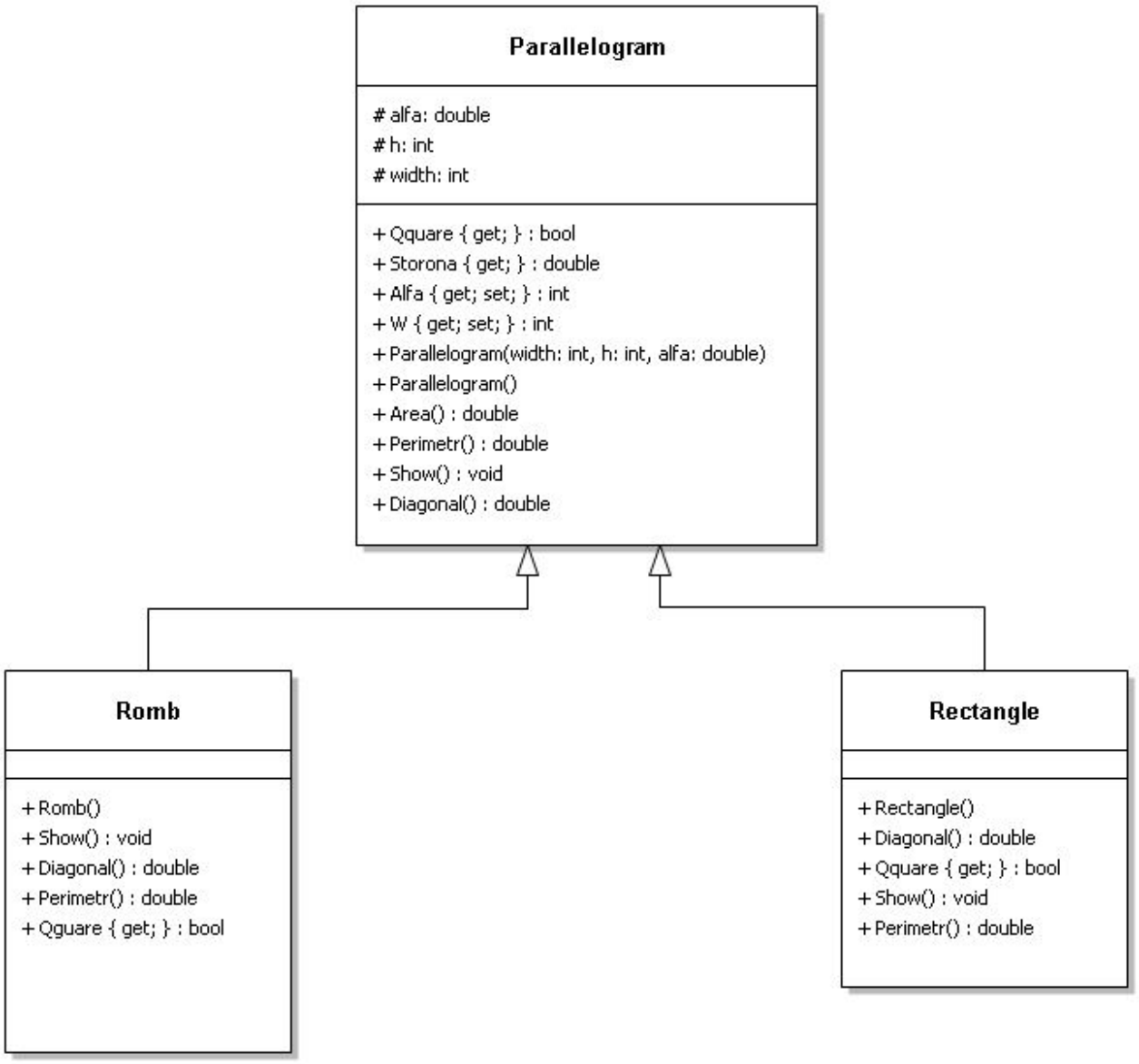
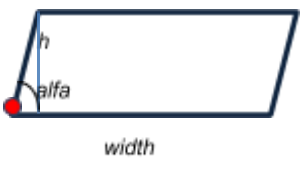
Задача №5.2 Иерархия классов с использованием виртуальных методов

Вариант №1: **«Геометрические фигуры: Четырехугольники»**

Создать иерархию классов по приведенной UML-диаграмме. **Схема является лишь частичной, ВНИМАТЕЛЬНО читайте описание классов.** Элементы, отмеченные знаками '-' – private, '#' – protected, '+' – public.



Необходимо создать класс **Parallelogram**, который определяет тип геометрической фигуры – параллелограмм. Параллелограмм задаётся стороной **width**, высотой **h** и углом **alfa**.



