

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН  
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
РЕСПУБЛИКИ ДАГЕСТАН «ТЕХНИЧЕСКИЙ КОЛЛЕДЖ ИМЕНИ Р.Н. АШУРАЛИЕВА»

## **ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

междисциплинарного курса МДК.04.01. Внедрение и поддержка  
компьютерных систем

Специальность: 09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Квалификация выпускника: специалист по работе с искусственным интеллектом

## ОГЛАВЛЕНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	3
1.1 Место междисциплинарного курса в структуре программы подготовки специалистов среднего звена.....	4
1.2 Цель и планируемые результаты освоения междисциплинарного курса .....	4
1.3 Условные обозначения.....	4
2. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ .....	6
2.1. Формы промежуточной аттестации по профессиональному модулю .....	8
3. КРИТЕРИИ ОЦЕНИВАНИЯ ЗНАНИЙ И УМЕНИЙ ОБУЧАЮЩИХСЯ, СФОРМИРОВАННОСТИ ОБЩИХ И ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ .....	9
3.1 Критерии оценки устного ответа обучающихся в 5-балльной системе .....	9
3.2 Общая классификация ошибок .....	9
3.3 Критерии оценивания выполнения лабораторных занятий обучающихся в 5-балльной системе.....	10
4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ ОБУЧАЮЩИХСЯ ПО МДК.04.01. ВНЕДРЕНИЕ И ПОДДЕРЖКА КОМПЬЮТЕРНЫХ СИСТЕМ .....	11
4.1 Перечень компетенций с указанием этапов их формирования и типовые контрольные тесты для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.....	11
5 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ .....	45

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Формирование фондов оценочных средств (далее - ФОС) - необходимое условие реализации основной профессиональной образовательной программы. Под **фондом оценочных средств** понимается комплект материалов, предназначенных для оценивания знаний, умений, сформированных общих и профессиональных компетенций на разных стадиях обучения. ФОС по междисциплинарному курсу МДК.04.01. Внедрение и поддержка компьютерных систем состоит из оценочных средств для текущего и рубежного контроля знаний и умений обучающихся и для проведения промежуточной аттестации в соответствии с учебным планом ГБПОУ РД «Технический колледж им. Р.Н. Ашуралиева».

Фонды оценочных средств для самоконтроля, текущего контроля, промежуточной аттестации по междисциплинарному курсу:

- комплект текущего контроля по теме (разделу) (комплекты тестов и задания, перечень контрольных вопросов для проверки готовности студентов к выполнению лабораторных занятий и оценки, приобретенных им в процессе выполнения работы знаний, умений, практических навыков и др.);
- комплект промежуточной аттестации (перечень вопросов для подготовки к дифференцированному зачету).

Фонд оценочных средств разработан на основе требований к результатам обучения и позволяют оценить знания, умения, сформированность общих и профессиональных компетенций, обучающихся на соответствие (или несоответствие) уровня их подготовки.

Формой промежуточной аттестации является дифференцированный зачет по МДК.04.01. Внедрение и поддержка компьютерных систем в 4 семестре.

К формам текущего контроля по учебной дисциплине относятся:

- устный опрос по теме;
- компьютерное тестирование;
- отчет по лабораторным и практическим заданиям.

Разработка оценочных материалов для включения в ФОС проводилась с учетом:

- форм проведения оценочных мероприятий (устный опрос, самостоятельная аудиторная работа, тестирование)
- уровней освоения учебного материала темы (ознакомительный, репродуктивный, продуктивный)
- видов деятельности, которые будут выполнять обучающиеся в процессе оценочных мероприятий (осознанное воспроизведение информации, применение информации, анализ, оценка)
- обучающих возможностей оценочных материалов
- возможности принятия решения об освоении обучающимися общих и профессиональных компетенций.

В состав ФОС включены материалы, выполняющие как контролирующие, так и обучающие функции. Они позволяют не только проверить уровень усвоения знаний, освоения умений, но и оценить различные качества личности обучающегося, уровень сформированности профессиональных и общих компетенций.

Чтобы обеспечить объективную оценку результатов контроля, разработаны критерии оценки показателей результатов обучения, эталоны выполнения заданий, «ключи» к тестам и т.п.

В материалы для оценочных мероприятий, проводимых в устной форме, включается перечень вопросов для подготовки обучающихся к оценочным мероприятиям. Тесты (в т.ч. для проведения компьютерного тестирования) формируются в соответствии с общими требованиями к оформлению и содержанию тестов.

## **1.1 Место междисциплинарного курса в структуре программы подготовки специалистов среднего звена**

ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем является обязательной частью профессионального цикла программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта.

## **1.2 Цель и планируемые результаты освоения междисциплинарного курса**

ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем направлен на формирование общих и профессиональных компетенций.

Освоение междисциплинарного курса должно способствовать формированию общих компетенций:

- ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
- ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;
- ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;
- ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;
- ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
- ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
- ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
- ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;
- ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

В результате освоения междисциплинарного курса обучающийся должен овладеть профессиональными компетенциями:

- ПК 4.1. Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем.
- ПК 4.3. Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика.

## **1.3 Условные обозначения**

В результате освоения дисциплины обучающийся должен

**уметь:**

<b>Код умения</b>	<b>Название умения</b>
У1	Подбирать и настраивать конфигурацию программного обеспечения компьютерных систем
У2	Проводить инсталляцию программного обеспечения компьютерных систем
У3	Производить настройку отдельных компонентов программного обеспечения компьютерных систем
У4	Определять направления модификации программного продукта
У5	Разрабатывать и настраивать программные модули программного продукта

**знать:**

<b>Код знания</b>	<b>Название знания</b>
З1	Основные методы и средства эффективного анализа функционирования программного обеспечения.
З2	Основные виды работ на этапе сопровождения ПО.

**Практический опыт**

<b>Код знания</b>	<b>Название знания</b>
П1	Выполнять инсталляцию, настройку и обслуживание программного обеспечения компьютерных систем.
П2	Настройка отдельных компонентов программного обеспечения компьютерных систем.
П3	Модифицировать отдельные компоненты программного обеспечения в соответствии с потребностями заказчика.
П4	Выполнение отдельных видов работ на этапе поддержки программного обеспечения компьютерных систем.

**Личностные результаты**

<b>Код знания</b>	<b>Название знания</b>
ЛР4	Проявляющий и демонстрирующий уважение к людям труда, осознающий ценность собственного труда. Стремящийся к формированию в сетевой среде лично и профессионального конструктивного «цифрового следа»
ЛР 10	Заботящийся о защите окружающей среды, собственной и чужой безопасности, в том числе цифровой
ЛР 13	Демонстрирующий умение эффективно взаимодействовать в команде, вести диалог, в том числе с использованием средств коммуникации
ЛР 14	Демонстрирующий навыки анализа и интерпретации информации из различных источников с учетом нормативно-правовых норм
ЛР 15	Демонстрирующий готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности

## 2. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

по профессиональному модулю ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем  
МДК.04.01. Внедрение и поддержка компьютерных систем

