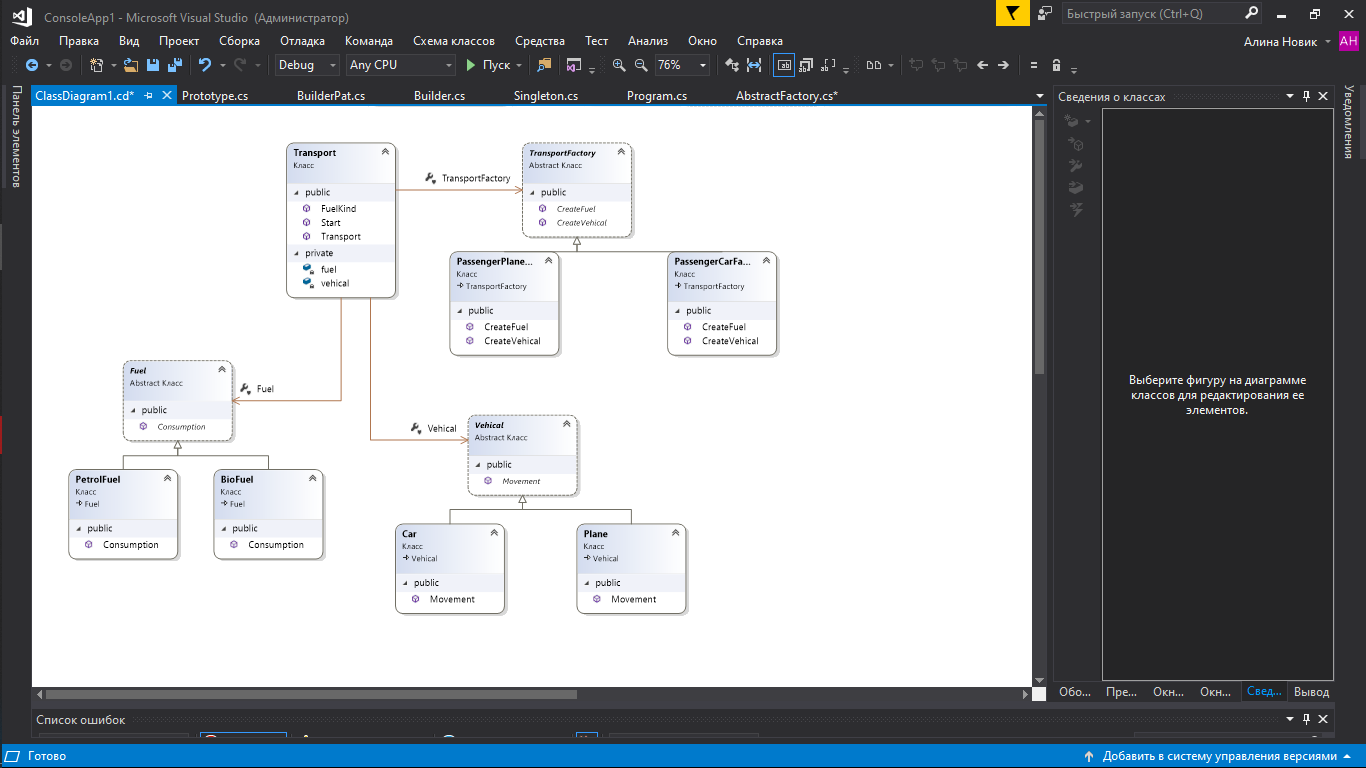
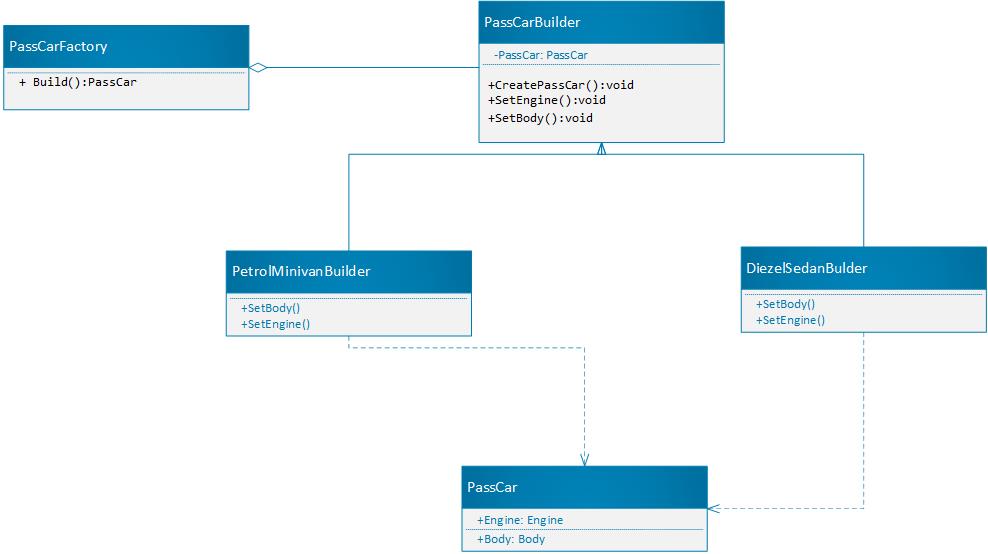
Abstract Factory