Класс **Parallelogram** имеет:

- свойства, которое позволяют обращаться и изменять поля,
- свойство **isQquare**, которое определяет, является ли параллелограмм прямоугольником

- метод **Storona()**, вычисляющий сторону параллелограмм по известным значениям высоты h и углу alfa,
- метод **Area()**, вычисляющий площадь параллелограмма.
- метод **Show()**, который выводит на экран значения заданных полей
- метод **Perimetr()**, вычисляющий периметр параллелограмма
- метод **Diagonal()**, считающий меньшую диагональ параллелограмма

ПОМНИТЕ: *внутри конструкторов производных классов поля базового класса НЕ создаются, они должны получить свои значения с помощью вызова конструктора базового класса! В производных же классах можно только уточнять поля базового класса, если это действительно необходимо.*

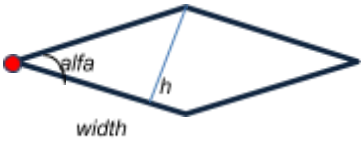
Класс **Rectangle** будет наследовать класс **Parallelogram**, он задаётся шириной **width** и высотой **h**, а угол **alfa** всегда должен быть равен 90.

Класс **Rectangle** имеет:

- конструктор, задающий поля базового класса под свои особенности
- переопределенный метод **Show()**, который выводит на экран значения заданных полей и уточняет, что мы имеем дело с прямоугольником
- переопределенный метод **Diagonal()**, считающий диагональ прямоугольника по известным width и h
- переопределенное свойств**о isQquare**, которое определяет, является ли прямоугольник квадратом

ПОМНИТЕ: *внутри конструкторов производных классов поля базового класса НЕ создаются, они должны получить свои значения с помощью вызова конструктора базового класса! В производных же классах можно только уточнять поля базового класса, если это действительно необходимо.*

Класс **Romb** также будет наследовать класс **Parallelogram**, задаётся шириной **width**, высотой **h** и углом **alfa**. Все стороны ромба равны и это надо проверять в момент его создания (в конструкторе).



Класс **Romb** имеет:

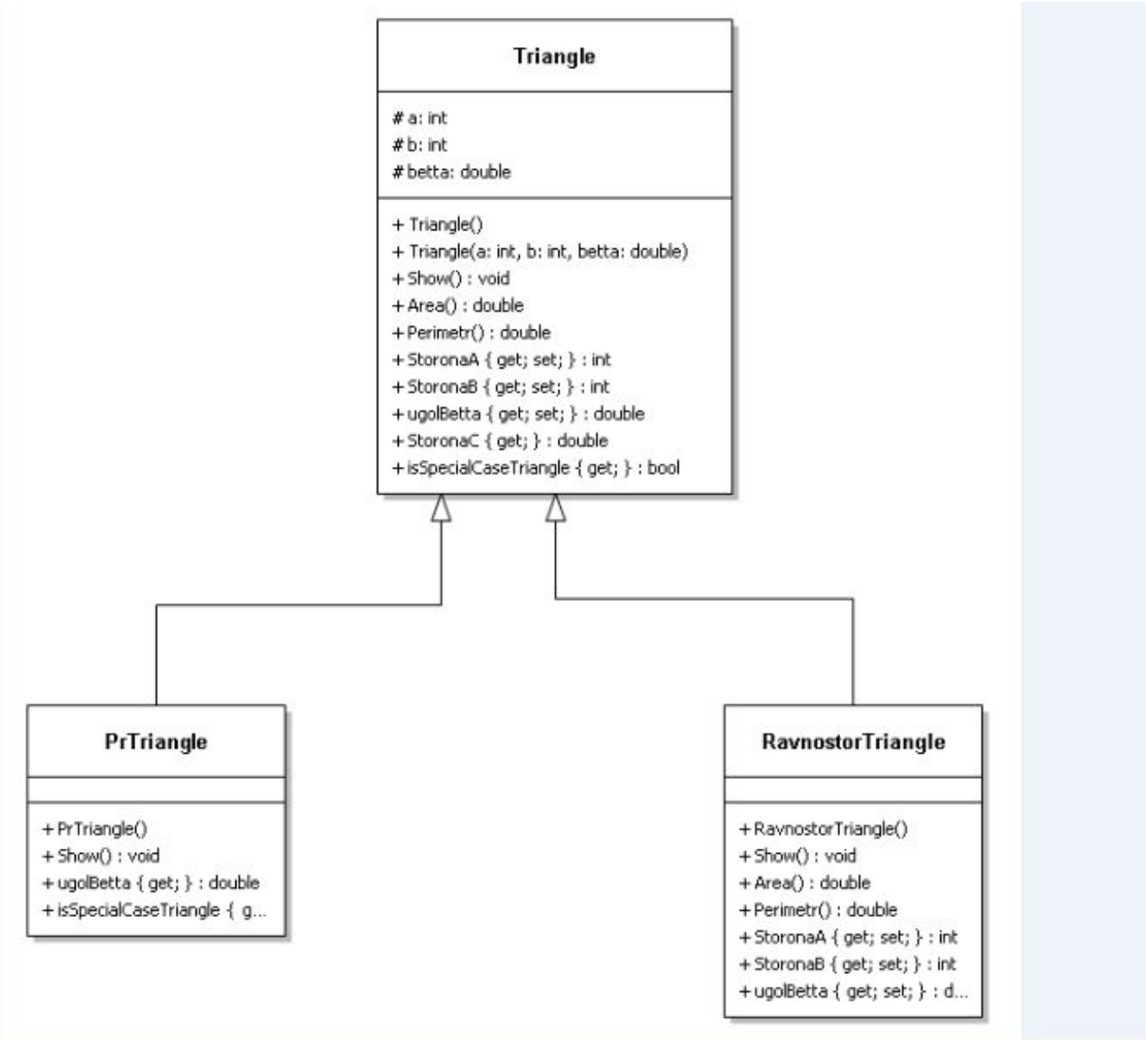
- конструктор, задающий поля базового класса под свои особенности (у ромба все стороны равны)
- метод **Show()**, который выводит на экран значения заданных полей и уточняет, что мы имеем дело с ромбом
- переопределенный метод **Diagonal()**, считающий большую диагональ ромба
- переопределенный метод **Perimetr()**, вычисляющий периметр параллелограмма по одной стороне
- переопределенное свойство **isQquare**, которое определяет, является ли ромб квадратом.

