1.Объясните понятие «визуальное моделирование». Перечислите основные цели визуального моделирования.

Использование стандартных графических средств (для облегчения общения); Применяется графическое представление (легче воспринимать информацию); Создание модели, для возможности спланировать и структурировать архитектуру проекта.

Основной целью визуального моделирования является общение между всеми участниками проекта.

Создавая визуальную модель системы, мы можем показать её работу на различных уровнях – можно моделировать взаимодействие между пользователем и системой, взаимодействие объектов внутри системы, а также взаимодействие между различными системами.

2. Перечислите основные достоинства единой системы обозначений.

Понимание обозначений по всему миру любыми людьми.

Удобство работы и интеграции т.к. это общепринятый стандарт.

Широкий выбор инструментов из-за унификации.

Важный вопрос визуального моделирования – выбор графической нотации для описания различных аспектов системы. Нотация должна быть понятна всем заинтересованным сторонам, иначе модель будет бесполезна.

3.(необязательно) Дайте краткую характеристику нотаций ориентированных на объектно-ориентированную методологию.

Среди нотаций, которые используются объектно-ориентированной методологией, поддержку получили метод Буча, технология объектного моделирования (OMT, Object Modeling Technology) и унифицированный язык моделирования (UML, Unified Modeling Language). Однако большинством производителей и такими комитетами по стандартам, как ANSI и Object Management Group (OMG), был принят стандарт UML.

**Метод Буча** назван по имени его изобретателя Гради Буча. Объекты в его модели представляются в виде облаков, иллюстрируя то, что они могут быть почти чем угодно. Нотация Буча предусматривает также несколько видов стрелок для отображения разных типов отношений между объектами.

**Нотация OMT** была разработана Джеймсом Рамбо. OMT использует более простую графику моделирования систем по сравнению с методом Буча.

**Унифицированный язык моделирования (UML)** является результатом совместных усилий Гради Буча, Джеймса Рамбо, Ивара Якобсона (Ivar Jacobson), Ребекки Вирс-Брок (Rebecca Wirfs-Brock), Питера Йордона (Peter Yourdon) и многих других. Символы UML сильно напоминают нотации Буча и OMT, но содержат также элементы из других нотаций. Якобсон первым описал процесс выявления и фиксации требований к системе в виде совокупностей транзакций, называемых вариантами использования (use case). Якобсон также разработал метод проектирования систем под названием «Объектно-ориентированное проектирование программного обеспечения» (Object Oriented Software Engineering, OOSE), концентрирующий внимание на анализе.

UML позволяет создавать несколько типов визуальных диаграмм, которые иллюстрируют различные аспекты системы:

1) Диаграммы Вариантов Использования;

2) Диаграммы Последовательности;

3) Кооперативные диаграммы;

4) Диаграммы Классов;

5) Диаграммы Состояний;

6) Диаграммы Компонентов;

7) Диаграммы Размещения.

4. UML - общая характеристика, назначение основных диаграмм.

**UML (Unified Modeling Language)** - результат совместных усилий множества специалистов. Хотя изначально UML предназначался для объектно-ориентированной проектной документации, он был расширен до большего набора проектной документации и был признан полезным во многих контекстах.

1) Диаграммы Вариантов Использования; позволяет наглядно представлять требования к системе.

2) Диаграммы Последовательности; показывают логическую последовательность событий в сценарии; показывает взаимодействие между действующими лицами и объектами во времени

3) Кооперативные диаграммы; связь со временем отсутствует, она больше внимания заостряет на связях между объектами; полезны в тех случаях, когда нужно оценить последствия сделанных изменений.

4) Диаграммы Классов; позволяют показать основные характеристики классов системы и связи между классами

5) Диаграммы Состояний; содержит информацию о состояниях, в которых может находиться объект, о том, как он переходит из одного состояния в другое и каким образом он ведет себя в этих состояниях; На диаграмме Состояний отображают жизненный цикл одного объекта, начиная с момента его создания и заканчивая разрушением.

