

Задание 3

1. Разработать иерархию классов для решения поставленной задачи, используя возможности ООП: классы, наследование, полиморфизм, инкапсуляция, возможности интерфейсов и множественное наследование, использовать возможности Enum и внутренних классов.
2. Каждый класс должен иметь исчерпывающие смысл, название и информативный состав.
3. Классы сгруппировать по смыслу в соответствующие библиотеки классов
4. При кодировании должны быть использованы соглашения об оформлении кода [C# code convention](#).
5. В классах должны быть методы ToString(), GetHashCode() и Equals().
6. Используя классы StreamReader и StreamWriter обеспечить загрузку/сохранение данных из XML-файла
7. Используя классы XmlReader и XmlWriter обеспечить чтение/сохранение данных в XML-файл
8. Обеспечить взаимное чтение/сохранение данных классами XmlWriter и StreamReader, StreamWriter и XmlReader
9. Широкий ассортимент типов данных, различный уровень доступа к полям и методам, использование статических полей и методов, констант приветствуются.
10. У девочки есть безразмерные листы пленки, бумаги и пластика. Из них она может вырезать различные геометрические фигуры (круг, прямоугольник, равносторонний треугольник и т.п.) определенного размера, причем из одних фигур можно вырезать другие. Вырезаемая фигура должна быть меньше той, из которой ее вырезают.
11. Еще у девочки есть набор красок. Бумажные фигуры можно красить, но только 1 раз. Фигуры из пластика можно многократно перекрашивать. Фигуры из пленки бесцветные и красить их нельзя.
12. Все фигуры должны обеспечивать возможность расчета площади и периметра
13. Так же, у девочки есть коробка на 20 фигур, куда можно складывать и доставать любые фигуры.
14. У коробки есть ряд функций:
 - добавить фигуру, (нельзя добавить одну и ту же фигуру дважды)
 - просмотреть по номеру (при этом фигура остается в коробке)
 - извлечь по номеру (при этом фигура удаляется из коробки)
 - заменить по номеру
 - найти фигуру по образцу (эквивалентную по своим характеристикам)
 - показать наличное количество фигур
 - суммарную площадь
 - суммарный периметр
 - достать все Круги
 - достать все Пленочные фигуры
 - достать все Пластиковые фигуры, которые ни разу не красились
 - сохранить все фигуры / только бумажные / только пластиковые / только плёночные из коробки в XML-файл, используя StreamWriter
 - сохранить все фигуры / только бумажные / только пластиковые / только плёночные из коробки в XML-файл, используя XmlWriter
 - загрузить все фигуры в коробку из XML-файла, используя StreamReader
 - загрузить все фигуры в коробку из XML-файла, используя XmlReader
15. Требования к реализации:
 - материал и форма не могут быть полями класса
 - вырезание фигуры из другой фигуры реализовать как соотв. конструктор
 - использовать возможности Enum с цветом
 - невозможность вырезания реализовать как соответствующее исключение
 - невозможность окрашивания фигуры реализовать как соответствующее исключение
 - обязательно использовать автодокументирование
 - реализовать модульные тесты для разработанных классов