№	Результаты обучения (освоенные умения и знания)	ПК, ОК	Наименование темы	Наименование контрольно-оценочного средства	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	6	7
<b>МДК. 04.01 Внедрение и поддержка компьютерных систем</b>					
<b>Тема 1.1. Основные методы внедрения и анализа функционирования программного обеспечения</b>					
1.	31, 32	ОК 1-9 ПК 4.1, 4.3. ЛР4, ЛР10, ЛР13-15	ГОСТ Р ИСО/МЭК 12207. Основные процессы информационной системе согласно стандартам. Взаимосвязь между документами в информационной системе согласно стандартам.	Устный опрос/ вопросы по теме	Тестирование
2.	31, 32		Виды внедрения. План внедрения. Стратегии внедрения. Цели и сценарии внедрения.	Устный опрос/ вопросы по теме	
3.	31, 32		Функции менеджера сопровождения и менеджера развертывания	Устный опрос/ вопросы по теме	
4.	31, 32		Типовые функции инструментария для автоматизации процесса внедрения информационной системы	Устный опрос/ вопросы по теме	
5.	31, 32		Оценка качества функционирования информационной системы	Устный опрос/ вопросы по теме	
6.	31, 32		CALS-технологии	Устный опрос/ вопросы по теме	
7.	31, 32		Организация процесса обновления в информационной системе. Регламенты обновления	Устный опрос/ вопросы по теме	
8.	31, 32		Тестирование программного обеспечения в процессе внедрения и эксплуатации	Устный опрос/ вопросы по теме	
9.	31, 32		Эксплуатационная документация	Устный опрос/ вопросы по теме	
10.	У1-У6 П1-П4	ОК 1-9 ПК 4.1, 4.3. ЛР4, ЛР10, ЛР13-15	Измерение и анализ эксплуатационных характеристик качества программного обеспечения	Отчет по лабораторному занятию	Защита лабораторного занятия
11.	У1-У6 П1-П4		Выявление и документирование проблем установки программного обеспечения	Отчет по лабораторному занятию	

12.	У1-У6 П1-П4		Тестирование программного обеспечения	Отчет по по лабораторному занятию	
13.	У1-У6 П1-П4	ОК 1-9 ПК 4.1, 4.3. ЛР4, ЛР10, ЛР13-15	Использование инструментария для автоматизации процесса внедрения информационной системы	Отчет по по практическому занятию	
14.	У1-У6 П1-П4		Разработка сценария внедрения программного продукта для рабочего места	Отчет по по практическому занятию	
15.	У1-У6 П1-П4		Разработка руководства оператора	Отчет по по практическому занятию	
16.	У1-У6 П1-П4		Разработка (подготовка) документации для внедрения программных средств	Отчет по по практическому занятию	
17.	У1-У6 П1-П4		Разработка (подготовка) отчетных форм для внедрения программных средств	Отчет по по практическому занятию	
<b>Тема 1.2. Загрузка и установка программного обеспечения</b>					
18.	31, 32	ОК 1-9 ПК 4.1, 4.3. ЛР4, ЛР10, ЛР13-15	Понятие совместимости программного обеспечения. Аппаратная и программная совместимость. Совместимость драйверов. Причины возникновения проблем совместимости. Методы выявления проблем совместимости ПО	Устный опрос/ вопросы по теме	Тестирование
19.	31, 32		Выполнение чистой загрузки. Выявление причин возникновения проблем совместимости ПО. Выбор методов выявления совместимости	Устный опрос/ вопросы по теме	
20.	31, 32		Проблемы перехода на новые версии программ. Мастер совместимости программ. Инструментарий учета аппаратных компонентов	Устный опрос/ вопросы по теме	
21.	31, 32		Анализ приложений с проблемами совместимости. Использование динамически загружаемых библиотек. Механизм решения проблем совместимости на основе «системных заплаток». Разработка модулей обеспечения совместимости	Устный опрос/ вопросы по теме	
22.	31, 32		Создание в системе виртуальной машины для исполнения приложений	Устный опрос/ вопросы по теме	
23.	31, 32		Изменение настроек по умолчанию в образе. Подключение к сетевому ресурсу. Настройка обновлений программ. Обновление драйверов. Решение проблем конфигурации с помощью групповых политик	Устный опрос/ вопросы по теме	
24.	31, 32		Тестирование на совместимость в безопасном режиме. Восстановление системы	Устный опрос/ вопросы по теме	
25.	31, 32		Производительность ПК. Проблемы производительности. Анализ журналов событий. Настройка управления питанием. Оптимизация использования процессора	Устный опрос/ вопросы по теме	

26.	31, 32		Оптимизация использования памяти. Оптимизация использования жесткого диска. Оптимизация использования сети. Инструменты повышения производительности программного обеспечения. Средства диагностики оборудования. Разрешение проблем аппаратного сбоя	Устный опрос/ вопросы по теме	
27.	31, 32		Аппаратно-программные платформы серверов. Аппаратно-программные платформы рабочих станций. Установка серверной части.	Устный опрос/ вопросы по теме	
28.	31, 32		Особенности эксплуатации различных видов серверного программного обеспечения. Установка, адаптация и сопровождение клиентского программного обеспечения	Устный опрос/ вопросы по теме	
29.	У1-У6 П1-П4	ОК 1-9 ПК 4.1, 4.3. ЛР4, ЛР10, ЛР13-15	Выявление и документирование проблем установки программного обеспечения	Отчет по по лабораторному занятию	Защита лабораторного занятия
30.	У1-У6 П1-П4		Устранение проблем совместимости программного обеспечения	Отчет по по лабораторному занятию	
31.	У1-У6 П1-П4		Конфигурирование программных и аппаратных средств	Отчет по по лабораторному занятию	
32.	У1-У6 П1-П4		Настройки системы и обновлений	Отчет по по лабораторному занятию	
33.	У1-У6 П1-П4		Создание образа системы. Восстановление системы	Отчет по по лабораторному занятию	
34.	У1-У6 П1-П4		Разработка модулей программного средства	Отчет по по лабораторному занятию	
35.	У1-У6 П1-П4		Настройка сетевого доступа	Отчет по по лабораторному занятию	
36.	У1-У6 П1-П4		Выявление и документирование проблем установки программного обеспечения	Отчет по по лабораторному занятию	

### 2.1. Формы промежуточной аттестации по профессиональному модулю

МДК	Формы промежуточной аттестации
МДК. 04.01 Внедрение и поддержка компьютерных систем	4 семестр
	Дифференцированный зачет



### **3. КРИТЕРИИ ОЦЕНИВАНИЯ ЗНАНИЙ И УМЕНИЙ ОБУЧАЮЩИХСЯ, СФОРМИРОВАННОСТИ ОБЩИХ И ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ**

#### **3.1 Критерии оценки устного ответа обучающихся в 5-балльной системе**

При оценке устного ответа обучающегося учитывается:

- 1) полнота и правильность ответа;
- 2) степень осознанности, понимания изученного;
- 3) языковое оформление ответа.

**Отметка «5»:** ответ исчерпывающий, точный, полный и правильный на основании изученного материала; материал изложен в определенной логической последовательности, литературным языком; ответ самостоятельный.

**Отметка «4»:** ответ полный, обнаруживающий хорошее знание и понимание изученного материала; материал изложен в определенной логической последовательности, последовательно и грамотно, возможны отдельные затруднения в формулировке выводов.

**Отметка «3»:** ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный ответ, изложенный нелогично, ставится за ответ, в котором в основном правильно, но схематично или с отклонениями от последовательности изложения раскрыт материал.

**Отметка «2»:** при ответе обнаружено непонимание обучающимся основного содержания учебного материала, неумение его анализировать допущены существенные ошибки, которые обучающийся не смог исправить при наводящих вопросах преподавателя, отсутствует логика в изложении материала, нет необходимых обобщений и самостоятельной оценки фактов; недостаточно сформированы навыки устной речи.

#### **3.2 Общая классификация ошибок**

При оценке знаний, умений, навыков учитываются все ошибки (грубые и негрубые), а также недочеты в работе.

*Грубыми считаются ошибки:*

- незнание определения основных понятий, законов, общепринятых символов обозначений величин;
- неумение выделить в ответе главное; обобщить результаты изучения;
- неумение применить знания для решения задач, объяснения явления;
- неумение подготовить установку или лабораторное оборудование, использовать полученные данные для выводов;
- неумение пользоваться первоисточниками, учебником, справочником;
- нарушение техники безопасности, небрежное отношение к оборудованию

*Негрубыми считаются ошибки:*

- неточность формулировок, определений, понятий, законов, вызванная неполнотой охвата основных признаков определяемого понятия или заменой 1-3 из этих признаков второстепенными;
- ошибки при снятии показаний с измерительных приборов, не связанные с определением цены деления шкалы;
- ошибки, вызванные несоблюдением условий проведения лабораторных занятий;
- недостаточно продуманный план устного ответа (нарушение логики изложения, подмена отдельных основных вопросов второстепенными);
- нерациональные методы работы со справочной литературой;
- неумение выполнять лабораторные задания в общем виде.

