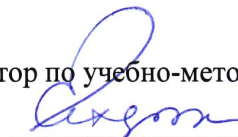


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ИНКЛЮЗИВНОГО ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО ЭКОНОМИЧЕСКИЙ
УНИВЕРСИТЕТ»

КАФЕДРА ЦИФРОВЫХ ТЕХНОЛОГИЙ

УТВЕРЖДАЮ

Проректор по учебно-методической работе

 Е.С. Сахарчук

«27» 01 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Средства отладки клиент-серверных приложений

образовательная программа направления подготовки 09.03.01 «Информатика и
вычислительная техника»
шифр, наименование

Направленность (профиль)

Программное обеспечение вычислительной техники и информационных систем

Квалификация (степень) выпускника: бакалавр

Форма обучения очная


Курс 2 семестр 4

Москва 2022

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования направления (специальности) 09.03.01 «Информатика и вычислительная техника», утвержденного приказом Министерства образования и науки Российской Федерации № 929 от «19» сентября 2017 г. Зарегистрировано в Минюсте России «10» октября 2017 г. № 48489

Разработчики рабочей программы:

МГГЭУ, доцент кафедры цифровых технологий
место работы, занимаемая должность

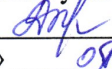
 Перепелкина Ю.В. 14.03 2022 г.
подпись Ф.И.О. Дата

Рабочая программа утверждена на заседании кафедры цифровых технологий
(протокол № 4 от «21» 03 2022 г.)

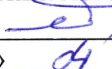
на заседании Учебно-методического совета МГГЭУ
(протокол № 1 от «27» 01 2022 г.)

СОГЛАСОВАНО:


Начальник учебно-методического управления

 И.Г. Дмитриева
«27» 09 2022 г.

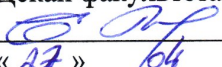
Начальник методического отдела

 Д.Е. Гапеенок
«27» 04 2022 г.

Заведующий библиотекой

 В.А. Ахтырская
«27» 09 2022 г.

Декан факультета ПМий

 Е.В.Петрунина
«27» 04 2022 г.

Содержание

- 1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ**
- 2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**
- 3. ОСОБЕННОСТИ ОБУЧЕНИЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ**
- 4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**
- 5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**
- 6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**
- 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)**
- 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)**

1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

1.1. Цели и задачи освоения учебной дисциплины (модуля)

Цель освоения дисциплины: приобретение слушателями прочных знаний и практических навыков в области, определяемой задачами курса.

Задачи:

- дать представление закономерностей развития клиент — серверных баз данных, свойств информации и особенностей информационных процессов, используя технологию клиент — сервер;
- ознакомить с основами современных теорий постреляционных баз данных, особенностями использования клиент —серверных технологий;
- обучить анализу и моделированию предметной области в среде клиент - сервер, связанных с широкомасштабным использованием информационно-коммуникационных технологий в различных сферах деятельности.

1.2. Место дисциплины (модуля) в структуре образовательной программы направления подготовки 09.03.03 «Прикладная информатика» (бакалавриат).

Учебная дисциплина «Средства отладки разработки клиент-серверных приложений» (Б1.В.ДВ.03.02) относится к блоку 1 (вариативная часть). Предшествующими дисциплинами, формирующими начальные знания, являются: Методы технологии разработки клиент-серверных приложений, Информационные и электронные ресурсы в организации научных исследований, Методы решения обратных задач. Последующими дисциплинами являются: Методы и технологии разработки клиент-серверных приложений, Подготовка к сдаче и сдача государственного экзамена, Практика по получению профессиональных умений и опыта профессиональной деятельности (научная практика).

1.3. Требования к результатам освоения учебной дисциплины (модуля)

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций:

Код компетенции	Содержание компетенции	Индикаторы достижения компетенции
-----------------	------------------------	-----------------------------------

ПК-4	Способен разрабатывать, внедрять и адаптировать прикладное программное	<p>ПК-4.1. Знает: методы разработки, анализа и проектирования ПО; функциональное и техническое проектирование; паттерны проектирования; номенклатуру инструментальных средств, поддерживающих создание программного обеспечения; техники распределения задач на разработку между исполнителями</p> <p>ПК-4.2. Умеет: работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем; использовать нотации для построения функциональной и процессной моделей исследуемой предметной области; проектировать компоненты программных средств</p> <p>ПК-4.3. Владеет: навыками определения содержания работ по созданию программного продукта; приемами работы с инструментальными средствами автоматизации, проектирования и реализации программного продукта; навыками проектирования прикладных программных продуктов, в том числе клиент-серверных приложений</p>
------	--	---

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

2.1. Объем учебной дисциплины (модуля).

Объем дисциплины «Цифровая обработка сигналов» составляет 3 зачетных единиц/108

Вид учебной работы	Всего, часов	Очная форма
		Курс, часов
	Очная форма	1 курс
Аудиторная работа обучающихся с преподавателем (по видам учебных занятий), всего в том числе:	108	108
Лекции	14	14
В том числе, практическая подготовка (ЛПП)		
Практические занятия	34	34
В том числе, практическая подготовка (ПЗПП)		
Лабораторные занятия		
В том числе, практическая подготовка (ЛРПП)		
Самостоятельная работа обучающихся	60	60
В том числе, практическая подготовка (СРПП)		
Промежуточная аттестация (подготовка и сдача), всего:		
Контрольная работа		
Курсовая работа		
Зачет	4	4
Экзамен		108
Итого: Общая трудоемкость учебной дисциплины (в часах, зачетных единицах)	108 часов (3 з.е.)	108 часов (3 з.е.)

часов:

2.2. Содержание разделов учебной дисциплины (модуля)

№ п/п	Наименование раздела (темы)	Содержание раздела (тематика занятий)	Формируемые компетенции (индекс)
1	Компьютерные сети. Основные определения	Структура сети. Протоколы . Адреса и имена.4 Основные протоколы транспортного уровня UDP и TCP1. Основные службы TCP/IP1.6 Порт	ПК-4
2	Программное обеспечение компьютерных сетей	Модель клиент/сервер Операционные системы. Серверное программное обеспечение Клиентское программное обеспечение	ПК-4
3	Программные интерфейсы	Сокеты. Основные понятия. Основные функции. API сокетов. Простейшая реализация модели клиент/сервер на основе сокетов. Описание API winsock. Каналы (Pipes). Создание каналов. Создание соединения с помощью именованных каналов. Передача данных по именованному каналу Простейший пример. Удаленный вызов процедур (RPC— remote call procedure) RPC для открытых систем. RPC для Windows. Подготовка сервера к соединению. Обслуживание клиентских вызовов. Соединение клиента с сервером. Создание дескриптора соединения Вызов удаленной процедуры. Нахождение серверной программы. Передача параметров от клиентского приложения серверному. Пример создания сетевого приложения на основе RPC Window. Определение интерфейса. Генерация UUID. IDL файл. Файл конфигурации. Генерация файла заглушки. Клиентское приложение. Серверное приложение. Завершение работы сервера	ПК-4
4	Разработка сетевых	Обзор технологий.	ПК-4

	приложений на основе www-сервера	Программирование CGI-скриптов. Описание интерфейса. Взаимодействие WWW-сервера и CGI-программы. Переменные среды о сервере Программный интерфейс ISAPI. Фильтры IIS	
--	----------------------------------	---	--

2.3. Разделы дисциплины и виды занятий

Очная форма обучения

№ п/п	Наименование раздела (темы)	Аудиторная работа		Внеауд. работа	Объем в часах
		Л	ПЗ/ЛР	СР	Всего
		в том числе, ЛПП	в том числе, ПЗПП/ЛРПП	в том числе, СРПП	в том числе, ПП
4 семестр					
1.	Компьютерные сети. Основные определения	2	8	15	25
2.	Программное обеспечение компьютерных сетей	4	8	15	29
3.	Программные интерфейсы	4	8	15	29
4.	Разработка сетевых приложений на основе www-сервера	4	10	15	29
	Зачет	4			
	<i>Итого:</i>	14	34	60	108
	<i>В том числе ПП:</i>				
	<i>Всего:</i>	14	34	60	108
	<i>В том числе ПП:</i>				

2.4. План самостоятельной работы обучающегося по дисциплине (модулю)

Очная форма обучения

№	Название разделов и тем	Виды самостоятельной работы	Трудоемкость	Формируемые компетенции	Формы контроля
1	Компьютерные сети. Основные определения	Работа с источниками	15	ПК-4	Устный опрос

2	Программное обеспечение компьютерных сетей	Работа с источниками	15	ПК-4	Устный опрос
3	Программные интерфейсы	Работа с источниками	15	ПК-4	Устный опрос
4	Разработка сетевых приложений на основе www-сервера	Работа с источниками	15	ПК-4	Устный опрос

3. ОСОБЕННОСТИ ОБУЧЕНИЯ ИНВАЛИДОВ И ЛИЦ С ОВЗ

Обучение лиц с ограниченными возможностями здоровья осуществляется с учетом индивидуальных психофизических особенностей, а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида.

