**Технология разработки программного обеспечения**

**Контрольные вопросы 3.**

**Современные принципы разработки программных приложений.**

1. В чем состоит метод нисходящего проектирования?

Метод нисходящего проектирования (метод пошаговой детализации, метод иерархического проектирования, top-down подход) - определение спецификаций компонентов системы путем последовательного выделения в ее составе отдельных составляющих и их постепенной детализации до уровня, обеспечивающего однозначное понимание того, что и как необходимо разрабатывать.

Метод эффективен при разработке сложных по характеру и больших по объему программ, когда к их разработке необходимо привлекать большое число программистов, работающих параллельно.

1. Что такое «программы-заглушки»?

Для ускорения разработки программного комплекса часто вместо некоторых программ нижнего уровня, находящихся в процессе разработки, могут применяться специальные "программы-заглушки" Программы-заглушки требуются только на ранних стадиях разработки для того, чтобы не сдерживать общий ход создания программного комплекса. Суть программы-заглушки заключается в том, что при обращении к ней в соответствии с заданным набором исходных тестовых данных она не формирует, а выбирает результат "решения" из заранее подготовленного набора.

1. В чем суть модульного проектирования?

Модульное проектирование. Модуль – это последовательность логически взаимосвязанных фрагментов задачи, оформленных как отдельная часть программы. При этом программные модули должны обладать следующими свойствами:

на модуль можно ссылаться (т.е. обращаться к нему) по имени, в том числе и из других модулей;

по завершении работы модуль должен возвращать управление тому модулю, который его вызывал;

модуль должен иметь один вход и выход;

модуль должен иметь небольшой размер, обеспечивающий его обозримость.

1. Перечислите преимущества модульного проектирования.

большую программу могут разрабатывать одновременно несколько исполнителей, и это позволяет сократить сроки ее разработки;

появляется возможность создавать и многократно использовать в дальнейшем библиотеки наиболее употребимых программ;

упрощается процедура загрузки больших программ в оперативную память, когда требуется ее сегментация;

возникает много естественных контрольных точек для наблюдения за осуществлением хода разработки программ, а в последующем для контроля за ходом исполнения программ;

обеспечивается более эффективное тестирование программ, проще осуществляются проектирование и последующая отладка.

1. Назовите цель структурного программирования.

Одной из целей структурного программирования было стремление облегчить разработку и отладку программных модулей, а главное - их последующее сопровождение и модификацию. Эффективность применения современных универсальных языков программирования во многом определяется удобством написания с их помощью структурных программ.

1. Опишите метод SADT.

Метод SADT (Structured Analysis and Design Technique) представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области.

Метод SADT разработан Дугласом Россом (SoftTech, Inc.) в 1969 г. для моделирования искусственных систем средней сложности. Данный метод успешно использовался в военных, промышленных и коммерческих организациях США для решения широкого круга задач, таких, как долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, разработка ПО для оборонных систем, управление финансами и материально-техническим снабжением и др.

1. Что такое диаграммы потоков данных.

Диаграммы потоков данных (Data Flow Diagrams – DFD) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

1. Опишите ER-модель.

Наиболее распространенным средством моделирования данных (предметной области) является модель "сущность-связь" (Entity-Relationship Model - ERМ).

Впервые введена Питером Ченом в 1976 г.

Предметная область в ER-модели представляется в виде набора сущностей (с атрибутами) и связей между ними.

1. В чем состоит сущность объектно-ориентированного подхода.

Объектно-ориентированный подход использует объектную декомпозицию, то есть поведение системы описывается в терминах взаимодействия объектов.

Класс - это абстракция множества сущностей реального мира, объединенных общностью структуры и поведения.

Объект - это элемент класса, то есть абстракция определенной сущности.

Объекты активны, у них есть не только внутренняя структура, но и поведение, которое описывается так называемыми методами объекта.

1. В чем состоят основные принципы объектно-ориентированного подхода?

Основным инструментом борьбы со сложностью в объектно-ориентированном подходе является инкапсуляция - сокрытие реализации объектов (их внутренней структуры и деталей реализации методов ) с предоставлением во вне только строго определенных интерфейсов.

Понятие "полиморфизм" может трактоваться как способность объекта принадлежать более чем одному классу. Введение этого понятия отражает необходимость смотреть на объекты под разными углами зрения, выделять при построении абстракций разные аспекты сущностей моделируемой предметной области, не нарушая при этом целостности объекта.

Наследование означает построение новых классов на основе существующих с возможностью добавления или переопределения данных и методов. Наследование является важным инструментом борьбы с размножением сущностей без необходимости. Общая информация не дублируется, указывается только то, что меняется. При этом класс -потомок помнит о своих "корнях".

1. Что такое CASE-технология?

CASE-технология представляет собой совокупность средств системного анализа, проектирования, разработки и сопровождения сложных программных систем, поддерживаемых комплексом взаимосвязанных инструментальных средств автоматизации всех этапов разработки программ.

Благодаря структурным методам CASE-технология на стадиях анализа и проектирования обеспечивает разработчиков широкими возможностями для различного рода моделирования, а централизованное хранение всей необходимой для проектирования информации и контроль за целостностью данных гарантируют согласованность взаимодействия всех специалистов, занятых в разработке ПО.