### 3.3 Критерии оценивания выполнения лабораторных занятий обучающихся в 5-балльной системе

**Отметка «5»:** работа выполнена полностью и правильно; сделаны правильные выводы.

**Отметка «4»:** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.

**Отметка «3»:** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.

**Отметка «2»:** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.

### 3.4. Оценка тестов

Оценка в баллах	Степень выполнения задания
Неуд.	Выполнено не менее 40 % предложенных заданий
Удов.	Выполнено не менее 41-70 % предложенных заданий
Хор.	Выполнено не менее 71-95% предложенных заданий
Отл.	Выполнено не менее 96-100% предложенных заданий

#### 4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ ОБУЧАЮЩИХСЯ ПО МДК.04.01. ВНЕДРЕНИЕ И ПОДДЕРЖКА КОМПЬЮТЕРНЫХ СИСТЕМ

4.1 Перечень компетенций с указанием этапов их формирования и типовые контрольные тесты для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

##### 1. Тема 4.1. Основные методы внедрения и анализа функционирования программного обеспечения

1. Программная инженерия:
  - a. **software engineering**
  - b. Инструменты создания программного обеспечения
  - c. Коллектив инженеров-программистов, разрабатывающих программное обеспечение для компьютеров
  - d. **Дисциплина, изучающая применение строгого систематического количественного подхода к разработке, эксплуатации и сопровождению программного обеспечения**
  - e. Комплекс программ, предназначенный для решения инженерных задач, связанных с большим количеством расчетов
  - f. Инженерная индустрия применения прикладного программного обеспечения
  - g. **Совокупность инженерных методов и средств создания программного обеспечения**
  - h. Прикладное программное обеспечение для решения офисных задач
2. Построение SADT-модели включает в себя выполнение следующих действий:
  - a. Написание программного обеспечения для разрабатываемой системы по требованиям заказчика
  - b. **Сбор информации об объекте, определение его границ**
  - c. **Определение цели и точки зрения модели, построение, обобщение и декомпозиция диаграмм**
  - d. Представление исследуемой системы в графическом виде
  - e. Представление исследуемого объекта средствами системного моделирования
  - f. **Критическая оценка, рецензирование и комментирование**
  - g. Разработка, отладка и тестирование программного обеспечения
  - h. Использование графических пакетов для представления системы в виде модели
3. Моделирование основывается на принципах:
  - a. **Выбор модели оказывает определяющее влияние на подход к решению проблемы и то, как будет выглядеть это решение**
  - b. Декомпозиции системы на отдельные подзадачи
  - c. Инкапсуляции и полиморфизма
  - d. Децентрализации управления системой
  - e. **Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности**
  - f. Открытой трансформируемой системы
  - g. **Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, почти независимых друг от друга**
  - h. Анализа и синтеза проектирования систем
4. В бизнес-процессах выделяют классы процессов:
  - a. Решающие бизнес-процессы

- b. Регламентирующие бизнес-процессы
  - c. Основные бизнес-процессы**
  - d. Бизнес-процессы поведения системы
  - e. Программируемые бизнес-процессы
  - f. Экономические бизнес-процессы
  - g. Обеспечивающие бизнес-процессы**
  - h. Бизнес-процессы управления**
5. CASE-средства классифицируются по следующим признакам:
- a. По применяемым методологиям и моделям систем и БД**
  - b. По используемому программному обеспечению
  - c. По этапам жизненного цикла программного обеспечения
  - d. По степени интегрированности с СУБД**
  - e. По уровням детализации и декомпозиции проектируемой системы
  - f. По доступным платформам**
  - g. По используемым языкам программирования
  - h. По степени сложности моделируемой системы
6. К малым интегрированным средствам моделирования относятся:
- a. ARIS Toolset
  - b. Design/IDEF
  - c. ERwin**
  - d. BPwin**
  - e. Designer/2000
  - f. Paradigm Plus
  - g. Model Mart**
  - h. Rational Rose
7. К средним интегрированным средствам моделирования относятся:
- a. Rational Rose
  - b. Design/IDEF**
  - c. BPwin
  - d. Designer/2000**
  - e. ARIS Toolset**
  - f. Model Mart
  - g. Paradigm Plus
  - h. ERwin
8. Объектно-ориентированная методология (ООМ) включает в себя составные части:
- a. Объектно-ориентированный анализ**
  - b. Объектно-ориентированный подкласс
  - c. Объектно-ориентированное проектирование**
  - d. Объектно-ориентированная парадигма
  - e. Объектно-ориентированная экспозиция
  - f. Объектно-ориентированное моделирование
  - g. Объектно-ориентированное программирование**
  - h. Объектно-ориентированная декомпозиция
9. К основным понятиям объектно-ориентированного подхода относятся:
- a. Обобщение
  - b. Полиморфизм**
  - c. Инкапсуляция**
  - d. Реализация
  - e. Агрегирование
  - f. Наследование**

10. Главные принципы объектного подхода:
  - a. **Абстрагирование**
  - b. Наследование
  - c. **Ограничение доступа или инкапсуляция**
  - d. Безграничный доступ или инкапсуляция
  - e. **Модульность и иерархия**
  - f. Агрегирование
  - g. Композиция
  - h. Обобщение и специализация
11. Дополнительные принципы объектного подхода:
  - a. Реализация
  - b. **Типизация**
  - c. **Параллелизм**
  - d. Внедрение
  - e. Перпендикулярность
  - f. **Сохраняемость или устойчивость**
  - g. Несохранимость или неустойчивость
  - h. Динамичность
12. К инструментальным средствам объектно-ориентированного анализа и проектирования относятся:
  - a. **Rational Rose**
  - b. Model Mart
  - c. **MS Visio**
  - d. **ARIS**
  - e. IDEF1X
  - f. Erwin
  - g. BPwin
  - h. JAM
13. К инструментальным средствам представления функциональных моделей относятся:
  - a. JAM
  - b. **Model Mart**
  - c. MS Visio
  - d. ARIS
  - e. IDEF0
  - f. **Erwin**
  - g. **BPwin**
  - h. Rational Rose
14. Методологии, поддерживаемые в BPwin:
  - a. IDEF1X
  - b. **IDEF0**
  - c. IDEF1
  - d. **IDEF3**
  - e. IDEFX
  - f. IDEF5
  - g. **DFD**
15. Диаграмма IDEF0 может содержать следующие типы диаграмм:
  - a. Диаграмму классов
  - b. **Контекстную диаграмму, диаграмму декомпозиции**
  - c. Диаграмму компонентов
  - d. **Диаграмму дерева узлов**