Для получения обучающимися, имеющими ограниченные физические возможности, качественного образования должны выполняться следующие важные условия: обучающийся должен иметь возможность беспрепятственно посещать образовательное учреждение и использовать в своём обучении дистанционные образовательные технологии.

Для обучения и контроля обучающихся с нарушениями координации движений предусмотрено проведение тестирования с использованием компьютера.

Во время аудиторных занятий обязательно использование средств обеспечения наглядности учебного материала с помощью мультимедийного проектора. Скорость изложения материала должна учитывать ограниченные физические возможности студентов.

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение дисциплины для организации самостоятельной работы студентов (содержит перечень основной литературы, дополнительной литературы, программного обеспечения и Интернет-ресурсы).

В распоряжении преподавателей и обучающихся имеется основное необходимое материально-техническое оборудование, Интернет-ресурсы, доступ к полнотекстовым электронным базам, книжный фонд библиотеки Московского государственного гуманитарно-экономического университета.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Интерактивные образовательные технологии, используемые в аудиторных занятиях самостоятельной работе обучающихся не предусмотрены.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

6.1. Организация входного, текущего и промежуточного контроля обучения

Входное тестирование – не предусмотрено

Текущий контроль – опрос, тестирование.

Промежуточная аттестация – зачет

6.2. Вопросы для устного опроса

1. Дайте понятие процесса, приведите его основные свойства, структуру и описание функции запуска.
2. Дайте понятие потока, опишите содержимое контекста потока, функции создания потока и ее параметров.
3. Опишите предназначение и использование переменных окружения.
4. Как определить дескриптор процесса?
5. Как определить дескриптор первичного потока?
6. Как определить текущий каталог?
7. Дайте определение каналам (pipes).
8. В чем отличие именованных каналов от анонимных?
9. Перечислите основные функции для работы с каналами.
10. Дайте понятие «мапирование файлов».
11. Общая схема использования мапирования файлов для организации обмена данными между процессами.
12. Перечислите основные функции мапирования файлов.
13. Для чего нужна синхронизация потоков?
14. Укажите достоинства и недостатки синхронизации потоков в пользовательском режиме?
15. Для чего нужны функции WaitSingleObject, WaitMultipleObject?
16. Укажите особенности использования объектов «событие» для организации синхронизации потоков.

6.2. Тематика рефератов, проектов, творческих заданий, эссе и т.п.

6.4. Пример теста

1. Установите соответствие

Расположение компонентов на стороне клиента или сервера определяет следующие основные модели их взаимодействия в рамках двухуровневой архитектуры:

- а) сервер терминалов — распределенное представление данных;
- б) файл-сервер — доступ к удаленной базе данных и файловым ресурсам;
- в) сервер БД — удаленное представление данных;
- г) сервер приложений — удаленное приложение.

2. Вставьте пропущенное название архитектуры

модель _____, представлявшего доступ файловым ресурсам, в том числе и к удаленной базе данных. В этом случае выделенный узел сети является файловым сервером, на котором размещены файлы базы данных. На клиентах выполняются приложения, в которых совмещены компонент представления и прикладной компонент (СУБД и прикладная программа), использующие подключенную удаленную базу как

локальный файл. Протоколы обмена при этом представляют набор низкоуровневых вызовов операций файловой системы. Такая модель показала свою неэффективность ввиду того, что при активной работе с таблицами БД возникает большая нагрузка на сеть. Частичным решением является поддержка тиражирования (репликации) таблиц и запросов. В этом случае, например при изменении данных, обновляется не вся таблица, а только модифицированная ее часть.

3. Вставьте пропущенное слово

Архитектура _____ — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемыми серверами, и заказчиками услуг, называемыми клиентами.

4. Выберите верные варианты ответов

Существует два вида представления архитектуры клиент-сервер

- а) двухуровневая
- б) файловая
- в) трёхуровневая
- г) видовая

5. Вставьте пропущенные слова

Модель _____ реализовывалась на универсальной ЭВМ (мэйнфрейме), выступавшей в роли сервера, с подключенными к ней алфавитно-цифровыми терминалами. Пользователи выполняли ввод данных с клавиатуры терминала, которые затем передавались на мэйнфрейм и там выполнялась их обработка, включая формирование «картинки» с результатами. Эта «картинка» и возвращалась пользователю на экран терминала.

6. Выберите верный вариант ответа

На рабочей станции установлены только программные средства, поддерживающие интерфейс с БД. На сервере БД находится БД под управлением СУБД, архитектура сети «клиент-сервер». В архитектуре ИС выделен сервер приложений, на котором находятся программные средства общего пользования. Эти серверы выполняют всю содержательную обработку данных.

- а) Многоуровневый «клиент-сервер»
- б) Двухуровневый «клиент-сервер»
- в) Файловый «клиент-сервер»

7. Какая модель описана выберите верный вариант ответа.

В этом случае ядро СУБД функционирует на сервере, прикладная программа на клиенте, а протокол обмена обеспечивается с помощью языка SQL. Такой подход ведет к уменьшению загрузки сети и унификации интерфейса «клиент-сервер». Однако, сетевой трафик остается достаточно высоким, кроме того, по-прежнему невозможно

удовлетворительное администрирование приложений, поскольку в одной программе совмещаются различные функции.

- а) сервер терминалов;
- б) файл-сервер;
- в) сервер БД;
- г) сервер приложений.

8. Вставьте пропущенные слова

Реализация прикладного компонента на стороне сервера представляет следующую модель — _____ Перенос функций прикладного компонента на сервер снижает требования к конфигурации клиентов и упрощает администрирование, но представляет повышенные требования к производительности, безопасности и надежности сервера.

5. Значение архитектуры клиент-сервер – в ... доступа

- (1) безопасности и надежности...
- (2) эргономике и удобстве ...
- (3) простоте и невысокой стоимости ...

6. Архитектура "клиент-сервер" использует концепцию открытых систем посредством ...

- (1) независимости и стандартизации
- (2) безопасности и надежности...
- (3) эргономики и удобства ...

7. Открытые системы обеспечивают совершенствование приложений посредством ...

- (1) компонентной архитектуры
- (2) стандартизации интерфейсов
- (3) удобства использования

8. К задачам сервера НЕ относится: ...

- (1) запрос доступа
- (2) хранение файлов
- (3) производство вычислений

9. Особенность обработки данных в архитектуре "файл-сервер" – в том, что...

- (1) СУБД расположена на клиенте
- (2) СУБД расположена на сервере
- (3) приложение расположено на сервере

10. Сервер БД – это ...

- 1) СУБД типа "клиент-сервер"
- (2) СУБД типа "файл-сервер"
- (3) локальная СУБД

11. Особенность архитектуры сервера БД – в ...

- (1) общей БД для клиентов
- (2) выделенных клиенте и сервере
- (3) хранении БД на одном компьютере

12. Интерфейс СУБД в архитектуре "клиент-сервер" основан на применении ...

- (1) протокола TCP
- (2) языка SQL
- (3) технологии RPC

13. SQL-сервер – это ...

- (1) сервер БД на основе SQL
- (2) файл-сервер на основе SQL
- (3) телекоммуникационный сервер

14. Основные компоненты трехуровневой архитектуры "клиент-сервер" – это ...

- (1) клиент, сервер, СУБД
- (2) клиент, сервер, БД
- (3) презентационная логика, бизнес-логика, логика доступа

15. Модели взаимодействия "клиент-сервер" – это ...

- (1) "тонкий клиент", "толстый клиент" , "файл-сервер"
- (2) "тонкий клиент", "толстый клиент" , "сервер приложения"
- (3) "тонкий клиент", "толстый клиент" , "сервер БД"

16. "Толстый" клиент – это ...

