Задачи по C#

1. **Снова фильтры**

Переписать консольное приложение по обработке изображений различными фильтрами на C#. В программе нельзя использовать классы из пространства имён System.Drawing.

Написать к приложению юнит-тесты, проверяющие все фильтры, а также случай, когда пользователь ввёл что-то не то. Все данные для юнит-тестов должны быть включены в проект юнит-тестов. В юнит-тестах можно использовать пространство имён System.Drawing.

1. **Случай в казино**

Реализовать одну из азартных игр. Игра к реализации определяется как 1 + остаток от деления суммы длин имени, фамилии и отчества на 3.

* 1. Реализовать иерархию классов для игры в блэкджек с базовым вариантом правил и 8 замешанными колодами (<https://en.wikipedia.org/wiki/Blackjack>).
  2. Реализовать иерархию классов для игры в рулетку с 1 зеро и ставками на цвет, чётность, дюжину или конкретное число.
  3. Реализовать иерархию классов для игры в баккара, вариант punto banco (<https://en.wikipedia.org/wiki/Baccarat_(card_game)>).

Написать для игры ботов, реализующих как минимум 2 разные стратегии на усмотрение автора и имеющих некоторую стартовую сумму денег. Проверить корректность работы юнит-тестами. С их же помощью показать, сколько в среднем денег остаётся у каждого бота после 400 ставок.

1. **Простая иерархия**

Реализовать на выбор одну из иерархий объектов:

а) Создать абстрактный класс "Танк" с некоторым набором характеристик на ваше усмотрение (вооружение, броня, страна-производитель и т.д.), а также методом или свойством позволяющим получить полную, агрегированную информацию о модели. Реализовать описание нескольких конкретных моделей автомобилей.

б) Создать абстрактный класс "Мороженое" с некоторым набором набором характеристик на ваше усмотрение (тип (сливочное, фруктовый лёд), вкус, количество шариков (если применимо) и т.д.), а также методом или свойством позволяющим получить рецепт его приготовления на основе этих характеристик. Реализовать описание нескольких конкретных тортов.

В обоих случаях разнести базовый класс и реализации по отдельным библиотекам. В основной программе выводить в консоль требуемую информацию через методы или свойства абстрактного класса. Написать юнит-тесты, проверяющие корректность работы всех классов иерархии.

1. **Дженерики**

Написать с использованием дженериков класс, реализующий одну из перечисленных коллекций с возможностью добавления, удаления и поиска по ней. Возможно, в реализации потребуется сделать более, чем один класс. Коллекция к реализации определяется как 1 + остаток от деления суммы длин имени, фамилии и отчества на 4.

* Двусвязный список;
* Хэш-таблица;
* Динамический массив;
* Несбалансированное бинарное дерево поиска несбалансированное с числовым ключом.

1. **Слабые ссылки**

С помощью слабых ссылок реализовать коллекцию из предыдущей задачи, которая хранит объекты не менее задаваемого интервала времени, например, минуты. Интервал задаётся параметром конструктора. При необходимости предусмотреть освобождение ресурсов. Опустевшие элементы допустимо удалять из коллекции.

1. **Плагины (или Почти как у Bethesda)**

Реализовать программу, которая ищет по заданному пути библиотеки с плагинами, удовлетворяющими некоторому интерфейсу (определяется в отдельной библиотеке), загружает их и создаёт по одному экземпляру каждого класса, реализующего этот интерфейс.

1. **IoC**

Настроить IoC-контейнер по своему выбору, с его помощью инициализировать классы задачи 2-2 и запустить игру. Решение оформить в виде юнит-теста.

1. **UI**

Написать приложение, использующее, WinForms, которое на экранной форме строит алгебраическую кривую не ниже второго порядка из нескольких заранее определённых (оси координат и сама кривая). Как минимум у одной кривой порядка N коэффициенты как при XN, так и при YN должны быть ненулевыми. Список кривых отображается в комбобоксе. Реализовать возможность масштабирования показываемого изображения. Классы, реализующие связанную с кривыми математику, следует вынести в отдельную общую библиотеку.