- e. Диаграмму взаимодействий
  - f. Диаграмму только для экспозиции (FEO)**
  - g. Диаграмму последовательности, диаграмму кооперации
  - h. Диаграмму узлов
16. Уровни логической модели:
- a. Диаграмма сущность
  - b. Диаграмма связь
  - c. Диаграмма пакетов
  - d. Диаграмма сущность-связь**
  - e. Модель данных, основанная на классах
  - f. Модель данных, основанная на ключах**
  - g. Полная операционная модель
  - h. Полная атрибутивная модель**
17. Внутренние стрелки не входящие в состав диаграммы IDEF0:
- a. mechanism- output**output-input
  - b. mechanism- input**
  - c. utput-control
  - d. output-input feedback
  - e. output-control feedback
  - f. output-mechanism
  - g. control feedback- mechanism**
18. Типы стрелок не входящие в состав диаграммы IDEF0:
- a. Input
  - b. Editor**
  - c. Control
  - d. Properties**
  - e. Output
  - f. Mechanism
  - g. Call
  - h. Dictionary**
19. Quick Reports – создание простейших отчетов – позволяет создавать отчеты:
- a. Group/Totals. Табличный отчет с автоматической группировкой и сортировкой данных
  - b. Report Header. Печатается единожды в начале отчета
  - c. Columnar. Простой табличный отчет
  - d. Page Header. Печатается в верхней части каждой страницы
  - e. Vertical. Простой вертикальный отчет
  - f. Group Header. Печатается в начале каждой группы
  - g. Blank Report. Бланк. Создается пустой бланк отчета, в который не включаются данные
  - h. Detail. Печатается для каждой строчки набора данных
20. BРwin допускает следующие переходы с одной нотации на другую:
- a. IDEF3 → DFD
  - b. DFD → IDEF0
  - c. IDEF0 → DFD**
  - d. DFD → DFD
  - e. IDEF3 → IDEF0
  - f. IDEF0 → IDEF3**
  - g. IDEF3 → IDEF3

21. DFD описывает:
- a. Функции обработки стрелок (arrow)
  - b. Функции обработки информации (работы)**
  - c. Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации
  - d. Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации**
  - e. Функции обработки внешних ссылок
  - f. Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data stor+ E)**
  - g. Функции обработки документов
  - h. Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке внешних стрелок
22. ВРwin позволяет создавать на диаграмме DFD типы граничных стрелок:
- a. Обычная граничная стрелка**
  - b. Специальная стрелка
  - c. Внутренняя ссылка
  - d. Межстраничная ссылка и тоннельная стрелка**
  - e. Внешняя ссылка**
  - f. Страничная ссылка и тневая стрелка
  - g. Контрольная стрелка
  - h. Стрелка механизм
23. Создать отчет в ВРwin возможно с помощью:
- a. **Встроенных шаблонов**
  - b. Программных модулей, создаваемых разработчиком на языке Visual Basic
  - c. Создать отчет в ВРwin не возможно
  - d. Report Template Builder**
  - e. Отчет создается разработчиком
  - f. Отдельно поставляемых программ
  - g. Встроенных мастер-функций
  - h. RPTwin**
24. В ВРwin 4.0 отчеты могут быть экспортированы в распространенные форматы:
- a. Текстовый**
  - b. Символьный
  - c. MS Office**
  - d. Графический
  - e. HTML**
  - f. Internet Explorer
  - g. Acrobat
25. Поддерживаемые в RPTwin типы операторов:
- a. Текстовый оператор конкатенации (&)**
  - b. Символ
  - c. Текст
  - d. Дата
  - e. Арифметические**
  - f. Графический оператор конкатенации (&)
  - g. Логические**
  - h. Номер
26. Инструментальное средство ERwin позволяет:

- a. Редактировать и отлаживать программы
  - b. Проектировать на физическом и логическом уровне модели данных**
  - c. Управлять процессом конструирования ПО
  - d. Проектировать диаграммы вариантов использования и взаимодействий
  - e. Проводить процессы прямого и обратного проектирования баз данных**
  - f. Управлять процессом трансляции и отладки программ
  - g. Выравнивать модель и содержимое системного каталога после редактирования**
  - h. Проектировать контекстные диаграммы и диаграммы декомпозиции
27. ERwin позволяет создавать модели следующих типов:
- a. Модель, имеющую только логический уровень**
  - b. Модель, имеющую абстрактный уровень
  - c. Модель, имеющую абстрактный и физический уровни
  - d. Модель, имеющую только физический уровень**
  - e. Модель, имеющую абстрактный и логический уровни
  - f. Модель, имеющую как логический уровень, так и физический уровень**
  - g. Модель, имеющую концептуальный уровень
  - h. Модель, имеющую контекстный уровень
28. Для создания моделей ERwin используют международно признанные системы обозначений (нотации):
- a. IDEF0
  - b. IDEF1X**
  - c. IDEF3DFD
  - d. IE**
  - e. DM**
  - f. IDEFDFD
  - g. IDEF3
29. К основным компонентам диаграммы ERwin относятся:
- a. Сущности**
  - b. Переходы
  - c. Атрибуты**
  - d. Классы**
  - e. Слияния
  - f. Разветвления
  - g. Использования
  - h. Связи**
30. Точки зрения организации в ARIS:
- a. Структура внедрения и структура потоков
  - b. Организационная структура**
  - c. Управленческая структура
  - d. Поведенческая структура
  - e. Функциональная структура**
  - f. Коммуникационная структура
  - g. Структура данных и структура процессов**
  - h. Обобщенная структура
31. Уровни точки зрения в ARIS:
- a. Описание структуры
  - b. Описание требований**
  - c. Описание поведения
  - d. Описание разработки



- e. **Описание спецификации**
  - f. **Описание внедрения**
  - g. Описание процессов
  - h. Описание классов
32. Методы описания, используемые в ARIS:
- a. EPT – метод описания потоков
  - b. **EPC - метод описания процессов**
  - c. ERM - модель сущность-связь для описания структуры объектов
  - d. **ERM - модель сущность-связь для описания структуры данных**
  - e. EPP – метод описания пакетов
  - f. EPC – метод описания компонентов
  - g. **UML - унифицированный язык моделирования**
  - h. EPT – метод описания нитей
33. К основным компонентам инструментов ARIS Toolset относятся:
- a. Internet (интернет)
  - b. WordPad (ввод текстовых данных)
  - c. Media (средство для медиа описания моделей)
  - d. **Explorer (проводник)**
  - e. Acrobat (чтение текстовых данных)
  - f. **Designer (средство для графического описания моделей)**
  - g. Document (для ввода различных параметров и атрибутов) и выноски
  - h. **Таблица (для ввода различных параметров и атрибутов) и мастер (Wizards)**
34. ARIS Business Optimizer позволяет:
- a. **Определять целевые затраты и рассчитывать стоимость продукта: во что компании обходится предоставление отдельных продуктов**
  - b. Принимать решения о времени начала и окончания работы над проектом
  - c. **Принимать решения по аутсорсингу: стоит ли поручить выполнение бизнес-процессов внешнему поставщику услуг**
  - d. **Определять последовательность работ, выполняемых в ходе работы над проектом**
  - e. **Определять требования к персоналу компании, которая в дальнейшем будет эксплуатировать программное обеспечение**
  - f. **Рассчитывать заработную плату сотрудников компании после внедрения программного обеспечения**
  - g. **Планировать требования к обслуживающему персоналу, сопровождающему программное обеспечение**
35. «Взгляды» ARIS:
- a. **Процессы**
  - b. Потоки
  - c. **Функции (с целями)**
  - d. **Данные и организация**
  - e. Процедуры
  - f. Управление и внедрение
  - g. Нити
  - h. Память
36. Уровни анализа ARIS для каждого «взгляда»:
- a. Поведение
  - b. **Требования**
  - c. **Спецификации**
  - d. Функции
  - e. Процедуры

