1. **Что такое UML?**

UML (Unified Modeling Language) — это стандартный язык для визуального моделирования систем. Он используется для спецификации, визуализации, разработки и документирования различных аспектов программных систем. UML помогает разработчикам и аналитикам описывать архитектуру и дизайн системы на различных уровнях абстракции.

1. **Перечислите типы диаграмм UML.**

UML включает несколько типов диаграмм, которые можно разделить на две большие категории: структурные и поведенческие диаграммы.

* + **Структурные диаграммы:**
    - Диаграмма классов (Class Diagram)
    - Диаграмма объектов (Object Diagram)
    - Диаграмма компонентов (Component Diagram)
    - Диаграмма развёртывания (Deployment Diagram)
    - Диаграмма пакетов (Package Diagram)
    - Диаграмма композитной структуры (Composite Structure Diagram)
    - Диаграмма профилей (Profile Diagram)
  + **Поведенческие диаграммы:**
    - Диаграмма использования (Use Case Diagram)
    - Диаграмма активности (Activity Diagram)
    - Диаграмма состояний (State Machine Diagram)
    - Диаграмма последовательности (Sequence Diagram)
    - Диаграмма кооперации (Communication Diagram)
    - Диаграмма временных соотношений (Timing Diagram)
    - Диаграмма взаимодействия (Interaction Overview Diagram)

1. **Какие отношения между классами могут быть на диаграмме классов?**
   * **Ассоциация (Association):** Простое отношение между классами, показывающее, что один класс связан с другим.
   * **Агрегация (Aggregation):** Специальный вид ассоциации, указывающий на отношение "часть-целое".
   * **Композиция (Composition):** Более сильная форма агрегации, указывающая на жизненный цикл частей, зависящий от целого.
   * **Наследование (Generalization):** Показ отношений между базовым и производным классами.
   * **Реализация (Realization):** Показ того, что класс реализует интерфейс.
2. **Как обозначаются абстрактные классы на диаграмме классов?**

Абстрактные классы обозначаются курсивом (например, **АбстрактныйКласс**) или с использованием стереотипа **<<abstract>>**.

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

Интерфейсы обозначаются с использованием стереотипа **<<interface>>** или специальным значком с полукругом (обычно кружок, соединенный линией с реализацией).

1. **Как отображается доступность членов класса на диаграмме классов?**

Доступность членов класса (атрибутов и методов) отображается следующими символами перед именем члена:

* + **+** : Публичный (public)
  + **#** : Защищённый (protected)
  + **-** : Приватный (private)
  + **~** : Пакетный (package, default)

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

Агрегация — это специальный вид ассоциации, который указывает на отношение "часть-целое" между объектами. Она обозначается пустым ромбом на стороне целого (например, **КлассЦелого ⬨-- КлассЧасти**).

1. **Что такое ассоциация?**

Ассоциация — это отношение между классами, показывающее, что объекты одного класса связаны с объектами другого класса. Она может быть двунаправленной или однонаправленной и часто сопровождается указанием кратности (например, один ко многим).

1. **Какие обозначения используют на диаграмме последовательности?**

На диаграмме последовательности используют следующие обозначения:

* + **Акторы и объекты:** Квадратные или прямоугольные блоки с именами.
  + **Линии жизни (Lifelines):** Вертикальные пунктирные линии, исходящие от акторов или объектов.
  + **Сообщения:** Горизонтальные стрелки между линиями жизни, обозначающие взаимодействия (синхронные и асинхронные).
  + **Активности:** Прямоугольные блоки на линии жизни, обозначающие период выполнения.
  + **Фреймы:** Ограниченные области, обозначающие условные конструкции (например, альтернативы, циклы).

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

Диаграмма последовательности используется для моделирования взаимодействий между объектами в определенный момент времени. Она показывает, какие сообщения передаются между объектами и в каком порядке, что помогает понять динамику системы и определить временную последовательность операций.

1. **Каково назначение диаграммы использования, пакетов и активности?**
   * **Диаграмма использования (Use Case Diagram):** Моделирует функциональные требования системы, показывая, какие функции доступны для различных акторов (пользователей или других систем). Она помогает определить взаимодействие пользователей с системой.
   * **Диаграмма пакетов (Package Diagram):** Показывает организацию и структуру различных компонентов системы в виде пакетов. Она помогает организовать сложные системы и управлять зависимостями между компонентами.
   * **Диаграмма активности (Activity Diagram):** Моделирует бизнес-процессы или логические потоки управления внутри системы. Она помогает визуализировать поток управления и деятельности, включая условные переходы и параллельные процессы.