftDr/vXTsP93ZJ). Материалы занятия – в прикрепленном файле.

## C#. Теоретическая часть

* Понятие о формате JSON
* Атрибуты сериализации [DataContract], [DataMember]
* Сериализация данных в формате JSON
* Понятие о NuGet-пакетах, NuGet-репозитории, менеджере пакетов NuGet

## Практическая часть

**Задача 2.**Напишите консольное приложение для учета заявок на авиабилеты. Каждая заявка содержит: номер заявки, пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета. Для хранения данных использовать класс **List<>**.

Приложение должно обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

* формирование коллекции заявок, сериализация коллекции в формате JSON
* десериализаця из файла в формате JSON в список заявок
* вывод всех заявок из коллекции в консоль
* выборка в еще один список заявок по заданному номеру рейса и дате вылета при помощи именованного итератора;
* добавление заявки в список, сериализация коллекции в формате JSON;
* удаление заявок из списка по номеру заявки, сериализация коллекции в формате JSON;
* удаление всех заявок из списка, сериализация коллекции в формате JSON;
* упорядочивание списка заявок по номеру рейса, сериализация коллекции в формате JSON
* упорядочивание списка заявок по желаемой дате рейса, сериализация коллекции в формате JSON

## Дополнительно

Запись занятия можно скачать [**по этой ссылке**](https://cloud.mail.ru/public/ftDr/vXTsP93ZJ). Материалы задания – в архиве.