- f. Проверка
  - g. Внедрение**
  - h. Тестирование
37. MS Visio позволяет создавать схемы, чертежи, диаграммы с помощью:
- a. Встроенных шаблонов**
  - b. Панели инструментов
  - c. Трафаретов**
  - d. Графических редакторов
  - e. Дополнительного программного обеспечения
  - f. Панели рисования
  - g. Стандартных модулей**
  - h. Панели автофигур
38. Язык UML – это:
- a. Язык программирования высокого уровня
  - b. Унифицированный язык моделирования**
  - c. Язык для разработки систем искусственного интеллекта
  - d. Unified Modeling Language**
  - e. Язык управления базами данных
  - f. Язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем**
  - g. Язык создания запросов в базах данных
  - h. Язык программирования низкого уровня
39. Моделирование в UML позволяет решать задачи:
- a. Анализа и синтеза систем управления
  - b. Разработать и отладить программное обеспечение
  - c. Визуализировать систему в ее текущем или желательном для нас состоянии**
  - d. Провести тестирование разработанного программного обеспечения
  - e. Описать структуру или поведение системы; получить шаблон, позволяющий сконструировать систему**
  - f. Смоделировать разрабатываемую информационную систему
  - g. Документировать принимаемые решения, используя полученные модели**
  - h. Рассчитать экономическую эффективность от внедрения программного обеспечения
40. Словарь UML включает строительные блоки:
- a. Зависимости
  - b. Сущности**
  - c. Слияния
  - d. Разветвления
  - e. Связи**
  - f. Группировки
  - g. Диаграммы**
  - h. Декомпозиции
41. UML, как язык документирования, помимо исполняемого кода производит и другие продукты, включающие:
- a. Требования, архитектуру, проектные решения**
  - b. Спецификацию технических средств
  - c. Дизайн, исходный код, проектные планы,**
  - d. Требования к уровню квалификации разработчиков
  - e. Набор заданий для тестирования программного обеспечения
  - f. Требования к уровню квалификации персонала сопровождения
  - g. Тесты, прототипы, релизы (версии)**

42. UML включает синтаксические и семантические правила для:
- a. Агрегации
  - b. Тестирования
  - c. Имен, областей действия**
  - d. Сборки
  - e. Сопровождения
  - f. Видимости, целостности**
  - g. Вывода из эксплуатации
  - h. Исполнения**
43. Применение языка UML существенно упрощает последовательное использование механизмов:
- a. Спецификации, дополнения**
  - b. Принятые разделения**
  - c. Выработки требований
  - d. Создания плана работ
  - e. Механизмы расширения**
  - f. Тестирования программного обеспечения
  - g. Конструирования ПО
  - h. Сопровождения ПО
44. Механизмы расширения UML включают:
- a. Исключения
  - b. Стереотипы**
  - c. Дополнения
  - d. Управления
  - e. Помеченные значения**
  - f. Слияния
  - g. Ограничения**
  - h. Объединения
45. Язык UML предназначен для:
- a. Визуализации**
  - b. Тестирования
  - c. Сопровождения
  - d. Специфицирования**
  - e. Снятия с эксплуатации
  - f. Конструирования, документирования**
  - g. Анализа требований
  - h. Обучения персонала
46. В объектно-ориентированном моделировании между классами существуют типы связей:
- a. Слияние
  - b. Линейность
  - c. Зависимость**
  - d. Разветвление
  - e. Цикличность
  - f. Обобщение**
  - g. Ассоциация**
  - h. Агрегация
47. В состав графического представления класса в языке UML входят части:
- a. Отношения
  - b. Имя**
  - c. Связи

- d. **Атрибуты**
  - e. Описание
  - f. Сущности
  - g. **Операции**
  - h. Механизмы
48. Программное обеспечение делится на классы:
- a. Системное ПО и прикладное ПО
  - b. **Системное ПО, прикладное ПО и инструментальные средства разработки программ**
  - c. Операционные системы, прикладное ПО, утилиты и драйверы
  - d. Прикладное ПО и инструментальные средства разработки программ
  - e. Системное ПО и инструментальные средства разработки программ
  - f. **Системное ПО, прикладное ПО и системы программирования**
  - g. Операционные оболочки, операционные системы, офисные программы
  - h. **Системное ПО, прикладное ПО и инструментальное ПО**
49. Инструментальные средства разработки программ – это:
- a. **Средства создания новых программ**
  - b. Сервисные средства разработки ПО
  - c. Аналитические средства разработки ПО
  - d. **Программное обеспечение, предназначенное для разработки и отладки новых программ**
  - e. Средства отладки ПО
  - f. Средства тестирования ПО
  - g. **Аппаратные и программные инструменты разработки нового ПО**
  - h. Технические инструментальные средства разработки ПО
50. Аппаратные инструментальные средства разработки ПО – это:
- a. Система для разработки новых программ на конкретном языке программирования
  - b. Средства создания и редактирования текстов программ
  - c. **Микропроцессор и подключаемые (внешние) устройства**
  - d. **Устройства вычислительной системы, специально предназначенные для поддержки разработки ПО**
  - e. **Периферийные устройства, микропроцессор вычислительного комплекса, предназначенные для разработки нового ПО**
  - f. Программное обеспечение, написанное на языках программирования низкого уровня
  - g. Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ
  - h. Программы, используемые для корректировки и тестирования других прикладных или системных программ
51. Программные инструментальные средства разработки ПО – это:
- a. **Программы, позволяющие выполнить все работы, определенные методологией проектирования ПО**
  - b. Системное программное обеспечение, позволяющее сопровождать офисные программные пакеты
  - c. Средства создания текстовых документов
  - d. **Программное обеспечение, используемое на всех стадиях разработки нового ПО**
  - e. Программное обеспечение для настройки офисных приложений на условия конкретного применения
  - f. **Программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ**

- g. Устройство компьютера, специально предназначенное для поддержки разработки программных средств
  - h. Средства создания и редактирования текстовых документов
52. Транслятор – это:
- a. **Программа, выполняющая перевод программы с одного языка программирования на другой**
  - b. Комплекс программ мультимедийных технологий
  - c. **Программа, которая выполняет перевод программы с одного языка программирования на машинные коды**
  - d. Программа-переводчик с одного иностранного языка на другой
  - e. Техническое устройство передачи и преобразования аудио и видеосигналов
  - f. Техническое устройство для кодирования и декодирования информации
  - g. Программное обеспечение для обеспечения защиты информации на компьютере
  - h. **Одно из основных средств автоматизации программирования для преобразования программы, написанный на машинно-независимом языке, в программу на машинном языке конкретной ЭВМ**
53. Компилятор – это:
- a. **Один из видов трансляторов**
  - b. Прикладное программное обеспечение
  - c. Специальная утилита системного ПО
  - d. Операционная оболочка
  - e. **Переводит в коды сразу всю программу и создает независимый исполняемый файл**
  - f. Программное обеспечение, используемое в издательских системах
  - g. **Программа, которая переводит программу, написанную на языке программирования высокого уровня в программу на машинном языке не участвуя в ее исполнении**
  - h. Переводит в машинные коды 1 строчку программы и сразу ее выполняет
54. Интерпретатор:
- a. Программа для создания и редактирования электронных таблиц
  - b. **Программа, анализирующая команды или операторы исходной программы и немедленно выполняющая их**
  - c. Переводит в коды сразу всю программу и создает независимый исполняемый файл
  - d. **Переводит в машинные коды 1 строчку программы и сразу ее выполняет**
  - e. Программа для создания и редактирования текстовых документов
  - f. **Один из видов трансляторов**
  - g. Программа создания и управления базами данных
  - h. Программа создания файлов мультимедиа
55. Компоновщик – это:
- a. Программа для компоновки и оформления тестовых документов
  - b. **Редактор связей**
  - c. Комплекс программ, для создания и ведения баз данных
  - d. **Программа, которая из одного или нескольких объектных модулей с привлечением библиотечных программ и стандартных подпрограмм формирует загрузочный модуль**
  - e. Программное обеспечение для создания презентаций
  - f. **Программа сборки загрузочного модуля из полученных в результате отдельной компиляции объектных модулей с автоматическим поиском и присоединением библиотечных подпрограмм и процедур**
  - g. Программа для поиска синтаксических и семантических ошибок в программе