- (1) прикладная логика и бизнес-логика
- (2) прикладная логика и логика доступа
- (3) бизнес-логика и логика доступа

17. "Тонкий" клиент – это ...

- (1) прикладная логика
- (2) логика доступа
- (3) бизнес-логика

18. Сервер приложения – это ...

- (1) прикладная логика
- (2) логика доступа
- (3) бизнес-логика

19. Особенность обработки данных с "толстым" клиентом –...

- (1) децентрализация бизнес-логики
- (2) централизация бизнес-логики
- (3) использование веб-браузера

20. Особенность обработки данных с "тонким" клиентом –...

- (1) децентрализация бизнес-логики
- (2) централизация бизнес-логики
- (3) использование веб-браузера

21. Особенность обработки данных с сервером приложения - ...

- (1) гибкость бизнес-правил
- (2) децентрализация бизнес-логики
- (3) использование веб-браузера

22. Особенность трехуровневой архитектуры "клиент-сервер" –...

- (1) использование расширений серверной части
- (2) использование расширений клиентской части

(3) использование протокола HTTP и веб-браузера

23. Назначение программ расширения серверной части –...

- (1) обмен данными между браузером и сервером БД
- (2) обмен транзакциями между браузером и сервером БД
- (3) обмен данными между браузером и файл-сервером

24. Основная идея организации локальной вычислительной сети – в ...

- (1) специализации ресурсов
- (2) высокой производительности
- (3) надежности и безопасности

25. Сервер локальной вычислительной сети характеризуется ...

- (1) представлением интерфейса пользователя
- (2) мощными вычислительными ресурсами
- (3) надежным шифрованием данных

26. Основные роли в локальной вычислительной сети – это ...

- (1) клиент и сервер
- (2) клиент и СУБД
- (3) СУБД и сервер

27. К какому уровню модели OSI относится протокол IP

- 1) Канальный
- 2) Сетевой
- 3) Транспортный
- 4) Сеансовый

28. 2 К какому уровню модели OSI относится протокол TCP

- 1) Канальный
- 2) Сетевой
- 3) Транспортный
- 4) Сеансовый

29. 3 К какому уровню модели OSI относится протокол FTP

- 1) Канальный
- 2) Сетевой
- 3) Транспортный
- 4) Сеансовый

30. 4 К какому уровню модели OSI относится протокол Ethernet

- 1) Канальный
- 2) Сетевой
- 3) Транспортный
- 4) Сеансовый

31. 5 К какому уровню модели OSI относится протокол HTTP

- 1) Канальный
- 2) Сетевой
- 3) Транспортный
- 4) Сеансовый

32. 6 Пусть записано утверждение относительно компьютерной сети в виде 4 чисел, например, 191.200.182.101. Что это записано:

- 1) IP адрес
- 2) Порт
- 3) Сокет
- 4) Доменное имя

33. 7 Пусть записано утверждение относительно компьютерной сети в виде www.tusur.ru Что это записано:

- 1) IP адрес
- 2) Порт
- 3) Сокет
- 4) Доменное имя

34. 8 Пусть записано утверждение относительно компьютерной сети в виде 5 чисел, например, [191.200.182.101.],80 Что это записано:

- 1) IP адрес
- 2) Порт
- 3) Сокет
- 4) Доменное имя

35. 9 Если при обработке запроса сервер выполняет основные операции, а клиент ожидает ответа, то говорят о

- 1) толстом клиенте
- 2) тонком клиенте
- 3) смешанном клиенте
- 4) многозвенной архитектуре

36. 10 Если при обработке запроса сервер выполняет основные операции, а клиент ожидает ответа, то говорят о

- 1) толстом клиенте
- 2) тонком клиенте
- 3) смешанном клиенте
- 4) многозвенной архитектуре

37. 11 Если при обработке запроса сервер и клиент выполняет приблизительно равное число операций, то говорят о

- 1) толстом клиенте
- 2) тонком клиенте
- 3) смешанном клиенте
- 4) многозвенной архитектуре

38. 12 Какое ключевое слово не обозначает метод протокола HTTP

- 1) POST
- 2) GET
- 3) HEAD

4) URL

39. 13 Какой из протоколов не относится к электронной почте

- 1) SMTP
- 2) MIME
- 3) POP
- 4) FTP

40. 14 Какая функция API связывает сокет с номером порта и IP-адресом

- 1) socket()
- 2) bind()
- 3) accept()
- 4) listen()

41. 15 Какая функция устанавливает очередь для запросов на соединение

- 1) socket()
- 2) bind()
- 3) accept()
- 4) listen()

42. 16 Какая функция принимает запрос на соединение

- 5) socket()
- 6) bind()
- 7) accept()
- 8) listen()

43. 17 Какой объект синхронизации не используется в системном режиме

- 1) Критическая секция
- 2) Mutex
- 3) Event
- 4) Semaphore

44. 18 Какой интерфейс не относится к программированию WWW сервера

- 1) CGI
- 2) PHP
- 3) ISAPI
- 4) WINSOCK

45. 19 Какой интерфейс из перечисленных используется в языке Java

- 1) Servlet
- 2) CGI
- 3) PHP
- 4) ISAPI

46. Как называется модуль для создания клиент-серверных приложений на языке Python

- 1) Servlet
- 2) Socket
- 3) PHP
- 4) ISAPI

43. Отладка это

- тестирование программного кода на этапе разработки программного обеспечения
- поиск ошибок при выполнении программ в тестовой или моделируемой среде
- попытка найти ошибки при выполнении программы в реальной среде

44. Контроль это

- тестирование программного кода на этапе разработки программного обеспечения
- поиск ошибок при выполнении программ в тестовой или моделируемой среде
- попытка найти ошибки при выполнении программы в реальной среде

45. Испытание это

- тестирование программного кода на этапе разработки программного обеспечения
- поиск ошибок при выполнении программ в тестовой или моделируемой среде
- попытка найти ошибки при выполнении программы в реальной среде

46. документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

- План тестирования
- Тест-дизайн
- Тест-кейс
- Баг-репорт

47. процесс тестирования ПО, на котором проектируются и создаются тест-кейсы в соответствии с определенными ранее критериями качества и целями тестирования.

- Тест-дизайн
- План тестирования
- Тест-кейс
- Баг-репорт

48. документ, описывающий последовательность действий, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или ее части.

- Тест-кейс
- Тест-дизайн
- План тестирования
- Баг-репорт

49. Тест-кейсы подразделяют по ожидаемому результату на:

- позитивные и негативные
- правильные и неправильные
- первичные и вторичные
- положительные и отрицательные

50. Каждый кейс-комплект содержит общую информацию о тестировании:

- название проекта
- номер версии
- имя тестера
- даты тестирования
- номер тестирования
- название теста
- количество тестов

51. План работы над тест-дизайном включает в себя:

- анализ документации (спецификации, требования, планы), модели, исполняемого кода и т.д.
- написание спецификации по тест-дизайну
- проектирование и создание
- критерии окончания тестирования (результаты тестирования удовлетворяют критериям качества продукта);
- необходимое для тестирования оборудование и программные средства (тестовый стенд и его конфигурация, программы для автоматизированного тестирования и т.д.)

