

## Контрольные вопросы:

1) Предварительное проектирование формирует абстракции архитектурного уровня, детальное проектирование уточняет эти абстракции, добавляет подробности алгоритмического уровня. Результат предварительного проектирования становится основой для следующего шага - детального проектирования.

2)

- Репозиторий
- Клиент/сервер
- Объектно – ориентированный
- Многоуровневая система
- Потоки данных

3) Паттерны централизованного управления:

Вызов – возврат

Диспетчер

Паттерны управления, основанные на событиях:

Передача сообщений

Управляемый прерываниями

4) Связность модуля – внутренняя характеристика модуля, характеризующая меру прочности соединения функциональных и информационных объектов внутри одного модуля.

Существует 7 типов связности:

**Функционально связный модуль** содержит объекты, предназначенные для решения одной единственной задачи.

**В последовательно связанном модуле** его объекты охватывают подзадачи, для которых выходные данные одной из подзадач являются входными для другой.

**Информационно связный модуль** содержит объекты, использующие одни и те же входные или выходные данные.

**В процедурно связанном модуле** объекты включены в различные подзадачи, в которых управление переходит от одной подзадачи к следующей.

**Модуль с временной связностью** – модуль, в котором объекты модуля привязаны к конкретному промежутку времени. Элементы данного модуля почти не связаны друг с другом за исключением того, что должны выполняться в определенное время.

**Модуль с логической связностью** – модуль, объекты которого содействуют решению одной общей подзадачи, для которой эти объекты отобраны во внешнем по отношению к модулю мире.

**Модуль со связностью по совпадению** содержит объекты, которые слабо связаны друг с другом.

Вид связности	Сила связности
По совпадению	0 (слабая связность)
Логическая	1
Временная	3
Процедурная	5
Коммуникативная	7
Последовательная	9
Функциональная	10 (сильная связность)

5) Сцепление модулей — это мера относительной независимости модулей. Эта характеристика является до некоторой степени обратной связности модулей.

Модули имеют **сцепление по кодам**, если коды их команд перемежаются друг с другом.

Модуль **сцеплен по внешним ссылкам**, если у него есть доступ к данным в другом модуле через внешнюю точку входа. Таким путем осуществляется неявное управление функционированием другого модуля.

Модули имеют **сцепление по управлению**, если какой-либо из них управляет решениями внутри другого с помощью передачи флагов, переключателей и т.д., предназначенных для выполнения функций управления, т.е. один из модулей знает о функциях другого.

Модули **сцеплены по общей области**, если они разделяют одну и ту же глобальную структуру данных.

Модули **сцеплены по образцу**, если параметры содержат структуры данных.

Модули **сцеплены по данным**, если они имеют общие единицы, которые передаются от одного к другому как параметры, представляющие собой простые элементы данных, т.е. вызывающий модуль "знает" только имя вызываемого модуля, а также типы и значения некоторых его переменных

**Независимое сцепление** возможно в том случае, если модули не вызывают друг друга или не обрабатывают одну и ту же информацию.

Вид сцепления	Степень сцепления модуля
По кодам	9 (сильное сцепление)
По внешним ссылкам	7
По управлению	5
По общей области	4
По образцу	3
По данным	1
Независимое	0 (слабое сцепление)

б) **Шаблон делегирования** (Delegation pattern) — Объект внешне выражает некоторое поведение, но в реальности передаёт ответственность за выполнение этого поведения связанному объекту.

**Неизменяемый интерфейс** (Immutable interface) — Создание неизменяемого объекта.

**Интерфейс** (Interface) — Общий метод для структурирования компьютерных программ для того, чтобы их было проще понять.

**Model-View-Controller** (MVC) — Модель-представление-контроллер:

**Model.** Первая компонента/модуль — так называемая модель. Она содержит всю бизнес-логику приложения.

**View.** Вторая часть системы — вид. Данный модуль отвечает за отображение данных пользователю. Все, что видит пользователь, генерируется видом.

**Controller.** Третьим звеном данной цепи является контроллер. В нем хранится код, который отвечает за обработку действий пользователя (любое действие пользователя в системе обрабатывается в контроллере).