Паттерн определяет следующих участников:

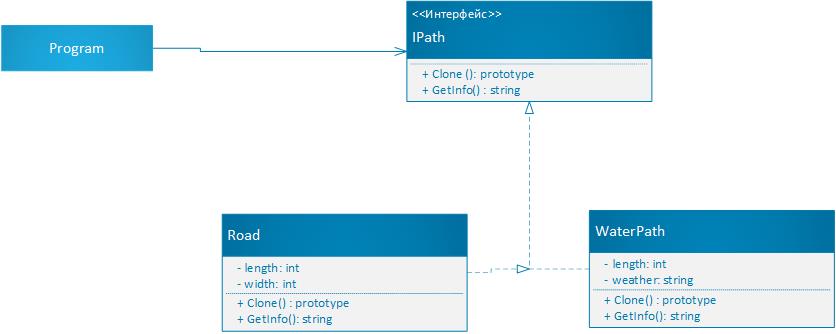
* Абстрактные классы **Fuel** и **Vehical** определяют интерфейс для классов, объекты которых будут создаваться в программе.
* Конкретные классы **PetrolFuel/ BioFuel** и **Car/ Plane** представляют конкретную реализацию абстрактных классов
* Абстрактный класс фабрики **TransportFactory** определяет методы для создания объектов. Причем методы возвращают абстрактные продукты, а не их конкретные реализации.
* Конкретные классы фабрик **PassengerCarFactory** и **PassengerPlaneFactory** реализуют абстрактные методы базового класса и непосредственно определяют какие конкретные продукты использовать
* Класс клиента **Transport** использует класс фабрики для создания объектов. При этом он использует исключительно абстрактный класс фабрики TransportFactory и абстрактные классы продуктов Fuel и Vehical и никак не зависит от их конкретных реализаций

Builder



* **PassCar**: представляет объект, который должен быть создан. В данном случае все части объекта заключены в списке parts.
* **PassCarBuilder**: определяет интерфейс для создания различных частей объекта Product
* **PetrolMinivanBuilder and DiezelSedanBuilder**: конкретная реализация PassCarBuildera. Создает объект PassCar и определяет интерфейс для доступа к нему
* **PassCarFactory**: распорядитель - создает объект, используя объекты PassCarBuilder

Protorype



* **IPath**: определяет интерфейс для клонирования самого себя, который, как правило, представляет метод Clone()
* **Road**и **WaterPath**: конкретные реализации прототипа. Реализуют метод Clone()
* **Program**: создает объекты прототипов с помощью метода Clone()

Singleton