1. В чем состоит основное достоинство CASE-технологии?

CASE-технологии - относительно новое направление, формировавшееся на рубеже 80-х годов.

CASE-технологии делятся на две группы:

встроенные в систему реализации, в которых все решения по проектированию и реализации привязаны к выбранной системе явления базами данных (СУБД);

независимые от системы реализации, в которых все решения по проектированию ориентированы на унификацию начальных этапов жизненного цикла, средств их документирования и обеспечивают большую гибкость в выборе средств реализации.

Основное достоинство CASE-технологии - поддержка коллективной работы над проектом за счет возможности работы в локальной сети разработчиков, экспорта/импорта любых фрагментов проекта, организационного управления проектом.

1. В чем сущность технологии RAD?

В начале 80-х годов появилась методология, по которой разработка программы начиналась не после завершения процесса выработки окончательных требований к ней, а как только устанавливались требования на первый, “стартовый” (пилотный) вариант прикладной программы, позволяющий начать содержательную работу по ее реализации на компьютере.

Это дало пользователю возможность, получая уже с первых шагов конкретное представление о характере реализации задачи, уточнять ее постановку. Тем самым облегчался процесс экспериментального поиска нужного решения автоматизации задачи.

1. Что отражают основные типы данных DataWarehouse?

Метаданные (данные о данных) облегчают возможность визуального представления содержимого Data Warehouse, позволяют, "перемещаясь" по хранилищу, быстро отбирать необходимые данные для последующей обработки.

Основные типы метаданных Data Warehouse отражают:

структуру и содержимое хранилища;

соответствие между исходными и выходными данными;

объемные характеристики данных;

критерии архивирования;

отношения между данными;

информацию по кодированию;

интервал жизни данных и т.п.

1. Опишите структуру OLAP-систем.

Система OLAP (On-Line Analytical Process) предоставляет возможность разработки информационных систем, ориентированных на организацию многомерных баз данных и создание корпоративных сетей, а также обеспечивает поддержку Web-технологий в сетях Internet/Intranet.

Успешное применение инструментальных средств OLAP-систем объясняется быстротой разработки приложений, гибкостью и широкими возможностями в области доступа к данным и их преобразования. В настоящее время на рынке ПО предлагается большое число OLAP-стем, разработчиками которых являются различные фирмы, например IBM, Informix, Microsoft, Oracle, Sybase и др.

1. Опишите способы хранения данных в системе OLAP.

Локально. Данные размещаются на компьютерах пользователей. Обработка, анализ и управление данными выполняется на локальных рабочих местах. Такая структура OLAP системы имеет существенные недостатки, связанные со скоростью обработки данных, защищенностью данных и ограниченным применением многомерного анализа.

Реляционные базы данных. Эти базы данных используются при совместной работе OLAP системы с CRM системой или ERP системой. Данные хранятся на сервере этих систем в виде реляционных баз данных или хранилищ данных. OLAP сервер обращается к этим базам данных для формирования необходимых многомерных структур и проведения анализа.

Многомерные базы данных. В этом случае данные организованы в виде специального хранилища данных на выделенном сервере. Все операции с данными осуществляются на этом сервере, который преобразует исходные данные в многомерные структуры. Такие структуры называют OLAP кубом. Источниками данных для формирования OLAP куба являются реляционные базы данных и/или клиентские файлы. Сервер данных осуществляет предварительную подготовку и обработку данных. OLAP сервер работает с OLAP кубом не имея непосредственного доступа к источникам данных (реляционным базам данных, клиентским файлам и др.).

1. Какие подходы к программированию интерфейса помогли преодолеть кризис программирования?

Выход из этой ситуации обозначился благодаря двум подходам.

Первый - стандартизация многих функций интерфейса, благодаря чему появилась возможность использовать библиотеки, имеющиеся, например, в Windows. В итоге при смене стиля графического интерфейса (например, при переходе от Windows 3.x к Windows 95) приложения смогли автоматически приспосабливаться к новой системе без какого-либо перепрограммирования.

Вторым революционным шагом явилось появление визуального программирования, возникшего в Visual Basic и нашедшего блестящее воплощение в Delphi и С++Builder фирмы Borland.

1. Перечислите наиболее популярные среды визуального проектирования программ.

Basic: Microsoft Visual Basic;

Pascal: Borland Delphi;

C++: Borland C++Bulider;

Java: Symantec Cafe.

1. Перечислите виды архитектуры программных систем.

Автономные приложения, Приложения в файл-серверной архитектуре, Приложения в клиент-серверной архитектуре, Приложения в многозвенной архитектуре, Приложения в распределенной архитектуре.

1. Каковы современные тенденции в программной инженерии?

индивидуумы и взаимодействия между ними ценятся выше процессов и инструментов;

работающее ПО ценится выше всеобъемлющей документации;

сотрудничество с заказчиками ценится выше формальных договоров;

реагирование на изменения ценится выше строгого следования плану.

**Источники**: конспект, презентация.