Практическое задание: **Система бронирования**  
  
Цель: Разработать консольное приложение на языке C# для управления системой бронирования мест в гостинице.  
  
Задачи:  
  
1. Определение структуры данных:  
    - Создайте класс `Room` с полями: `Number`, `Type` (одноместный, двухместный и т.д.), `Price` и `IsBooked`.  
    - Создайте класс `Booking` с полями: `BookingId`, `GuestName`, `RoomNumber`, `CheckInDate` и `CheckOutDate`.  
  
2. Базовый интерфейс:  
    - При запуске приложения пользователь должен видеть меню с следующими опциями: `1. Просмотреть все номера`, `2. Бронировать номер`, `3. Отменить бронь`, `4. Выход`.  
  
3. Просмотр номеров:  
    - При выборе соответствующей опции, пользователь должен видеть список всех номеров, их тип, стоимость и статус (забронирован или свободен).  
  
4. Бронирование номера:  
    - При выборе опции бронирования, пользователь вводит свое имя, выбирает номер и указывает даты прибытия и отъезда. После этого номер помечается как забронированный.  
    - Учтите проверку на доступность выбранного номера в указанные даты.  
  
5. Отмена брони:  
    - Пользователь может ввести идентификатор брони, чтобы отменить ее. После отмены номер считается свободным.  
  
6. Хранение данных:  
    - Используйте коллекции (например, `List<T>`) для хранения данных о номерах и бронированиях.  
    - Для продвинутых: реализуйте сохранение и загрузку данных из файла.  
  
7. Обработка ошибок:  
    - Добавьте базовую обработку ошибок, чтобы приложение корректно реагировало на неверные вводы пользователя.  
  
Дополнительные задачи для продвинутого уровня:  
  
1. Расширенный поиск: Позвольте пользователю фильтровать номера по типу, стоимости или датам доступности.  
  
2. Логирование: Реализуйте запись всех действий пользователя (бронирование, отмена) в отдельный файл-лог.  
  
3. Динамическое меню: При добавлении новых функций в приложение, меню должно автоматически обновляться.  
  
Это задание позволит студентам практиковаться в основных концепциях программирования на C#, включая работу с классами, коллекциями, обработкой исключений и вводом/выводом данных