56. Отладчик:
- a. **Программа, облегчающая программисту выполнение отладки разрабатываемых им программ**
  - b. Программа для создания системы защиты файла
  - c. Программа создания системы защиты от вирусных атак
  - d. **Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку**
  - e. Операционная оболочка для создания и управления файловыми структурами
  - f. Системное программное обеспечение для настройки операционной системы
  - g. Программа создания и редактирования графических файлов
  - h. **Программа, позволяющая выполнять остановки в заданных точках, просмотреть текущие значения переменных и изменять их значения**
57. К этапам развития технологии разработки программного обеспечения относятся:
- a. **«Процедурное» программирование**
  - b. Программирование на алгоритмических языках высокого уровня
  - c. **Структурный подход к программированию**
  - d. Программирование на языках низкого уровня
  - e. **Компонентный подход и CASE-технологии**
  - f. Машинно-ориентированное программирование
  - g. Машинно-независимое программирование
58. «Стихийное» программирование:
- a. Разработка программного обеспечения без предварительного составления плана-графики работ
  - b. **Первый этап в истории развития технологии разработки программного обеспечения, когда программирование фактически было искусством**
  - c. **Период в истории разработки программного обеспечения, когда программа создавалась одним программистом, способным отслеживать последовательность выполняемых операций и местонахождения данных в программе**
  - d. Разработка программ с использованием различных языков программирования низкого и высокого уровня
  - e. Разработка программ с элементами случайного выбора алгоритмов решения задачи
  - f. **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**
  - g. Разработка программного обеспечения для решения задач теории вероятностей и математической статистики
  - h. Разработка программного обеспечения для решения задач, построенных на алгоритмах случайного поиска
59. Структурный подход к программированию – это:
- a. **Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения**
  - b. Создание программного обеспечения на основе структурной схемы решаемой задачи
  - c. Подход, требующий разработки структурной схемы алгоритма и программы решения задачи
  - d. **Подход, в основе которого лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм**
  - e. Подход к решению задачи, требующий создание структурной схемы этапов работ поразработке программного обеспечения

60. Объектный подход к программированию – это:
- a. Технология создания сложного программного обеспечения, основанная на представлении задачи исследования как объекта
  - b. Технология создания сложного программного обеспечения, предназначенного для автоматизации технологических объектов
  - c. **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**
  - d. Технология создания сложного программного обеспечения, основанная на представлении программы как единого объекта
  - e. **Технология создания сложного программного обеспечения, позволяющая вести практически независимую разработку отдельных частей (объектов) программы**
  - f. Технология создания сложного программного обеспечения, основанная на объектном представлении кода программы
61. Компонентный подход:
- a. **Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения**
  - b. **Предполагает взаимодействие между компонентами через стандартизованные двойные интерфейсы и позволяет использовать исполняемые файлы в любом языке программирования, поддерживающем соответствующую технологию**
  - c. Позволяет рассматривать объект исследования, как структуру, состоящую из отдельных компонент
  - d. способ написания исходного кода программного обеспечения
  - e. **Позволяет собрать объекты-компоненты в динамически вызываемые библиотеки или исполняемые файлы, и распространять в двоичном виде (без исходных текстов)**
  - f. Способ отладки и тестирования программного обеспечения
  - g. Способ внедрения и опытной эксплуатации программного обеспечения.
  - h. Метод выработки требований к разработке программного обеспечения
62. Управление требованиями:
- a. Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям
  - b. **Процесс систематического выявления, организации и документирования требований к сложной системе**
  - c. Выявление требований заказчика и управление ими
  - d. **Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности**
  - e. Процесс создания программного обеспечения и адаптация его под требования заказчика
  - f. Разработка требований к программному обеспечению и создание ПО на основе этих требований
  - g. **Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе**
  - h. Разработка программного обеспечения и выработка требований к изменению работы системы заказчика

63. Компонентный подход:
- a. **Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения**
  - b. **Предполагает взаимодействие между компонентами через стандартизованные двойные интерфейсы и позволяет использовать исполняемые файлы в любом языке программирования, поддерживающую соответствующую технологию**
  - c. Позволяет рассматривать объект исследования, как структуру, состоящую из отдельных компонент
  - d. способ написания исходного кода программного обеспечения
  - e. **Позволяет собрать объекты-компоненты в динамически вызываемые библиотеки или исполняемые файлы, и распространять в двоичном виде (без исходных текстов)**
  - f. Способ отладки и тестирования программного обеспечения
  - g. Способ внедрения и опытной эксплуатации программного обеспечения.
  - h. Метод выработки требований к разработке программного обеспечения
64. Управление требованиями:
- a. Задача выявления изначальных проблем заказчика и создание системы, удовлетворяющей этим требованиям
  - b. **Процесс систематического выявления, организации и документирования требований к сложной системе**
  - c. Выявление требований заказчика и управление ими
  - d. **Задача, состоящая в том, чтобы понимать проблемы заказчиков в их предметной области и на их языке и создавать системы, удовлетворяющие их потребности**
  - e. Процесс создания программного обеспечения и адаптация его под требования заказчика
  - f. Разработка требований к программному обеспечению и создание ПО на основе этих требований
  - g. **Процесс, в ходе которого вырабатывается и обеспечивается соглашение между заказчиком и выполняющей проект группой по поводу меняющихся требований к системе**
  - h. Разработка программного обеспечения и выработка требований к изменению работы системы заказчика
65. К методам выявления требований относятся:
- a. Беседы с первыми руководителями предприятия, для которого разрабатывается программное обеспечение
  - b. Анализ научной и технической литературы, посвященной вопросам разработки программного обеспечения
  - c. Личные встречи и беседы со всеми сотрудниками предприятия
  - d. Анализ технической документации и на основе нее разработка требований к системе
  - e. На начальном этапе требования не выявляются, а формируются по мере разработки программного обеспечения
  - f. **Интервьюирование и анкетирование, мозговой штурм и отбор идей**
  - g. **Совещания, посвященные требованиям, создание прототипов**
  - h. **Раскадровки, прецеденты, обыгрывание ролей**
66. Требования к разрабатываемой системе должны включать:



- a. Разработку программного обеспечения и выработка требований к изменению работы системы заказчика
  - b. 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)
  - c. Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
  - d. Описание выполняемых системой функций**
  - e. Технологию создания сложного программного обеспечения, основанную, а объектом представления кода программы
  - f. Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)**
  - g. Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
  - h. Технологию разработки программного обеспечения на базе структурной схемы развития языков программирования
67. Типы средств, иллюстрирующие цели моделирования системы:
- a. Функции, которые система должна выполнять**
  - b. Отношения между данными**
  - c. Зависящее от времени поведение системы (аспекты реального времени)**
  - d. Способы отладки и тестирования программного обеспечения
  - e. Создание программного обеспечения на основе структурной схемы исследуемого объекта или процесса
  - f. Выявление требований заказчика и управление ими
  - g. Технология разработки программного обеспечения на базе структурной схемы развития языков программирования
  - h. Построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
68. Преимущества объектно-ориентированного подхода:
- a. Быстрота написания программного кода
  - b. Статичность конфигурации системы
  - c. Возможность многократного использования**
  - d. Низкая стоимость проекта
  - e. Восприимчивость к изменениям**
  - f. Отсутствие необходимости документирования
  - g. Простота реализуемых моделей
  - h. Реалистичное моделирование**
69. Требования – это:
- a. Документ, регулирующий отношения между заказчиком информационной системы и проектировщиком
  - b. Некоторое свойство программного обеспечения, необходимое пользователю для решения проблемы при достижении поставленной цели**
  - c. Оформленное заказчиком в виде документа задание на проектирование программного обеспечения
  - d. Возможность, которую должна обеспечивать система**
  - e. Характеристика проектируемого программного обеспечения с точки зрения разработчика

