1. **Что такое UML?**

язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнеспроцессов, системного проектирования

1. **Перечислите типы диаграмм UML.**

Диаграмму вариантов использования

Диаграмму классов

Диаграмму последовательности

Диаграмму деятельности

Диаграмму компонентов или/и размещения

Диаграмма автомат

Диаграмма коммуникации

Диаграмма компонентов

Диаграмма синхронизаци

1. **Какие отношения между классами могут быть на диаграмме классов?**

**ассоциация** между классами - Ассоциация показывает отношения между объектами-экземплярами класса.[*Ассоциация*](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) показывает, что объекты одной сущности (класса) связаны с объектами другой сущности таким образом, что можно перемещаться от объектов одного класса к другому.

►**обобщение** между классами ; Обобщение показывает, что один из двух связанных классов (*подтип*) является частной формой другого (*надтипа*), который называется **обобщением** первого.

►**зависимости** (различных типов) - [Зависимость](https://ru.wikipedia.org/w/index.php?title=%D0%97%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C_(UML)&action=edit&redlink=1)[[en]](https://en.wikipedia.org/wiki/Dependency_(UML)) обозначает такое отношение между классами, что изменение спецификации класса-поставщика может повлиять на работу зависимого класса, но не наоборот.

[***Агрегация***](https://ru.wikipedia.org/wiki/%D0%90%D0%B3%D1%80%D0%B5%D0%B3%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))— это разновидность ассоциации при отношении между целым и его частями.

[*Композиция*](https://ru.wikipedia.org/w/index.php?title=%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D1%8F_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)&action=edit&redlink=1) — более строгий вариант агрегации. *Композиция* имеет жёсткую зависимость времени существования экземпляров класса контейнера и экземпляров содержащихся классов.

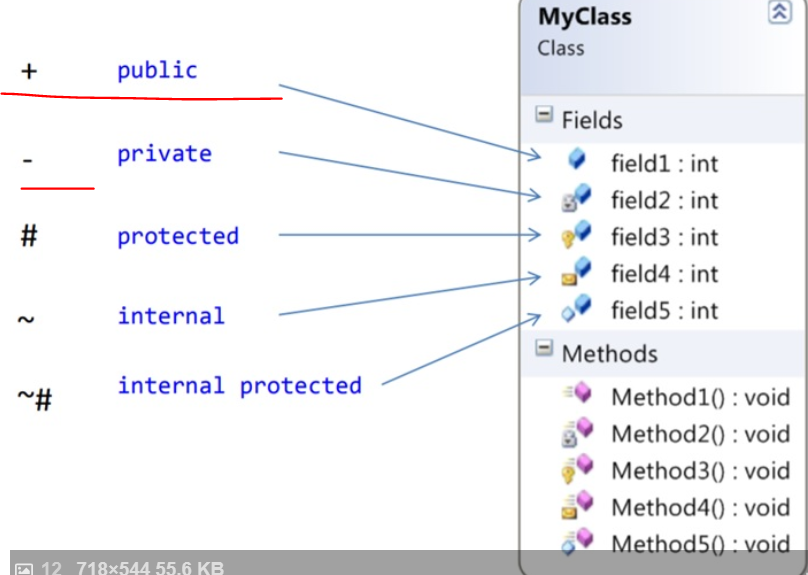
1. **Как обозначаются абстрактные классы на диаграмме классов?**

Если класс абстрактный — то его имя пишется полужирным курсивом.