52. В соответствии с этапом обработки, на котором появляются ошибки, различают:

- синтаксические ошибки
- ошибки компоновки
- ошибки выполнения
- глобальная ошибка

53. ошибки, фиксируемые компилятором (транслятором, интерпретатором) при выполнении синтаксического и частично семантического анализа программы

54. Ошибки выполнения подразделяются на:

- ошибки определения данных
- логические ошибки
- ошибки накопления погрешностей
- ошибки компоновки
- синтаксические ошибки
- ошибки определения переменных

55. Методика отладки программного обеспечения разделяется на:

- 3 этапа
- 5 этапов
- 2 этапа
- 4 этапа

56. Расставить этапы методики отладки ПО в правильном порядке

- изучение проявления ошибок
- локализация ошибок
- определение причины ошибки
- исправление ошибки

- повторное тестирование

1. Программная инженерия:

- **software engineering**

- Инструменты создания программного обеспечения
- Коллектив инженеров-программистов, разрабатывающих программное обеспечение для компьютеров
- **Дисциплина, изучающая применение строгого систематического количественного подхода к разработке, эксплуатации и сопровождению программного обеспечения**
- Комплекс программ, предназначенный для решения инженерных задач, связанных с большим количеством расчетов
- Инженерная индустрия применения прикладного программного обеспечения
- **Совокупность инженерных методов и средств создания программного обеспечения**
- Прикладное программное обеспечение для решения офисных задач

2. Построение SADT-модели включает в себя выполнение следующих действий:

- Написание программного обеспечения для разрабатываемой системы по требованиям заказчика
- **Сбор информации об объекте, определение его границ**
- **Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм**
- Представление исследуемой системы в графическом виде
- Представление исследуемого объекта средствами системного моделирования
- **Критическая оценка, рецензирование и комментирование**
- Разработка, отладка и тестирование программного обеспечения
- Использование графических пакетов для представления системы в виде модели

3. Моделирование основывается на принципах:

- **Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение**
- Декомпозиции системы на отдельные подзадачи
- Инкапсуляции и полиморфизма
- Децентрализации управления системой
- **Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности**
- Открытой трансформируемой системы
- **Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, почти независимых друг от друга**
- Анализа и синтеза проектирования систем

4. В бизнес-процессах выделяют классы процессов:

- Решающие бизнес-процессы
- Регламентирующие бизнес-процессы
- **Основные бизнес-процессы**
- Бизнес-процессы поведения системы
- Программируемые бизнес-процессы
- Экономические бизнес-процессы
- **Обеспечивающие бизнес-процессы**
- **Бизнес-процессы управления**

5. CASE-средства классифицируются по следующим признакам:

- **По применяемым методологиям и моделям систем и БД**
- По используемому программному обеспечению
- По этапам жизненного цикла программного обеспечения
- **По степени интегрированности с СУБД**
- По уровням детализации и декомпозиции проектируемой системы
- **По доступным платформам**
- По используемым языкам программирования
- По степени сложности моделируемой системы

6. К малым интегрированным средствам моделирования относятся:

- ARIS Toolset
- Design/IDEF
- **ERwin**
- **BPwin**
- Designer/2000
- Paradigm Plus
- **Model Mart**
- Rational Rose

7. К средним интегрированным средствам моделирования относятся:

- Rational Rose
- **Design/IDEF**
- BPwin
- **Designer/2000**
- **ARIS Toolset**
- Model Mart
- Paradigm Plus
- ERwin

8. Объектно-ориентированная методология (ООМ) включает в себя составные части:

- **Объектно-ориентированный анализ**
- Объектно-ориентированный подкласс
- **Объектно-ориентированное проектирование**
- Объектно-ориентированная парадигма
- Объектно-ориентированная экспозиция
- Объектно-ориентированное моделирование
- **Объектно-ориентированное программирование**
- Объектно-ориентированная декомпозиция

9. К основным понятиям объектно-ориентированного подхода относятся:

- Обобщение
- **Полиморфизм**
- **Инкапсуляция**
- Реализация
- Агрегирование
- **Наследование**
- Ассоциация
- Композиция

10. Главные принципы объектного подхода:

- **Абстрагирование**

- Наследование
- **Ограничение доступа или инкапсуляция**
- Безграничный доступ или инкапсуляция
- **Модульность и иерархия**
- Агрегирование
- Композиция
- Обобщение и специализация

11. Дополнительные принципы объектного подхода:

- Реализация
- **Типизация**
- **Параллелизм**
- Внедрение
- Перпендикулярность
- **Сохраняемость или устойчивость**
- Несохранимость или неустойчивость
- Динамичность

12. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся:

- **Rational Rose**
- Model Mart
- **MS Visio**
- **ARIS**
- IDEF1X
- Erwin
- BPwin
- JAM

13. К инструментальным средствам представления функциональных моделей относятся:

- JAM
- **Model Mart**
- MS Visio
- ARIS
- IDEF0
- **Erwin**
- **BPwin**
- Rational Rose

14. Методологии, поддерживаемые в BPwin:

- IDEF1X
- **IDEF0**
- IDEF1
- **IDEF3**
- IDEFX
- IDEF5
- **DFD**
- DFD1X

15. Диаграмма IDEF0 может содержать следующие типы диаграмм:

- Диаграмму классов
- **Контекстную диаграмму, диаграмму декомпозиции**

- Диаграмму компонентов
- **Диаграмму дерева узлов**
- Диаграмму взаимодействий
- **Диаграмму только для экспозиции (FEO)**
- Диаграмму последовательности, диаграмму кооперации
- Диаграмму узлов

16. Уровни логической модели:

- Диаграмма сущность
- Диаграмма связь
- Диаграмма пакетов
- **Диаграмма сущность-связь**
- Модель данных, основанная на классах
- **Модель данных, основанная на ключах**
- Полная операционная модель
- **Полная атрибутивная модель**

17. Внутренние стрелки не входящие в состав диаграммы IDEF0:

- **mechanism- output**
- output-input
- **mechanism- input**
- output-control
- output-input feedback
- output-control feedback
- output-mechanism
- **control feedback- mechanism**

18. Типы стрелок не входящие в состав диаграммы IDEF0:

- Input
- **Editor**
- Control
- **Properties**
- Output
- Mechanism
- Call
- **Dictionary**

19. Quick Reports – создание простейших отчетов – позволяет создавать отчеты:

- Group/Totals. Табличный отчет с автоматической группировкой и сортировкой данных
- Report Header. Печатается единожды в начале отчета
- **Columnar. Простой табличный отчет**
- Page Header. Печатается в верхней части каждой страницы
- **Vertical. Простой вертикальный отчет**
- Group Header. Печатается в начале каждой группы
- **Blank Report. Бланк. Создается пустой бланк отчета, в который не включаются данные**
- Detail. Печатается для каждой строчки набора данных

20. BPwin допускает следующие переходы с одной нотации на другую:

- IDEF3 → DFD
- DFD → IDEF0
- **IDEF0 → DFD**

- DFD → DFD
- IDEF3 → IDEF0
- IDEF0 → IDEF3
- IDEF3 → IDEF3
- **DFD → IDEF3**

21. DFD описывает:

- Функции обработки стрелок (arrow)
- **Функции обработки информации (работы)**
- Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации
- **Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации**
- Функции обработки внешних ссылок
- **Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data stor+ E)**
- Функции обработки документов
- Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке внешних стрелок

22. ВРwin позволяет создавать на диаграмме DFD типы граничных стрелок:

- **Обычная граничная стрелка**
- Специальная стрелка
- Внутренняя ссылка
- **Межстраничная ссылка и тоннельная стрелка**
- **Внешняя ссылка**
- Страничная ссылка и тневая стрелка
- Контрольная стрелка
- Стрелка механизм

23. Создать отчет в ВРwin возможно с помощью:

- **Встроенных шаблонов**
- Программных модулей, создаваемых разработчиком на языке Visual Basic
- Создать отчет в ВРwin не возможно
- **Report Template Builder**
- Отчет создается разработчиком
- Отдельно поставляемых программ
- Встроенных мастер-функций
- **RPTwin**

24. В ВРwin 4.0 отчеты могут быть экспортированы в распространенные форматы:

- **Текстовый**
- Символьный
- **MS Office**
- Графический
- **HTML**
- Internet Explorer
- Acrobat
- IBM Rational

25. Поддерживаемые в RPTwin типы операторов:

- **Текстовый оператор конкатенации (&)**

- Символ
- Текст
- Дата
- **Арифметические**
- Графический оператор конкатенации (&)
- **Логические**
- Номер

26. Инструментальное средство ERwin позволяет:

- Редактировать и отлаживать программы
- **Проектировать на физическом и логическом уровне модели данных**
- Управлять процессом конструирования ПО
- Проектировать диаграммы вариантов использования и взаимодействий
- **Проводить процессы прямого и обратного проектирования баз данных**
- Управлять процессом трансляции и отладки программ
- **Выравнивать модель и содержимое системного каталога после редактирования**
- Проектировать контекстные диаграммы и диаграммы декомпозиции

27. ERwin позволяет создавать модели следующих типов:

- **Модель, имеющую только логический уровень**
- Модель, имеющую абстрактный уровень
- Модель, имеющую абстрактный и физический уровни
- **Модель, имеющую только физический уровень**
- Модель, имеющую абстрактный и логический уровни
- **Модель, имеющую как логический уровень, так и физический уровень**
- Модель, имеющую концептуальный уровень
- Модель, имеющую контекстный уровень

