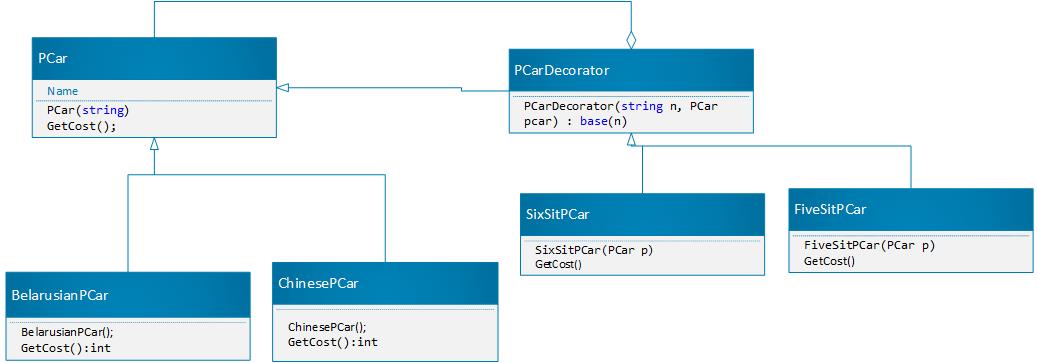


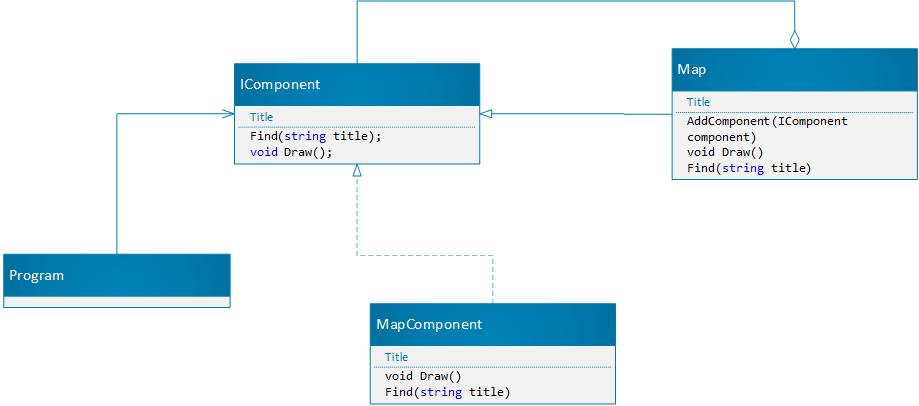
* **IPath**: представляет объекты, которые используются клиентом
* **Driver**: использует объекты IPath для реализации своих задач
* **Helicopter**: представляет адаптируемый класс, который мы хотели бы использовать у клиента вместо объектов IPath
* **HelicopterToTransportAdapter**: собственно адаптер, который позволяет работать с объектами Helicopter как с объектами IPath.



* **PCar**: абстрактный класс, который определяет интерфейс для наследуемых объектов
* **BelarusianPCar and ChinesePCar**: конкретная реализация компонента, в которую с помощью декоратора добавляется новая функциональность
* **PCarDecorator**: собственно декоратор, реализуется в виде абстрактного класса и имеет тот же базовый класс, что и декорируемые объекты. Поэтому базовый класс PCar должен быть по возможности легким и определять только базовый интерфейс.

Класс декоратора также хранит ссылку на декорируемый объект в виде объекта базового класса PCar и реализует связь с базовым классом как через наследование, так и через отношение агрегации.

* Классы **SixSitPCar** и **FiveSitPCar** представляют дополнительные функциональности, которыми должен быть расширен объект BelarusianPCar and ChinesePCar.



* **IComponent**: определяет интерфейс для всех компонентов в древовидной структуре
* **Map**: представляет компонент, который может содержать другие компоненты и реализует механизм для их добавления и удаления
* **MapComponent**: представляет отдельный компонент, который не может содержать другие компоненты
* **Program**: клиент, который использует компоненты