**Практическая работа 2.**

**#2A**

Создайте базовый класс Shape (фигура) с защищенными полями типа double площадь, периметр и строковым полем название. В этом же классе определите конструктор по умолчанию, который будет задавать значения полям (0, 0, «без названия»). Также определите метод show(), который выводит на консоль значения полей.

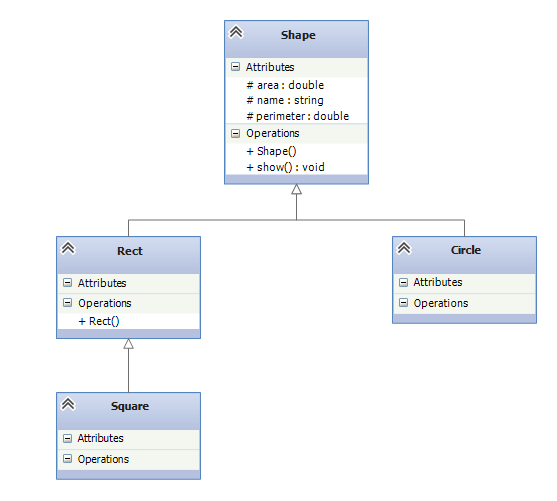
Создайте классы-наследники класса Shape – Rect (прямоугольник) и Circle (окружность), в каждом из которых определите нужные поля для этой фигуры, методы для расчета площади и периметра, а также два конструктора – по умолчанию (задающий начальные значения (0, 0, «прямоугольник») и с параметрами.

Создайте класс Square (квадрат) – наследник класса Rectangle. Определите требуемые поля. Переопределите конструктор этого класса через базовый конструктор класса Rect.

Достройте UML-диаграмму классов по проекту, представленную ниже.

Создайте консольное приложение, запрашивающее у пользователя тип фигуры и вводимые значения. Приложение выводит на консоль результаты функции.

Создайте приложение WPF, реализующее те же действия, что и консольное приложение



**2B**

**1.** Создайте Modelling Project, определяющий UML-диаграмму следующих классов:

* Класс Card (карта), который будет служить контейнером для двух полей закрытого типа: suit (масть) и rank (старшинство).

В классе должны быть определены:

* закрытый конструктор по умолчанию, чтобы не было возможности создать пустую карту;
* конструктор с параметрами.

Кроме того, в классе Card должен быть переопределен метод ToString() класса System.Object таким образом, чтобы можно было простым способом получить строковое представление любой карты (rank+suit).

* Класс Deck (колода), поддерживающий массив 52 объектов Card.

В классе должны быть определены:

* конструктор по умолчанию, задающий колоду из 52 карт;
* метод SetCard(), который присваивает элементу массива card значение (rank+suit);
* метод GetCard(), который возвращает карту по заданному индексу (из массива объектов Card);
* метод Shuffle, который позволяет расположить карты в массиве случайным образом.

**2.** Реализуйте данные классы в C#.

**3.** **Создайте консольное приложение**, позволяющее выводить на экран карты из колоды, а также перетасовывать колоду и выводить на экран перетасованные карты.

**4.** **Создайте приложение WPF**, реализующее те же действия, что и консольное приложение (пример реализации на рис.1). Реализацию классов Deck и Card изменять нельзя. Элементы управления, где будут размещены «картинки» карт, должны добавляться в контейнер (Grid, StackPanel и т.д.) в цикле кодом C#.

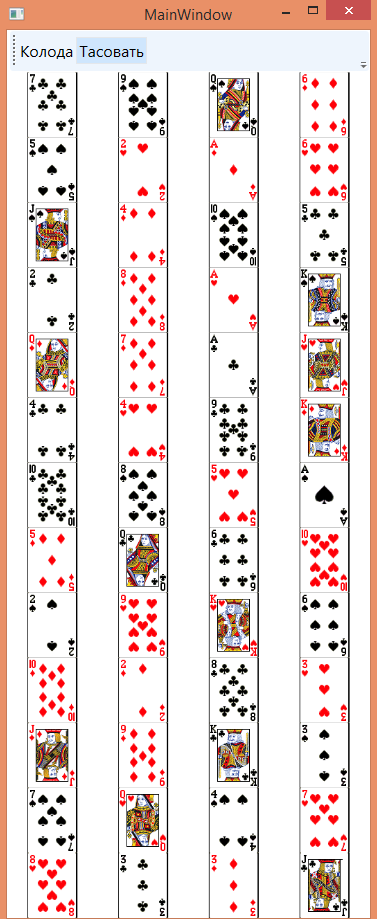
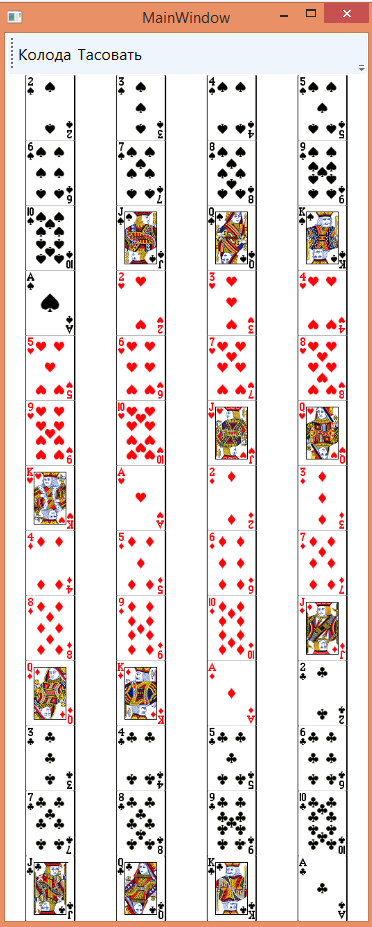


Рис.1.