- f. **Некоторое свойство программного обеспечения, которым должна обладать система или ее компонент, чтобы удовлетворить требования формальной документации**
  - g. Оформленное разработчиком в виде документа задание на проектирование программного обеспечения
  - h. Характеристика проектируемого программного обеспечения с точки зрения заказчика
70. Типичная схема процесса анализа С-требований включает в себя:
- a. **Идентификацию заказчика и проведение интервью с представителями заказчика**
  - b. Разработку программного обеспечения в соответствии с требованиями заказчика
  - c. Изложение заказчику требований к системе на основе разработанного программного обеспечения
  - d. **Написание С-Требований в форме стандартного документа**
  - e. Верификацию разработанного программного обеспечения в соответствии с требованиями заказчика
  - f. Составление плана мероприятий по анализу С-требований
  - g. **Проверку С-Требований и согласование их с заказчиком**
  - h. Адаптацию разработанного программного обеспечения в соответствии с требованиями заказчика
71. В классификацию требований к программной системе входят:
- a. Требования заказчика
  - b. Требования, накладываемые условиями эксплуатации
  - c. **Функциональные требования**
  - d. Требования, накладываемые аппаратными средствами
  - e. **Нефункциональные требования**
  - f. **Требования предметной области**
  - g. Экономические требования
  - h. Требования разработчиков
72. Процесс определения и анализа требований включает в себя:
- a. Анализ работы систем с аналогичной предметной областью
  - b. **Анализ предметной области, сбор и классификацию требований**
  - c. Проведение совместных совещаний с представителями заказчика
  - d. **Разрешение противоречий и определение приоритетов**
  - e. Адаптацию требований к разрабатываемому программному обеспечению
  - f. Декомпозицию общей задачи на подзадачи
  - g. **Проверку, специфицирование и документирование требований**
  - h. Верификацию требований в соответствии с разработанным программным обеспечением
73. Опорные точки зрения конечных пользователей системы программного обеспечения можно трактовать как:
- a. **Источник информации о системных данных**
  - b. Структуру требований
  - c. Источник событий
  - d. Структуру событий
  - e. **Структуру представлений**
  - f. Получателей требований
  - g. Источник сценариев
  - h. **Получателей системных сервисов**

74. При аттестации требований выполняются следующие типы проверок документации требований:
- a. Проверка на совместимость
  - b. Проверка на управляемость
  - c. Проверка правильности требований**
  - d. Проверка на непротиворечивость**
  - e. Проверка на соответствие
  - f. Проверка на обратимость
  - g. Проверка на полноту и на выполнимость**
  - h. Проверка на заменимость
75. К методам аттестации требований относится:
- a. Тестирование
  - b. Обзор требований**
  - c. Верификация
  - d. Сравнительный анализ
  - e. Прототипирование**
  - f. Генерация случайных данных
  - g. Генерация тестовых сценариев**
  - h. Декомпозиция
76. Уровни организационного управления при планировании разработки системы:
- a. Стратегический**
  - b. Tактический**
  - c. Оперативный**
  - d. Основной
  - e. Вспомогательный
  - f. Дополнительный
  - g. Системный
  - h. Аналитический
77. Для различных представлений проектируемой системы используют типы моделей:
- a. Статическая модель
  - b. Динамическая модель
  - c. Модель классов**
  - d. Модель декомпозиции
  - e. Модель размещения
  - f. Модель состояний**
  - g. Модель взаимодействия**
  - h. Модель агрегации
78. Классификация бизнес-процессов включает следующие классы процессов:
- a. Вспомогательные бизнес-процессы
  - b. Основные бизнес-процессы**
  - c. Дополнительные бизнес-процессы
  - d. Обеспечивающие бизнес-процессы**
  - e. Обслуживающие бизнес-процессы
  - f. Бизнес-процессы согласования
  - g. Бизнес-процессы управления**
  - h. Руководящие бизнес-процессы
79. Типы D-требований:
- a. Функциональные требования**
  - b. Интерфейсные требования

- c. **Нефункциональные требования**
  - d. Программные требования
  - e. **Обратные требования**
  - f. Ограниченные требования
  - g. Производительные требования
  - h. Надежность
80. Возможные способы организации D-требований:
- a. По атрибутам, по компонентам
  - b. По взаимоотношениям сущности
  - c. По пакетам и по иерархии компонентов
  - d. **По свойствам, по классам**
  - e. **По вариантам использования**
  - f. По узлам и по использованным процессам
  - g. **По состояниям и по иерархии функции**
  - h. По прецедентам, по кооперациям
81. К моделированию относится:
- a. **Система обозначений**
  - b. Система атрибутов
  - c. **Синтаксис языка моделирования**
  - d. Система свойств
  - e. Совокупность поведения объектов
  - f. **Совокупность графических объектов**
  - g. Семантика языка моделирования
  - h. Совокупность текстовых объектов
82. Классификация имитационных моделей:
- a. Статистическая
  - b. Адаптивная
  - c. **Статическая или динамическая**
  - d. Структурная
  - e. **Сетерминированная или стохастическая**
  - f. **Непрерывная или дискретная**
  - g. Объединенная
  - h. Декомпозиционная
83. Принципы разработки эффективного пользовательского интерфейса:
- a. Сложность, графика
  - b. **Структура, простота**
  - c. Связь, обработка
  - d. **Видимость, обратная связь**
  - e. Невидимость, сложность
  - f. **Толерантность, повторное использование**
  - g. Первое использование, итерация
  - h. Интеграция, повторение
84. Принципы разработки программного обеспечения:
- a. Коллективный процесс разработки
  - b. **Индивидуальный процесс разработки**
  - c. Параллельный процесс разработки
  - d. **Командный процесс разработки**

- e. Промежуточный процесс разработки
  - f. **Модель зрелости возможностей**
  - g. Модель законченности возможностей
  - h. Модель готовности процессов
85. Типы интерфейсных требований:
- a. **Пользовательские требования**
  - b. **Аппаратные требования**
  - c. Административные требования
  - d. Требования к производительности
  - e. **Программные и коммуникационные требования**
  - f. Требования к надежности
  - g. Требования к устойчивости
  - h. Атрибуты программной системы и другие требования
86. Технология проектирования определяется как совокупность составляющих:
- a. Поэтапная процедура
  - b. **Пошаговая процедура**
  - c. Модели и правила
  - d. **Критерий и правила**
  - e. Тестирование
  - f. **Нотаций**
  - g. Прецеденты
  - h. Классы
87. Разработка и сопровождение ИС в конкретной организации и конкретном проекте должна поддерживаться стандартами:
- a. Стандарт организации
  - b. Стандарт конкретного проекта
  - c. **Стандарт проектирования**
  - d. Стандарт оценки
  - e. **Стандарт оформления проектной документации**
  - f. Стандарт аудита
  - g. Стандарт оформления разработки
  - h. **Стандарт пользовательского интерфейса**
88. Результатами проектирования архитектуры являются:
- a. Модель административного интерфейса
  - b. **Модель процессов**
  - c. Модель потоков
  - d. Модель классов
  - e. **Модель данных**
  - f. **Модель пользовательского интерфейса**
  - g. Модель компонентов
  - h. Модель узлов
89. Какие работы включает процесс разработки программного обеспечения:
- a. Документирование, управление конфигурацией
  - b. Управление, создание инфраструктуры
  - c. Структура из процессов, работ, задач
  - d. Обеспечение качества, верификация
  - e. **Анализ требований, проектирование**

