Контрольные вопросы:

1. В чем отличие предварительного проектирования от детального?

В процессе предварительного внешнего проектирования структурируются цели проекта, определяется взаимодействие с пользователем, но не учитываются многие детали. Во время детального внешнего проектирования определяются внешние спецификации каждой функции для пользовательского проекта. Детальное проектирование, в свою очередь, определяется как процесс детализации и расширения предварительного проекта до такой степени, при которой проект полностью готов к реализации.

2. Перечислите и кратко опишите архитектурные системные паттерны.

Репозиторий. Все совместно используемые подсистемами данные хранятся в центральной базе данных, доступной всем подсистемам. Репозиторий является пассивным элементом, а управление им возложено на подсистемы.

Паттерн Клиент/сервер. Данные и процессы системы распределены между несколькими процессорами. Паттерн имеет три основных компонента: набор автономных серверов (предоставляют сервисы другим подсистемам), набор подсистем — клиентов (которые вызывают сервисы, предоставляемые серверами) и сеть (служит для доступа клиентов к сервисам). Клиенты должны знать имена серверов и сервисов, в то время как серверам не надо знать имена клиентов и их количество. Клиенты получают доступ к сервисам, предоставляемым серверами посредством удаленного вызова процедур.

Паттерн обьектно-ориентированный. Разделение приложения или подсистемы на три функциональные части (модель данных, представление (пользовательский интерфейс) и контроллер (управляющую логику)).

Сессия (Session), Рабочая нить (Worker Thread). В системах распределённой обработки обеспечивает возможность серверам различать клиентов, что позволяет повысить эффективность многопоточных приложений.

Паттерн Многоуровневая система (Layers) или Абстрактная машина. В соответствии с паттерном Многоуровневая система структурные элементы системы организуются в отдельные уровни со взаимосвязанными обязанностями таким образом, чтобы на нижнем.

3. Перечислите и кратко опишите паттерны управления.

Вызов—возврат. Вызов программных процедур осуществляется сверху вниз, т. е. управление начинается на вершине иерархии процедур и через вызовы передается на нижние уровни иерархии.

Диспетчер. Один системный компонент назначается диспетчером и управляет запуском и завершением других процессов системы и координирует эти процессы.

Передача сообщений. В рамках данного паттерна событие представляет собой передачу сообщения всем подсистемам. Любая подсистема, которая обрабатывает данное событие, отвечает на него.

4. Что такое «связность модуля»? Перечислите и кратко опишите типы связности модуля с указанием значения силы связности для каждого типа.

Связность модуля – внутренняя характеристика модуля, характеризующая меру прочности соединения функциональных и информационных объектов внутри одного модуля. Связность модуля характеризует степень его «плотности», степень зависимости его частей и направленности на решение определенной задачи.

5. Что такое «сцепление модулей»? Перечислите и кратко опишите типы сцепления модулей.

Сцепление модулей — это мера относительной независимости модулей. Модули являются полностью независимыми, если каждый из них не содержит о другом никакой информации.

6. Опишите следующие фундаментальные паттерны проектирования: делегирование, неизменяемый объект, интерфейс.

Делегирование — основной шаблон проектирования, в котором объект внешне выражает некоторое поведение, но в реальности передаёт ответственность за выполнение этого поведения связанному объекту. Шаблон делегирования является фундаментальной абстракцией, на основе которой реализованы другие шаблоны - композиция примеси и аспекты .

Неизменяемый объект - в объектно-ориентированном программировании объект, который не может быть изменён после своего создания.

Интерфейс — основной шаблон проектирования, являющийся общим методом для структурирования компьютерных программ для того, чтобы их было проще понять. В общем, интерфейс — это класс, который обеспечивает программисту простой или более программно-специфический способ доступа к другим классам.