28. Для создания моделей ERwin используют международно признанные системы обозначений (нотации):

- IDEF0
- **IDEF1X**
- IDEF3
- DFD
- **IE**
- **DM**
- IDEFDFD
- IDEF3

29. К основным компонентам диаграммы ERwin относятся:

- **Сущности**
- Переходы
- **Атрибуты**
- Классы
- Слияния
- Разветвления
- Использования
- **Связи**

30. Точки зрения организации в ARIS:

- Структура внедрения и структура потоков
- **Организационная структура**

- Управленческая структура
- Поведенческая структура
- **Функциональная структура**
- Коммуникационная структура
- **Структура данных и структура процессов**
- Обобщенная структура

31. Уровни точки зрения в ARIS:

- Описание структуры
- **Описание требований**
- Описание поведения
- Описание разработки
- **Описание спецификации**
- **Описание внедрения**
- Описание процессов
- Описание классов

32. Методы описания, используемые в ARIS:

- EPT – метод описания потоков
- **EPC - метод описания процессов**
- ERM - модель сущность-связь для описания структуры объектов
- **ERM - модель сущность-связь для описания структуры данных**
- EPP – метод описания пакетов
- EPC – метод описания компонентов
- **UML - унифицированный язык моделирования**
- EPT – метод описания нитей

33. К основным компонентам инструментов ARIS Toolset относятся:

- Internet (интернет)
- WordPad (ввод текстовых данных)
- Media (средство для медиа описания моделей)
- **Explorer (проводник)**
- Acrobat (чтение текстовых данных)
- **Designer (средство для графического описания моделей)**
- Document (для ввода различных параметров и атрибутов) и выноски
- **Таблица (для ввода различных параметров и атрибутов) и мастер (Wizards)**

34. ARIS Business Optimizer позволяет:

- **Определять целевые затраты и рассчитывать стоимость продукта: во что компании обходится предоставление отдельных продуктов**
- Принимать решения о времени начала и окончания работы над проектом
- **Принимать решения по аутсорсингу: стоит ли поручить выполнение бизнес-процессов внешнему поставщику услуг**
- Определять последовательность работ, выполняемых в ходе работы над проектом
- Определять требования к персоналу компании, которая в дальнейшем будет эксплуатировать программное обеспечение
- Рассчитывать заработную плату сотрудников компании после внедрения программного обеспечения
- Планировать требования к обслуживающему персоналу, сопровождающему программное обеспечение
- **Планировать требования к персоналу: сколько необходимо сотрудников для оптимального выполнения работ**

35. «Взгляды» ARIS:

- **Процессы**
- Потоки
- **Функции (с целями)**
- **Данные и организация**
- Процедуры
- Управление и внедрение
- Нити
- Память

36. Уровни анализа ARIS для каждого «взгляда»:

- Поведение
- **Требования**
- **Спецификации**
- Функции
- Процедуры
- Проверка
- **Внедрение**
- Тестирование

37. MS Visio позволяет создавать схемы, чертежи, диаграммы с помощью:

- **Встроенных шаблонов**
- Панели инструментов
- **Трафаретов**
- Графических редакторов
- Дополнительного программного обеспечения
- Панели рисования
- **Стандартных модулей**
- Панели автофигур

38. Язык UML – это:

- Язык программирования высокого уровня
- **Унифицированный язык моделирования**
- Язык для разработки систем искусственного интеллекта
- **Unified Modeling Language**
- Язык управления базами данных
- **Язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем**
- Язык создания запросов в базах данных
- Язык программирования низкого уровня

39. Моделирование в UML позволяет решать задачи:

- Анализа и синтеза систем управления
- Разработать и отладить программное обеспечение
- **Визуализировать систему в ее текущем или желательном для нас состоянии**
- Провести тестирование разработанного программного обеспечения
- **Описать структуру или поведение системы; получить шаблон, позволяющий сконструировать систему**
- Смоделировать разрабатываемую информационную систему
- **Документировать принимаемые решения, используя полученные модели**
- Рассчитать экономическую эффективность от внедрения программного обеспечения

40. Словарь UML включает строительные блоки:

- Зависимости
- **Сущности**
- Слияния
- Разветвления
- **Связи**
- Группировки
- **Диаграммы**
- Декомпозиции

41. UML, как язык документирования, помимо исполняемого кода производит и другие продукты, включающие:

- **Требования, архитектуру, проектные решения**
- Спецификацию технических средств
- **Дизайн, исходный код, проектные планы,**
- Требования к уровню квалификации разработчиков
- Набор заданий для тестирования программного обеспечения
- Требования к уровню квалификации персонала сопровождения
- **Тесты, прототипы, релизы (версии)**
- Требования к выбору языка программирования

42. UML включает синтаксические и семантические правила для:

- Агрегации
- Тестирования
- **Имен, областей действия**
- Сборки
- Сопровождения
- **Видимости, целостности**
- Вывода из эксплуатации
- **Исполнения**

43. Применение языка UML существенно упрощает последовательное использование механизмов:

- **Спецификации, дополнения**
- **Принятые разделения**
- Выработки требований
- Создания плана работ
- **Механизмы расширения**
- Тестирования программного обеспечения
- Конструирования ПО
- Сопровождения ПО

44. Механизмы расширения UML включают:

- Исключения
- **Стереотипы**
- Дополнения
- Управления
- **Помеченные значения**
- Слияния
- **Ограничения**
- Объединения

45. Язык UML предназначен для:

- **Визуализации**
- Тестирования
- Сопровождения
- **Специфицирования**
- Снятия с эксплуатации
- **Конструирования, документирования**
- Анализа требований
- Обучения персонала

46. В объектно-ориентированном моделировании между классами существуют типы связей:

- Слияние
- Линейность
- **Зависимость**
- Разветвление
- Цикличность
- **Обобщение**
- **Ассоциация**
- Агрегация

47. В состав графического представления класса в языке UML входят части:

- Отношения
- **Имя**
- Связи
- **Атрибуты**
- Описание
- Сущности
- **Операции**
- Механизмы

48. Программное обеспечение делится на классы:

- Системное ПО и прикладное ПО
- **Системное ПО, прикладное ПО и инструментальные средства разработки программ**
- Операционные системы, прикладное ПО, утилиты и драйверы
- Прикладное ПО и инструментальные средства разработки программ
- Системное ПО и инструментальные средства разработки программ
- **Системное ПО, прикладное ПО и системы программирования**
- Операционные оболочки, операционные системы, офисные программы
- **Системное ПО, прикладное ПО и инструментальное ПО**

49. Инструментальные средства разработки программ – это:

- **Средства создания новых программ**
- Сервисные средства разработки ПО
- Аналитические средства разработки ПО
- **Программное обеспечение, предназначенное для разработки и отладки новых программ**
- Средства отладки ПО
- Средства тестирования ПО
- **Аппаратные и программные инструменты разработки нового ПО**
- Технические инструментальные средства разработки ПО

50. Аппаратные инструментальные средства разработки ПО – это:

- Система для разработки новых программ на конкретном языке программирования
- Средства создания и редактирования текстов программ
- Микропроцессор и подключаемые (внешние) устройства
- Устройства вычислительной системы, специально предназначенные для поддержки разработки ПО
- Периферийные устройства, микропроцессор вычислительного комплекса, предназначенные для разработки нового ПО
- Программное обеспечение, написанное на языках программирования низкого уровня
- Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ
- Программы, используемые для корректировки и тестирования других прикладных или системных программ

51. Программные инструментальные средства разработки ПО – это:

- **Программы, позволяющие выполнить все работы, определенные методологией проектирования ПО**
- Системное программное обеспечение, позволяющее сопровождать офисные программные пакеты
- Средства создания текстовых документов
- **Программное обеспечение, используемое на всех стадиях разработки нового ПО**
- Программное обеспечение для настройки офисных приложений на условия конкретного применения
- **Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ**
- Устройство компьютера, специально предназначенное для поддержки разработки программных средств
- Средства создания и редактирования текстовых документов

52. Транслятор – это:

- **Программа, выполняющая перевод программы с одного языка программирования на другой**
- Комплекс программ мультимедийных технологий
- **Программа, которая выполняет перевод программы с одного языка программирования на машинные коды**
- Программа-переводчик с одного иностранного языка на другой
- Техническое устройство передачи и преобразования аудио и видеосигналов
- Техническое устройство для кодирования и декодирования информации
- Программное обеспечение для обеспечения защиты информации на компьютере
- Одно из основных средств автоматизации программирования для преобразования программы, написанной на машинно-независимом языке, в программу на машинном языке конкретной ЭВМ

