

## Java.SE.01.OOP

Создать консольное приложение:

1. Использовать возможности ООП: классы, наследование, полиморфизм, инкапсуляция.
2. Использовать возможности интерфейсов Java (множественное наследование).
3. Использовать возможности Enum.
4. Использование внутренних классов.
5. Каждый класс должен иметь исчерпывающие смысл, название и информативный состав.
6. При кодировании должны быть использованы соглашения об оформлении кода **java code convention**.
7. Классы должны быть грамотно разложены по пакетам.
8. В классах должны быть методы `toString()`, `hashCode()` и `equals()`.
9. В тестовом классе для демонстрации возможностей не вводить данные с консоли (ввод «хардкодом»), вывод в консоль только из `main`.
10. Широкий ассортимент типов данных, различный уровень доступа к полям и методам, использование статических полей и методов, констант приветствуются.
11. Добавить комментарии в программу в виде */\*\* комментарий \*/* (с использованием дескрипторов вида `@param`, `@return` при необходимости в описании методов), извлечь эту документацию в HTML-файл и просмотреть полученную страницу Web-браузером.

У девочки есть безразмерные листы пленки и бумаги. 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. Еще у девочки есть набор красок. Бумажные фигуры можно красить, но только 1 раз. Фигуры из пленки бесцветные и красить их нельзя.

Так же, у девочки есть коробка на 20 фигур, куда можно складывать и доставать любые фигуры. У коробки есть функции (см. 4 этап реализации).

### 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

### Этапы реализации

- 1) Иерархия классов фигур (Фигура, Круг, Прямоугольник, ...). Общее – возможность расчета площади и периметра, но реализация у каждой конкретной фигуры своя.
- 2) Реализация возможности вырезания одних фигур из других.
- 3) Интерфейсы-материалы (Бумага, Пленка). Материал Бумага подразумевает (как минимум) наличие функций покраски и проверки цвета. В свете требования об одноразовой покраске тут можно подумать о внутреннем классе ... Дополнение иерархии фигур классами пленочных и бумажных фигур (БумажныйТреугольник, ПленочныйТреугольник, и т.п.), дополнение возможности вырезания с учетом материала. Реализация функций покраски и проверки цвета.
- 4) Коробка для складывания фигур. Методы:
  - 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

+ полученная на основе комментариев документация  
На проверку присылать поэтапно!