В Main() нужно продемонстрировать работу со всеми классами, являющимися последними в цепочке наследования. Доступ к объектам осуществлять через ссылку на базовый класс!!!

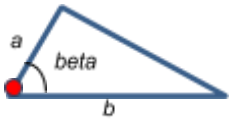
Создать массив ссылок на базовый класс и привязать к ним объекты производных классов.

Вариант №2: **«Геометрические фигуры: Треугольники»**

Создать иерархию классов по приведенной UML-диаграмме. **Схема является лишь частичной, ВНИМАТЕЛЬНО читайте описание классов.** Элементы, отмеченные знаками '-' – private, '#' – protected, '+' – public.



Создать класс **Triangle**, который определяет тип геометрической фигуры – треугольник. Треугольник задаётся двумя сторонами **a**, **b** и углом **beta** между ними.



Класс **Triangle** теперь должен иметь:

- свойства, которое позволяют обращаться и изменять поля,
- свойство **StoronaC**, вычисляющий третью сторону треугольника по двум известным сторонам и углу между ними,

- свойство **isSpecialCaseTriangle**, определяющее является ли треугольник равнобедренным,
- свойство **MiddleLine**, определяющее среднюю линию треугольника (*Средняя линия треугольника равна половине его основания и параллельна ему*),
- метод **Area()**, вычисляющий площадь треугольника,
- метод **Show()**, который выводит на экран значения заданных сторон и угла треугольника,,
- метод **Perimetr()**, вычисляющий периметр треугольника.

ПОМНИТЕ: *внутри конструкторов производных классов поля базового класса НЕ создаются, они должны получить свои значения с помощью вызова конструктора базового класса! В производных же классах можно только уточнять поля базового класса, если это действительно необходимо.*

Класс **PrTriangle** – прямоугольный треугольник, в свою очередь, наследует класс Triangle и имеет следующую особенность: угол beta всегда равен 90 – даже если пользователь захочет его переобозначить, класс содержит:

- конструктор класса, который принудительно задает угол beta, всегда равный 90
- перегруженное свойство **isSpecialCaseTriangle**, определяющее, является ли прямоугольный треугольник равнобедренным *(Равнобедренный прямоугольный треугольник имеет равные углы при основании (гипотенузе). Каждый из этих углов содержит 45°.),*
- перегруженное свойство **поля betta**, которое контролирует угол beta (его нельзя изменять**)**
- перегруженный метод **Show()**, который уточняет, что треугольник прямоугольный

ПОМНИТЕ: *внутри конструкторов производных классов поля базового класса НЕ создаются, они должны получить свои значения с помощью вызова конструктора базового класса! В производных же классах можно только уточнять поля базового класса, если это действительно необходимо.*

Класс **RavnostorTriangle** – равносторонний треугольник - тоже наследует класс Triangle и имеет следующую особенность: угол beta всегда равен 60 и сторона b = a, класс содержит:

- конструктор класса, который принудительно задает угол beta, всегда равный 60, и поля a и b одинаковыми
- перегруженное свойство **поля betta**, которое контролирует угол beta и не дает его изменять
- перегруженнЫЕ свойствА на поля a и b, не дающее поменять ни одну из сторон, не проверив, что другая имеет тоже значение
- перегруженный метод **Show()**, который уточняет, что треугольник равносторонний
- перегруженный метод **Area()**, вычисляющий площадь равностороннего треугольника (формула должна отличаться от формулы площади просто треугольника)
- перегруженный метод **Perimetr()**, вычисляющий периметр равностороннего треугольника (формула должна отличаться от формулы периметра треугольника с разными сторонами, достаточно одной стороны)

В Main() нужно продемонстрировать работу со всеми классами, являющимися последними в цепочке наследования. Доступ к объектам осуществлять через ссылку на базовый класс!!!

Создать массив ссылок на базовый класс и привязать к ним объекты производных классов.