**Практика 11. Создание и удаление объектов**

1. В классе банковский счет, созданном в предыдущих упражнениях, удалить методы заполнения полей. Вместо этих методов создать конструкторы. Переопределить конструктор по умолчанию, создать конструктор для заполнения поля баланс, конструктор для заполнения поля тип банковского счета, конструктор для заполнения баланса и типа банковского счета. Каждый конструктор должен вызывать метод, генерирующий номер счета.
2. Создать новый класс BankTransaction, который будет хранить информацию о всех банковских операциях. При изменении баланса 75 счета создается новый объект класса BankTransaction, который содержит текущую дату и время, добавленную или снятую со счета сумму. Поля класса должны быть только для чтения (readonly). Конструктору класса передается один параметр – сумма. В классе банковский счет добавить закрытое поле типа System.Collections.Queue, которое будет хранить объекты класса BankTransaction для данного банковского счета; изменить методы снятия со счета и добавления на счет так, чтобы в них создавался объект класса BankTransaction и каждый объект добавлялся в переменную типа System.Collections.Queue.
3. В классе банковский счет создать метод Dispose, который данные о проводках из очереди запишет в файл. Не забудьте внутри метода Dispose вызвать метод GC.SuppressFinalize, который сообщает системе, что она не должна вызывать метод завершения для указанного объекта.
4. В класс Song (из Практики 10, № 6) добавить следующие конструкторы: 1) параметры конструктора – название и автор песни, указатель на предыдущую песню инициализировать null. 2) параметры конструктора – название, автор песни, предыдущая песня. В методе Main создать объект mySong. Возникнет ли ошибка при инициализации объекта mySong следующим образом: Song mySong = new Song(); ? Исправьте ошибку, создав необходимый конструктор.

<https://metanit.com/sharp/tutorial/3.35.php>