**1. Что такое UML?**

UML (Unified Modeling Language, унифицированный язык моделирования) — это стандартный язык визуализации, спецификации, разработки и документирования программных систем. UML используется для описания архитектуры системы, процессов и взаимодействий между компонентами, а также для проектирования и анализа программного обеспечения.

**2. Перечислите типы диаграмм UML.**

UML включает 14 типов диаграмм, которые делятся на две группы:

**Структурные диаграммы:**

1. Диаграмма классов (Class Diagram)
2. Диаграмма объектов (Object Diagram)
3. Диаграмма компонентов (Component Diagram)
4. Диаграмма развертывания (Deployment Diagram)
5. Диаграмма пакетов (Package Diagram)
6. Диаграмма композитных структур (Composite Structure Diagram)
7. Диаграмма профилей (Profile Diagram)

**Поведенческие диаграммы:**

1. Диаграмма вариантов использования (Use Case Diagram)
2. Диаграмма активности (Activity Diagram)
3. Диаграмма последовательности (Sequence Diagram)
4. Диаграмма коммуникации (Communication Diagram)
5. Диаграмма взаимодействий (Interaction Overview Diagram)
6. Диаграмма состояний (State Machine Diagram)
7. Временная диаграмма (Timing Diagram)

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

На диаграмме классов возможны следующие виды отношений:

1. **Ассоциация (Association)** — связь между объектами, показывающая их взаимодействие.
2. **Агрегация (Aggregation)** — слабая форма "целое-часть".
3. **Композиция (Composition)** — сильная форма "целое-часть".
4. **Наследование (Generalization)** — отношение "является" между классами.
5. **Реализация (Realization)** — связь между классом и интерфейсом.
6. **Зависимость (Dependency)** — один элемент использует другой.
7. **Обобщение/Специализация** — общее отношение между суперклассами и подклассами.

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

Абстрактные классы обозначаются:

1. **Курсивным шрифтом** для имени класса и его методов.
2. **Словом <<abstract>>** над именем класса (опционально).

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

Интерфейсы обозначаются:

1. **Словом <<interface>>** над именем класса.
2. В виде прямоугольника (аналогично классу).
3. Альтернативное обозначение — кружок (в случае упрощённой записи).

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

Доступность членов класса указывается перед именем атрибута или метода:

1. **+** — публичный (public).
2. **-** — приватный (private).
3. **#** — защищённый (protected).
4. **~** — пакетный (package).

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

**Агрегация** — это отношение "целое-часть", где часть может существовать отдельно от целого (например, команда и игроки). Обозначается линией с пустым ромбом на стороне целого.

**8. Что такое ассоциация?**

**Ассоциация** — это отношение между объектами, показывающее их взаимодействие. Например, студент посещает курс. Ассоциация может быть:

* **Односторонняя или двусторонняя**.
* **С указанием кратности** (например, 1..\*). Обозначается линией между связанными классами.

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

На диаграмме последовательности используются:

1. **Прямоугольники** — объекты или участники (lifelines).
2. **Прерывающаяся вертикальная линия** — "жизненная линия" объекта.
3. **Сообщения**:
   * **Сплошные стрелки** — синхронные вызовы.
   * **Пунктирные стрелки** — асинхронные вызовы или возвраты.
4. **Прямоугольники на жизненной линии** — активность объекта.
5. **Условные блоки** (alt, opt, loop).

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

Диаграмма последовательности используется для:

* Моделирования порядка взаимодействия объектов в системе.
* Демонстрации вызова методов и обмена сообщениями между объектами.
* Анализа временной логики системы.

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

1. **Диаграмма использования (Use Case Diagram):**
   * Описывает функциональные требования системы через взаимодействие пользователей (актеров) с системой.
   * Используется для анализа требований.
2. **Диаграмма пакетов (Package Diagram):**
   * Показывает организацию и зависимость между модулями системы.
   * Полезна для разбиения системы на логические части.
3. **Диаграмма активности (Activity Diagram):**
   * Моделирует рабочие процессы и логику выполнения действий в системе.
   * Используется для анализа бизнес-процессов и потоков управления.