53. Компилятор – это:

- **Один из видов трансляторов**
- Прикладное программное обеспечение
- Специальная утилита системного ПО
- Операционная оболочка
- **Переводит в коды сразу всю программу и создает независимый исполняемый файл**
- Программное обеспечение, используемое в издательских системах
- **Программа, которая переводит программу, написанную на языке программирования высокого уровня в программу на машинном языке не участвуя в**

ее исполнении

- Переводит в машинные коды 1 строчку программы и сразу ее выполняет

54. Интерпретатор:

- Программа для создания и редактирования электронных таблиц
- Программа, анализирующая команды или операторы исходной программы и немедленно выполняющая их
- Переводит в коды сразу всю программу и создает независимый исполняемый файл
- **Переводит в машинные коды 1 строчку программы и сразу ее выполняет**
- Программа для создания и редактирования текстовых документов
- **Один из видов трансляторов**
- Программа создания и управления базами данных
- Программа создания файлов мультимедиа

55. Компоновщик – это:

- Программа для компоновки и оформления тестовых документов
- **Редактор связей**
- Комплекс программ, для создания и ведения баз данных
- Программа, которая из одного или нескольких объектных модулей с привлечением библиотечных программ и стандартных подпрограмм формирует загрузочный модуль
- Программное обеспечение для создания презентаций
- Программа сборки загрузочного модуля из полученных в результате отдельной компиляции объектных модулей с автоматическим поиском и присоединением **библиотечных подпрограмм и процедур**
- Программа для поиска синтаксических и семантических ошибок в программе
- Программа

56. Отладчик:

- **Программа, облегчающая программисту выполнение отладки разрабатываемых им программ**
- Программа для создания системы защиты файла
- Программа создания системы защиты от вирусных атак
- **Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку**
- Операционная оболочка для создания и управления файловыми структурами
- Системное программное обеспечение для настройки операционной системы
- Программа создания и редактирования графических файлов
- Программа, позволяющая выполнять остановы в заданных точках, просмотреть текущие значения переменных и изменять их значения

57. К этапам развития технологии разработки программного обеспечения относятся:

- **«Процедурное» программирование**
- Программирование на алгоритмических языках высокого уровня
- **Структурный подход к программированию**
- Программирование на языках низкого уровня
- **Компонентный подход и CASE-технологии**
- Машинно-ориентированное программирование
- Машинно-независимое программирование
- Подход к разработке ПО, основанный на стратегии поиска

58. «Стихийное» программирование:

- Разработка программного обеспечения без предварительного составления плана-графики

работ

- Первый этап в истории развития технологии разработки программного обеспечения, когда программирование фактически было искусством
- Период в истории разработки программного обеспечения, когда программа создавалась одним программистом, способным отслеживать последовательность выполняемых операций и местонахождения данных в программе
- Разработка программ с использованием различных языков программирования низкого и высокого уровня
- Разработка программ с элементами случайного выбора алгоритмов решения задачи
- Характеризуется тем, что типичная программа этого периода состояла из основной программы, области глобальных данных и набора подпрограмм (в основном библиотечных), выполняющих обработку всех данных или их части
- Разработка программного обеспечения для решения задач теории вероятностей и математической статистики
- Разработка программного обеспечения для решения задач, построенных на алгоритмах случайного поиска

59. Структурный подход к программированию – это:

- Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
- Создание программного обеспечения на основе структурной схемы решаемой задачи
- Подход, требующий разработки структурной схемы алгоритма и программы решения задачи
- Подход, в основе которого лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм
- Подход к решению задачи, требующий создание структурной схемы этапов работ по разработке программного обеспечения
- Процесс создания программного обеспечения на основе структурной схемы исследуемого объекта или процесса
- Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
- **Подход, требующий представления задачи в виде иерархии подзадач простейшей структуры**

60. Объектный подход к программированию – это:

- Технология создания сложного программного обеспечения, основанная на представлении задачи исследования как объекта
- Технология создания сложного программного обеспечения, предназначенного для автоматизации технологических объектов
- Технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств
- Технология создания сложного программного обеспечения, основанная на представлении программы как единого объекта
- **Технология создания сложного программного обеспечения, позволяющая вести практически независимую разработку отдельных частей (объектов) программы**
- Технология создания сложного программного обеспечения, основанная на объектном представлении кода программы
- Технология создания сложного программного обеспечения, в основе которой лежат новые способы организации программ, основанные на механизмах наследования,

полиморфизма, композиции, наполнения

- Технология создания сложного программного обеспечения, основанная на объектно-ориентированном программировании

61. Компонентный подход:

- Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения

- Предполагает взаимодействие между компонентами через стандартизированные двоичные интерфейсы и позволяет использовать исполняемые файлы в любом языке программирования, поддерживающем соответствующую технологию

- Позволяет рассматривать объект исследования, как структуру, состоящую из отдельных компонент

- способ написания исходного кода программного обеспечения

- **Позволяет собрать объекты-компоненты в динамически вызываемые библиотеки или исполняемые файлы, и распространять в двоичном виде (без исходных текстов)**

- Способ отладки и тестирования программного обеспечения

- Способ внедрения и опытной эксплуатации программного обеспечения.

- Метод выработки требований к разработке программного обеспечения

62. Управление требованиями:

- Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям

- **Процесс систематического выявления, организации и документирования требований к сложной системе**

- Выявление требований заказчика и управление ими

- **Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности**

- Процесс создания программного обеспечения и адаптация его под требования заказчика

- Разработка требований к программному обеспечению и создание ПО на основе этих требований

- **Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе**

- Разработка программного обеспечения и выработка требований к изменению работы системы заказчика

63. К методам выявления требований относятся:

- Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение

- Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения

- Личные встречи и беседы со всеми сотрудниками предприятия

- Анализ технической документации и на основе нее разработка требований к системе

- На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения

- Интервьюирование и анкетирование, мозговой штурм и отбор идей

- Совещания, посвященные требованиям, создание прототипов

- Раскадровки, прецеденты, обыгрывание ролей

64. Требования к разрабатываемой системе должны включать:

- Разработку программного обеспечения и выработку требований к изменению работы системы заказчика

- Совокупность условий, при которых предполагается эксплуатировать будущую систему (аппаратные и программные ресурсы, предоставляемые системе; внешние условия ее функционирования; состав людей и работ, имеющих к ней отношение)
- Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
- **Описание выполняемых системой функций**
- Технологию создания сложного программного обеспечения, основанную а объектном представлении кода программы
- Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, **обеспечивающие защиту информации**)
- Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
- Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования

65. Типы средств, иллюстрирующие цели моделирования системы:

- Функции, которые система должна выполнять
- Отношения между данными
- **Зависящее от времени поведение системы (аспекты реального времени)**
- Способы отладки и тестирования программного обеспечения
- Создание программного обеспечения на основе структурной схемы исследуемого объекта или процесса
- Выявление требований заказчика и управление ими
- Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
- Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения

66. Преимущества объектно-ориентированного подхода:

- Быстрота написания программного кода
- Статичность конфигурации системы
- **Возможность многократного использования**
- Низкая стоимость проекта
- **Восприимчивость к изменениям**
- Отсутствие необходимости документирования
- Простота реализуемых моделей
- **Реалистичное моделирование**

67. Требования – это:

- Документ, регулирующий отношения между заказчиком информационной системы и проектировщиком
- **Некоторое свойство программного обеспечения, необходимое пользователю для решения проблемы при достижении поставленной цели**
- Оформленное заказчиком в виде документа задание на проектирование программного обеспечения
- **Возможность, которую должна обеспечивать система**
- Характеристика проектируемого программного обеспечения с точки зрения разработчика
- **Некоторое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить требования формальной документации**
- Оформленное разработчиком в виде документа задание на проектирование

программного обеспечения

- Характеристика проектируемого программного обеспечения с точки зрения заказчика

68. Типичная схема процесса анализа С-требований включает в себя:

- **Идентификацию заказчика и проведение интервью с представителями заказчика**
- Разработку программного обеспечения в соответствии с требованиями заказчика
- Изложение заказчику требований к системе на основе разработанного программного обеспечения
- **Написание С-Требований в форме стандартного документа**
- Верификацию разработанного программного обеспечения в соответствии с требованиями заказчика
- Составление плана мероприятий по анализу С-требований
- **Проверку С-Требований и согласование их с заказчиком**
- Адаптацию разработанного программного обеспечения в соответствии с требованиями заказчика