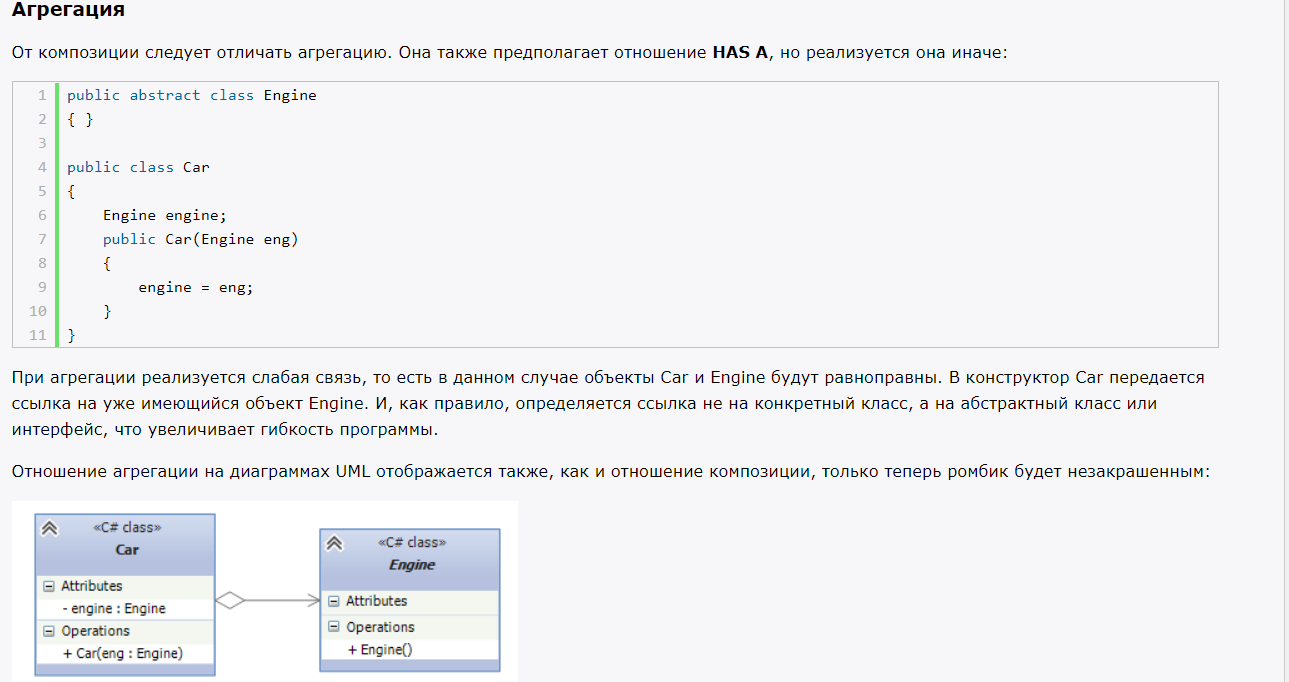
1. **Как обозначаются интерфейсы на диаграмме классов**?

**Интерфейс обозначается** ключевым словом «interface»

1. Как отображается доступность членов класса на диаграмме классов?



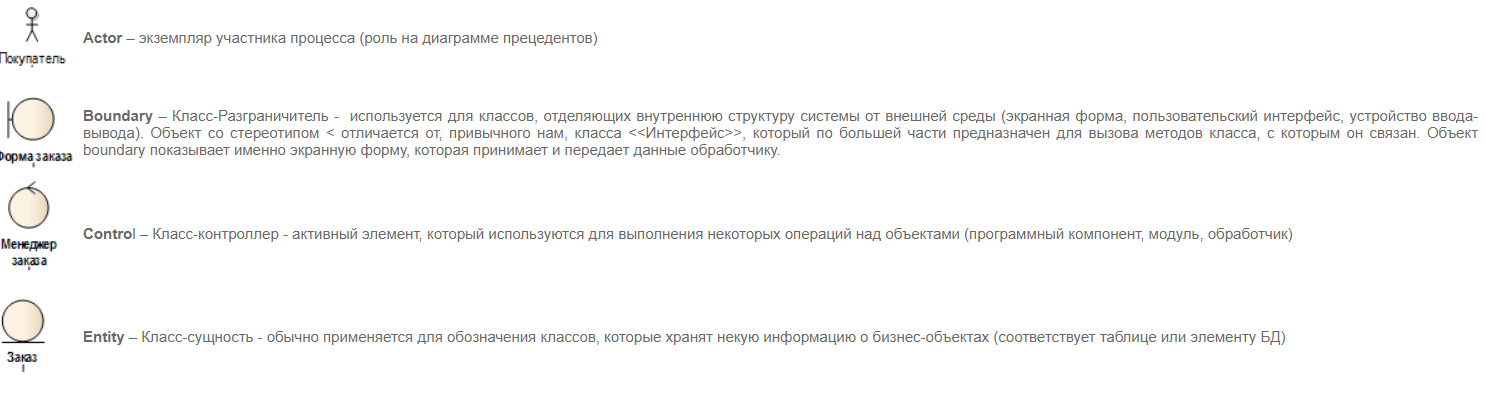
1. **Что такое агрегация? Как обозначается?**



**8. Что такое ассоциация?**

Есть выше

1. Какие обозначения используют на диаграмме последовательности?





1. **Для чего нужна диаграмма последовательности?**

Диаграмма последовательности (sequence diagram) ‒ это способ описания поведения системы на основе указания последовательности передаваемых сообщений.

1. **Каково назначение диаграммы использования, пакетов и активности?**

Проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых прецедентов. При этом актером (actor) или действующим лицом называется любая сущность, взаимодействующая с системой извне. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой.

**Пакеты применяют** , чтобы сгруппировать классы , обладающие некоторой общностью.

**Диаграмма деятельности** (activity diagram активности) ‒ способ описания поведения на основе указания потоков управления и потоков данных