6) Диаграммы Компонентов; содержащие дополнительную информацию об элементах диаграммы; они помогают разработчикам и другим участникам проекта лучше понять модель.

7) Диаграммы Размещения. Диаграммы содержат информацию о ключевых абстракциях системы, их связях и поведении. В установившемся состоянии проекта все диаграммы должны быть согласованы между собой и со всей моделью

5. Назначение диаграммы Вариантов Использования и краткое описание ее элементов (типы действующих лиц и связей).

Диаграмма Вариантов Использования позволяет наглядно представлять требования к системе. Этот тип диаграмм описывает общую функциональность системы. Все участники проекта, изучая диаграммы Вариантов Использования, могут понять, что система должна делать. Диаграмма Вариантов Использования содержит варианты использования системы, действующих лиц и связи между ними.

Вариант использования (use case) - это функции, выполняемые системой.

Действующее лицо (actor) - это все, что взаимодействует системой (люди или системы, получающие или передающие информацию в данную систему).

Связи позволяют показать способы взаимодействия между элементами системы.

Типы действующих лиц:

1.физические личности - к этому типу относятся клиенты и обслуживающий персонал

2.другие системы - Допустим, что у банка имеется кредитная система, используемая для работы с информацией о кредитных счетах клиентов

3.время - Время становится действующим лицом, если от него зависит запуск какихлибо событий в системе.

Абстрактным называется действующее лицо, не имеющее экземпляров.

Типы связей:

1. коммуникации (communication) - это связь между вариантом использования и действующим лицом. Связь коммуникации изображают в виде стрелки. Направление стрелки показывает, кто инициирует коммуникацию.
2. использования (uses) - позволяет одному варианту использования задействовать функциональность другого. С помощью таких связей обычно моделируют многократно применяемую функциональность, встречающуюся в двух или более вариантах использования. Связь использования изображается с помощью стрелок и слова «uses».
3. расширения (extends) - в отличие от использования позволяют варианту использования применять функциональные возможности, предоставляемые другим вариантом использования, только при необходимости. В связях расширения общая функциональность также выделяется в отдельный вариант использования. Связи расширения изображают в виде стрелки со словом «extends».
4. обобщения действующего лица (actor generalization) - показывают, общие черты нескольких действующих лиц. Это отношение моделируется с помощью нотации. Связи обобщения создаются не всегда. В общем случае они нужны, если поведение действующего лица одного типа отличается от поведения другого настолько, что это влияет на систему

6.(8) Назначение потока событий и краткая характеристика его составных частей (краткое описание, предусловие, постусловие, основной поток, альтернативный поток и поток ошибок).

Варианты использования начинают описывать, что должна будет делать система. Но чтобы фактически разработать систему, потребуются более конкретные детали. Они определяются в документе, называемом потоком. Диаграмма вариантов использования системы обработки заказов событий (flow of events). Целью потока событий является документирование процесса обработки данных, реализуемого в рамках варианта использования. Этот документ подробно описывает, что будут делать пользователи системы и сама система.

Обычно поток событий содержит:

1) краткое описание; Каждый вариант использования должен иметь связанное с ним краткое описание того, что он будет делать.

2) предусловия (pre-conditions); - это такие условия, которые должны быть выполнены, прежде чем вариант использования начнет свою работу. Не у всех вариантов использования бывают предварительные условия.

3) основной поток событий; 4) альтернативный поток событий; Поток событий поэтапно описывает, что должно происходить во время выполнения заложенной в вариант использования функциональности. Поток событий уделяет внимание тому, что (а не как) будет делать система, причем описывает это с точки зрения пользователя. Первичный и альтернативный потоки событий содержат:

1) описание того, каким образом запускается вариант использования;

2) различные пути выполнения варианта использования;

3) нормальный, или основной, поток событий варианта использования;

4) отклонения от основного потока событий (так называемые альтернативные потоки);

5) потоки ошибок;

6) описание того, каким образом завершается вариант использования.

5) постусловия (post-conditions) - называются такие условия, которые должны быть выполнены после завершения варианта использования.