Практическое задание: **Семейный бюджет**  
  
Цель: Разработать программу для учета и планирования семейного бюджета.  
  
Задачи:  
  
1. Базовая структура:  
    - Создайте класс `Transaction`, который будет содержать информацию о транзакции: дата, сумма, категория, тип (доход или расход) и комментарий.  
    - Создайте коллекцию для хранения всех транзакций.  
  
2. Добавление транзакций:  
    - Реализуйте функцию для добавления новой транзакции в коллекцию, запрашивая у пользователя необходимую информацию.  
  
3. Отображение истории транзакций:  
    - Пользователь должен иметь возможность просматривать историю транзакций, отсортированную по дате.  
    - Реализуйте фильтрацию транзакций по категории или дате.  
  
4. Статистика:  
    - Покажите пользователю общую сумму доходов и расходов за выбранный период (например, за месяц).  
    - Визуализируйте распределение расходов по категориям.  
  
5. Планирование бюджета:  
    - Позвольте пользователю устанавливать бюджет на месяц.  
    - Сравнивайте фактические траты с планируемым бюджетом и предупреждайте пользователя, если он приближается к лимиту или его превышает.  
  
6. Сохранение и загрузка данных:  
    - Реализуйте возможность сохранять историю транзакций и установленный бюджет в файл.  
    - Добавьте функцию загрузки данных из файла при старте программы.  
  
7. Интерфейс пользователя:  
    - Создайте простое текстовое меню для управления программой: добавление транзакции, просмотр истории, статистика, установка бюджета и т. д.  
  
Дополнительные задачи для более опытных студентов:  
  
1. Реализуйте графический интерфейс пользователя с использованием, например, WPF.  
2. Добавьте функциональность для учета регулярных платежей (например, аренда жилья, коммунальные услуги).  
3. Реализуйте возможность ввода данных о доходах и расходах в разной валюте с автоматическим конвертированием по актуальному курсу, используя внешние API для получения курса валют.  
  
Это задание позволит начинающим программистам практиковаться в работе с данными, создании интерфейса пользователя, файловой системе, а также в интеграции с внешними сервисами (если выполняется дополнительное задание).