69. В классификацию требований к программной системе входят:

- Требования заказчика
- Требования, накладываемые условиями эксплуатации
- **Функциональные требования**
- Требования, накладываемые аппаратными средствами
- **Нефункциональные требования**
- **Требования предметной области**
- Экономические требования
- Требования разработчиков

70. Процесс определения и анализа требований включает в себя:

- Анализ работы систем с аналогичной предметной областью
- **Анализ предметной области, сбор и классификацию требований**
- Проведение совместных совещаний с представителями заказчика
- **Разрешение противоречий и определение приоритетов**
- Адаптацию требований к разрабатываемому программному обеспечению
- Декомпозицию общей задачи на подзадачи
- **Проверку, специфицирование и документирование требований**
- Верификацию требований в соответствии с разработанным программным обеспечением

71. Опорные точки зрения конечных пользователей системы программного обеспечения можно трактовать как:

- **Источник информации о системных данных**
- Структуру требований
- Источник событий
- Структуру событий
- **Структуру представлений**
- Получателей требований
- Источник сценариев
- **Получателей системных сервисов**

72. При аттестации требований выполняются следующие типы проверок документации требований:

- Проверка на совместимость
- Проверка на управляемость
- **Проверка правильности требований**

- **Проверка на непротиворечивость**
- Проверка на соответствие
- Проверка на обратимость
- **Проверка на полноту и на выполнимость**
- Проверка на заменяемость

73. К методам аттестации требований относится:

- Тестирование
- **Обзор требований**
- Верификация
- Сравнительный анализ
- **Прототипирование**
- Генерация случайных данных
- **Генерация тестовых сценариев**
- Декомпозиция

74. Уровни организационного управления при планировании разработки системы:

- **Стратегический**
- **Тактический**
- **Оперативный**
- Основной
- Вспомогательный
- Дополнительный
- Системный
- Аналитический

75. Для различных представлений проектируемой системы используют типы моделей:

- Статическая модель
- Динамическая модель
- **Модель классов**
- Модель декомпозиции
- Модель размещения
- **Модель состояний**
- **Модель взаимодействия**
- Модель агрегации

76. Классификация бизнес-процессов включает следующие классы процессов:

- Вспомогательные бизнес-процессы
- Основные бизнес-процессы
- Дополнительные бизнес-процессы
- Обеспечивающие бизнес-процессы
- Обслуживающие бизнес-процессы
- Бизнес-процессы согласования
- Бизнес-процессы управления
- Руководящие бизнес-процессы

77. Типы D-требований:

- Функциональные требования
- Интерфейсные требования
- Нефункциональные требования
- Программные требования
- Обратные требования

- Ограниченные требования
- Производительные требования
- Надежность

78. Возможные способы организации D-требований:

- По атрибутам, по компонентам
- По взаимоотношениям сущности
- По пакетам и по иерархии компонентов
- По свойствам, по классам
- По вариантам использования
- По узлам и по использованным процессам
- По состояниям и по иерархии функции
- По прецедентам, по кооперациям

79. К моделированию относится:

- Система обозначений
- Система атрибутов
- Синтаксис языка моделирования
- Система свойств
- Совокупность поведения объектов
- Совокупность графических объектов
- Семантика языка моделирования
- Совокупность текстовых объектов

80. Классификация имитационных моделей:

- Статистическая
- Адаптивная
- Статическая или динамическая
- Структурная
- Сетерминированная или стохастическая
- Непрерывная или дискретная
- Объединенная
- Декомпозиционная

81. Принципы разработки эффективного пользовательского интерфейса:

- Сложность, графика
- Структура, простота
- Связь, обработка
- Видимость, обратная связь
- Невидимость, сложность
- Толерантность, повторное использование
- Первое использование, итерация
- Интеграция, повторение

82. Принципы разработки программного обеспечения:

- Коллективный процесс разработки
- Индивидуальный процесс разработки
- Параллельный процесс разработки
- Командный процесс разработки
- Промежуточный процесс разработки
- Модель зрелости возможностей

- Модель законченности возможностей
- Модель готовности процессов

83. Типы интерфейсных требований:

- Пользовательские требования
- Аппаратные требования
- Административные требования
- Требования к производительности
- Программные и коммуникационные требования
- Требования к надежности
- Требования к устойчивости
- Атрибуты программной системы и другие требования

84. Технология проектирования определяется как совокупность составляющих:

- Поэтапная процедура
- Пошаговая процедура
- Модели и правила
- Критерий и правила
- Тестирование
- Нотаций
- Прецеденты
- Классы

85. Разработка и сопровождение ИС в конкретной организации и конкретном проекте должна поддерживаться стандартами:

- Стандарт организации
- Стандарт конкретного проекта
- Стандарт проектирования
- Стандарт оценки
- Стандарт оформления проектной документации
- Стандарт аудита
- Стандарт оформления разработки
- Стандарт пользовательского интерфейса

86. Результатами проектирования архитектуры являются:

- Модель административного интерфейса
- Модель процессов
- Модель потоков
- Модель классов
- Модель данных
- Модель пользовательского интерфейса
- Модель компонентов
- Модель узлов

87. Какие работы включает процесс разработки программного обеспечения:

- Документирование, управление конфигурацией
- Управление, создание инфраструктуры
- Структура из процессов, работ, задач
- Обеспечение качества, верификация
- Анализ требований, проектирование
- Программирование, сборка, тестирование

- Ввод в действие, приемка
- Совместный анализ, аудит

88. Какие технологии разработки программ используются в современном программировании:

- Визуальные
- Событийные
- Структурные
- Объектно-ориентированные
- Модульные
- Текстуальные
- Графические
- Машинно-ориентированное

89. Объектно-ориентированное проектирование использует инструментальные средства:

- Model mart
- Rational Rose
- Bpwin
- ARIS
- Idef1X
- Erwin
- MS Visio
- Jam

90. Проектирование функциональных моделей поддерживается инструментальными средствами:

- Jam
- Model Mart
- MS visio
- ERwin
- Idef0
- Aris
- Rational rose
- BPwin

91. IEEE – это:

- Коммерческая организация ученых и исследователей
- Просто принятое обозначение, расшифровки не имеет
- Обозначение всемирной компьютерной сети
- Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей
- Такая аббревиатура нигде не используется
- Institute Of Electrical and Electronic Engineers, Inc
- Американская организация ученых-экономистов
- Институт инженеров радиоэлектроники и электротехники

92. Ядро знаний SWEBOOK – это:

- ГОСТ на разработку программного обеспечения
- Нормативный документ, разработанный IEEE
- ГОСТ на разработку информационных систем
- Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения

- основополагающий научно-технический документ, который отображает мнение специалистов в области программной инженерии
- документ, устанавливающий методику тестирования и испытания программного обеспечения
- документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207
- ГОСТ на разработку и комплектацию сопровождающей документации

93. Каждая область ядра знаний SWEBOOK представляется:

- Структурной схемой
- Общей схемой описания
- Диаграммой UML
- Описанием и комментариями
- Определением понятийного аппарата, методов и средств инженерной деятельности
- Определением языка программирования
- Определением инструментов поддержки инженерной деятельности
- Иерархической диаграммой

94. К основным областям знаний SWEBOOK относятся:

- Инженерия требований, проектирование ПО
- Анализ деятельности системы
- Управление проектами
- Конструирование ПО
- Управление персоналом
- Тестирование ПО, сопровождение ПО
- Управление конфигурацией
- Инженерия качества программных средств

95. К организационным областям знаний SWEBOOK относятся:

- Инженерия требований
- Управление конфигурацией, управление проектами
- Конструирование ПО
- Процесс инженерии программных средств, методы и средства программной инженерии
- Проектирование ПО
- Сопровождение ПО
- Тестирование ПО
- Инженерия качества программных средств

96. В рамках Rational Unified Process (RUP) набор действий по разработке программ включает этапы:

- Создание структурных схем
- Определения входных, выходных данных
- Согласование стоимости проекта
- Согласования требований с заказчиком
- Создания бизнес-моделей
- Определение требований
- Проектирование, программирование
- Тестирование, внедрение

97. Этапы разработки консалтинговых проектов включают в себя:

- Анализ первичных требований и планирование работ
- Снятие программного продукта с эксплуатации

- Декомпозицию задачи на подзадачи
- Разработку спецификации и документации
- Проведение обследования деятельности предприятия
- Тестирование и сопровождение программного обеспечения
- Построение моделей деятельности предприятия (модели AS – IS – “как есть” и модели TO – BE – “как должно быть”)
- Разработку программного обеспечения

98. Концепции, лежащие в основе модульного программирования:

- Объем реализации и время исполнения (реакции)
- Мера автоматизма в работе реализации и инструментах разработки
- Визуальность и тестируемость разработки
- Функциональная декомпозиция, пространственная и временная группировка информации (модульность)
- Упрощение связей
- Комментируемость функций и данных
- Надежность, устойчивость
- Безопасность

99. Инструмент разработки программ выбирается на основе:

- Визуальности, набора реализуемых технологий
- Мощности множества элементов разработки
- Системного подхода к анализу, проектированию и реализации ПО
- Функциональной декомпозиции, пространственной и временной группировка информации (модульность)
- Упрощения связей, комментируемости функций и данных
- Объема реализации и времени исполнения (реакции), надежности, устойчивости, безопасности
- Меры автоматизма в работе реализации и инструментах разработки
- Визуальности и тестируемости разработки

6.5. Курсовая работа

Не предусмотрено.

6.5. Вопросы к зачету

- 1 Раскройте понятие «протокол», укажите основное назначение протоколов IP, TCP, UDP.
- 2 Дайте определение понятию «сокет».
- 3 Перечислите основные функции для работы с сокетами.
- 4 Укажите последовательность вызовов функций для организации сокета на стороне клиен-
- 5 Укажите последовательность вызовов функций для организации сокета на стороне сервере-
- 6 Опишите основную схему организации сервера, основанного на сокетах.
- 7 Опишите основную схему организации клиента, основанного на сокетах.
- 8 Раскройте протокол HTTP.
- 9 Опишите структуру WWW сервера.
- 10 Опишите функции и методы модуля socket
- 11 Опишите функции и методы модуля threading
- 12 Запишите общую структуру сервера, реализованного на сокетах.

- 13 Запишите организацию основного цикла сервера.
- 14 Запишите основные механизмы обмена данными.
- 15 Опишите способы организации мульти-поточности
- 16 Раскройте механизмы синхронизации потоков с помощью модуля threading
- 17 Раскройте отличия объектов семафора от объекта события
- 18 Что такое сервлет?
- 19 Какие еще существуют технологии, похожие на сервлеты?
- 20 Какова структура каталогов web-приложения?
- 21 Какой класс является базовым для сервлетов?
- 22 Каков жизненный цикл у сервлета?
- 23 Каким образом послать ответ клиенту?
- 24 Каковы основные цели мониторинга сетевого трафика?
- 25 Чем отличается мониторинг трафика от фильтрации?
- 26 Основные функции и возможности Zabbix.

6.6. Вопросы к экзамену

Не предусмотрено.

1. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Основная литература

1. Коваленко, В. В. Проектирование информационных систем : учебное пособие / В.В. Коваленко. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2021. — 357 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/987869. - ISBN 978-5-00091-637-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/987869> (дата обращения: 01.04.2022). – Режим доступа: по подписке.
2. Заботина, Н. Н. Проектирование информационных систем : учебное пособие / Н. Н. Заботина. — Москва : ИНФРА-М, 2020. — 331 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-004509-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1036508> (дата обращения: 01.04.2022). – Режим доступа: по подписке.

7.2. Дополнительная литература

1. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев ; под научной редакцией Л. Г. Доросинского. — Москва : Издательство Юрайт, 2022. — 90 с. — (Высшее образование). — ISBN 978-5-9916-9975-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492224> (дата обращения: 10.02.2022).
2. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2022. — 218 с. — (Высшее образование). — ISBN 978-5-534-00515-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490128> (дата обращения: 10.02.2022).
3. Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2022. — 204 с. — (Высшее образование).

образование). — ISBN 978-5-534-13715-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496682> (дата обращения: 10.02.2022).

7.2. Программное обеспечение

1. Сетевой компьютерный класс, оснащенный современной техникой
2. Офисный программный пакет (например, Microsoft Office 2007 или более поздних версий).
3. Web-браузер Edge, Mozilla Firefox или Google Chrome
4. ПО для вывода на экран для проектора
5. Платформа Java.
6. Сетевой симулятор JavaNetSim.
7. Менеджер виртуальных машин VMware Player или VirtualBox.

Электронные ресурсы

1. Электронная библиотека «Знаниум»: <https://znanium.com/>
2. Электронная библиотека «Юрайт»: <https://urait.ru/>
3. Научная электронная библиотека «Elibrary.ru»: <https://www.elibrary.ru/defaultx.asp>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

№	Наименование оборудованных учебных кабинетов, лабораторий	Перечень оборудования и технических средств обучения
1	Аудитория №402	<p>11 компьютеров</p> <p>Системный блок 1: Процессор Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz 8192 ОЗУ HDD Объем: 500 ГБ Монитор Benq G922HDA- 22 дюйма</p> <p>Системный блок 2: Процессор Intel(R) Core(TM) i5-4170 CPU @ 3.70GHz 4096 МБ ОЗУ; HDD Объем: 500 ГБ Монитор DELL 178FP</p> <p>Системный блок 3: Процессор Intel(R) Core(TM) i3-6100 CPU @ 3.70GHz 4096 МБ ОЗУ; SSD Объем: 120 ГБ Монитор Samsung 940NW Акустическая система 2.0 Интерактивная доска Smart Board Проектор Epson EH-TW535W</p>
2	Аудитория №403	Системный блок:

		Процессор Intel® Pentium®Dual-Core E2180 2048 ОЗУ; 320 HDD Монитор АОС 2470W Проектор Epson EH-TW5300 с акустической системой
3	Аудитория №405	Системный блок: Процессор Intel® Pentium®Dual-Core E2180 2048 ОЗУ; 320 HDD Монитор АОС 2470W Проектор Epson EH-TW5300 с акустической системой
4	Аудитория №302	11 компьютеров Системный блок: Процессор Intel(R) Core(TM) i3-2100 CPU @ 3.10GHz 4096 МБ ОЗУ; HDD Объем: 320 ГБ Монитор Acer P206HL - 20 дюймов Акустическая система Sven Интерактивная доска Smart Board Проектор Epson EH-TW535W
5	Аудитория №303	Системный блок: Процессор Intel® Pentium®Dual-Core E5200 2048 ОЗУ; 320 HDD Монитор Samsung SyncMaster 940NW Акустическая система Sven Проектор Nec M260W
6	Аудитория №305	Системный блок: Процессор Intel® Core™2 Duo E8500 2048 ОЗУ; 250 HDD Монитор Samsung SyncMaster 940NW Акустическая система Sven Проектор Nec M260W
7	Аудитория №306	12 компьютеров Системный блок: Процессор Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz 8192 ОЗУ; HDD Объем: 500 ГБ Монитор DELL EX231W - 24 дюйма Интерактивная доска Elite Panaboard UB-T880W с акустической системой Проектор Epson EB-440W
8	Аудитория №308	Системный блок: Процессор Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz; 8192 ОЗУ HDD Объем: 500 ГБ Монитор DELL EX231W - 24 дюйма Интерактивная доска Elite Panaboard UB-T880W с акустической системой Проектор Epson EB-440W
9	Аудитория №2-120	Системный блок: Процессор Intel® Core™2 Duo E8500 2048 ОЗУ\$ 250 HDD

		<p>Монитор Samsung SyncMaster 940NW Акустическая система Sven Проектор Nec M260W</p>
10	Аудитория №109	<p>11 компьютеров Системный блок: Процессор Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 4096 МБ ОЗУ SSD Объем: 120 ГБ Монитор Philips PHL 243V5 - 24 дюйма Акустическая система Sven Интерактивная доска Smart Board Проектор Epson EH-TW535W</p>
11	Аудитории № 309, 310, 311, 410, 411	<p>Проектор переносной Epson EB-5350 (1080p)– 1 шт. Экран переносной Digis 180x180 – 1 шт. Ноутбук HP ProBook 640 G3 (Intel Core i5 7200U, 4gb RAM, 250 SSD) – 1 шт.</p>