- f. **Программирование, сборка, тестирование**
  - g. **Ввод в действие, приемка**
  - h. Совместный анализ, аудит
90. Какие технологии разработки программ используются в современном программировании:
- a. **Визуальные**
  - b. **Событийные**
  - c. Структурные
  - d. **Объектно-ориентированные**
  - e. Модульные
  - f. Текстуальные
  - g. Графические
  - h. Машинно-ориентированное
91. Объектно-ориентированное проектирование использует инструментальные средства:
- a. Model mart
  - b. **Rational Rose**
  - c. Bpwin
  - d. **ARIS**
  - e. Idef1X
  - f. Erwin
  - g. **MS Visio**
  - h. Jam
92. Проектирование функциональных моделей поддерживается инструментальными средствами:
- a. Jam
  - b. **Model Mart**
  - c. MS visio
  - d. **ERwin**
  - e. Idef0
  - f. Aris
  - g. Rational rose
  - h. **BPwin**
93. IEEE – это:
- a. Коммерческая организация ученых и исследователей
  - b. Просто принятое обозначение, расшифровки не имеет
  - c. Обозначение всемирной компьютерной сети
  - d. **Всемирная некоммерческая техническая профессиональная ассоциация ученых и исследователей**
  - e. Такая аббревиатура нигде не используется
  - f. **Institute Of Electrical and Electronic Engineers, Inc**
  - g. Американская организация ученых-экономистов
  - h. **Институт инженеров радиоэлектроники и электротехники**
94. Ядро знаний SWEBOOK – это:
- a. ГОСТ на разработку программного обеспечения
  - b. **Нормативный документ, разработанный IEEE**
  - c. ГОСТ на разработку информационных систем

- d. Документ, устанавливающий правовые отношения между заказчиком и разработчиком программного обеспечения
  - e. **Основополагающий научно-технический документ, который отображает мнение специалистов в области программной инженерии**
  - f. Документ, устанавливающий методику тестирования и испытания программного обеспечения
  - g. **Документ, который согласуется с современными регламентированными процессами жизненного цикла ПО стандарта ISO/IEC 12207**
  - h. ГОСТ на разработку и комплектацию сопровождающей документации
95. Каждая область ядра знаний SWEBOOK представляется:
- a. Структурной схемой
  - b. **Общей схемой описания**
  - c. Диаграммой UML
  - d. Описанием и комментариями
  - e. **Определением понятийного аппарата, методов и средств инженерной деятельности**
  - f. Определением языка программирования
  - g. **Определением инструментов поддержки инженерной деятельности**
  - h. Иерархической диаграммой
96. К основным областям знаний SWEBOOK относятся:
- a. **Инженерия требований, проектирование ПО**
  - b. Анализ деятельности системы
  - c. Управление проектами
  - d. **Конструирование ПО**
  - e. Управление персоналом
  - f. **Тестирование ПО, сопровождение ПО**
  - g. Управление конфигурацией
  - h. Инженерия качества программных средств
97. К организационным областям знаний SWEBOOK относятся:
- a. Инженерия требований
  - b. **Управление конфигурацией, управление проектами**
  - c. Конструирование ПО
  - d. **Процесс инженерии программных средств, методы и средства программной инженерии**
  - e. Проектирование ПО
  - f. Сопровождение ПО
  - g. Тестирование ПО
  - h. **Инженерия качества программных средств**
98. В рамках Rational Unified Process (RUP) набор действий по разработке программ включает этапы:
- a. Создание структурных схем
  - b. Определения входных, выходных данных
  - c. Согласование стоимости проекта
  - d. Согласования требований с заказчиком
  - e. Создания бизнес-моделей
  - f. **Определение требований**
  - g. **Проектирование, программирование**
  - h. **Тестирование, внедрение**
99. Этапы разработки консалтинговых проектов включают в себя:

- a. **Анализ первичных требований и планирование работ**
  - b. Снятие программного продукта с эксплуатации
  - c. Декомпозицию задачи на подзадачи
  - d. Разработку спецификации и документации
  - e. **Проведение обследования деятельности предприятия**
  - f. Тестирование и сопровождение программного обеспечения
  - g. **Построение моделей деятельности предприятия (модели AS – IS – “как есть” и модели TO – BE – “как должно быть”)**
  - h. Разработку программного обеспечения
100. Концепции, лежащие в основе модульного программирования:
- a. Объем реализации и время исполнения (реакции)
  - b. Мера автоматизма в работе реализации и инструментах разработки
  - c. Визуальность и тестируемость разработки
  - d. **Функциональная декомпозиция, пространственная и временная группировка информации (модульность)**
  - e. **Упрощение связей**
  - f. **Комментируемость функций и данных**
  - g. Надежность, устойчивость
  - h. Безопасность

#### Тема 4.2. Загрузка и установка программного обеспечения

1. Какие программы можно отнести к системному программному обеспечению:
  - a. **операционные системы;**
  - b. прикладные программы;
  - c. игровые программы.
2. Какие программы можно отнести к системному ПО:
  - a. **драйверы;**
  - b. текстовые редакторы;
  - c. электронные таблицы;
  - d. графические редакторы.
3. Специфические особенности ПО как продукта:
  - a. **продажа по ценам ниже себестоимости (лицензирование);**
  - b. низкие материальные затраты при создании программ;
  - c. возможность создание программ небольшие коллективом или даже одним человеком;
  - d. разнообразие решаемых задач с помощью программных средств.
4. Какие программы можно отнести к системному ПО:
  - a. программа расчета заработной платы;
  - b. электронные таблицы;
  - c. **СУБД (системы управления базами данных).**
5. Какие программы нельзя отнести к системному ПО:
  - a. **игровые программы;**
  - b. компиляторы языков программирования;
  - c. операционные системы;
  - d. системы управления базами данных.
6. Какие программы можно отнести к прикладному программному обеспечению:
  - a. **электронные таблицы;**
  - b. таблицы решений;
  - c. СУБД (системы управления базами данных).
7. Какие программы можно отнести к прикладному ПО:
  - a. **программа расчета заработной платы;**
  - b. диспетчер программ;



- c. программа «Проводник» (Explorer).
- 8. Какие программы нельзя отнести к прикладному ПО:
  - a. **компиляторы и (или) интерпретаторы;**
  - b. текстовые и (или) графические редакторы;
  - c. электронные таблицы.
- 9. Можно ли отнести операционную систему к программному обеспечению:
  - a. **да;**
  - b. нет.
- 10. Можно ли отнести операционную систему к прикладному программному обеспечению:
  - a. да;
  - b. нет.**
- 11. Специфические особенности ПО как продукта:
  - a. **низкие затраты при дублировании;**
  - b. универсальность;
  - c. простота эксплуатации;
  - d. наличие поддержки (сопровождения) со стороны разработчика.
- 12. Какие программы можно отнести к системному ПО:
  - a. **утилиты;**
  - b. экономические программы;
  - c. статистические программы;
  - d. мультимедийные программы.
- 13. Этап, занимающий наибольшее время, в жизненном цикле программы:
  - a. **сопровождение;**
  - b. проектирование;
  - c. тестирование;
  - d. программирование;
  - e. формулировка требований.
- 14. Этап, занимающий наибольшее время, при разработке программы:
  - a. **тестирование;**
  - b. сопровождение;
  - c. проектирование;
  - d. программирование;
  - e. формулировка требований
- 15. Первый этап в жизненном цикле программы:
  - a. **формулирование требований;**
  - b. анализ требований;
  - c. проектирование;
  - d. автономное тестирование;
  - e. комплексное тестирование.
- 16. Один из необязательных этапов жизненного цикла программы:
  - a. **оптимизация;**
  - b. проектирование;
  - c. тестирование;
  - d. программирование;
  - e. анализ требований.
- 17. Самый большой этап в жизненном цикле программы:
  - a. **эксплуатация;**
  - b. изучение предметной области;
  - c. программирование;
  - d. тестирование;
  - e. корректировка ошибок.
- 18. Какой этап выполняется раньше:
  - a. **отладка;**
  - b. тестирование.
- 19. Какой этап выполняется раньше:
  - a. отладка;

- b. оптимизация;
  - c. **программирование;**
  - d. тестирование.
20. Что выполняется раньше:
- a. **компиляция;**
  - b. отладка;
  - c. компоновка;
  - d. тестирование.
21. Что выполняется раньше:
- a. **проектирование;**
  - b. программирование;
  - c. отладка;
  - d. тестирование.
22. В стадии разработки программы не входит:
- a. **автоматизация программирования;**
  - b. постановка задачи;
  - c. составление спецификаций;
  - d. эскизный проект;
  - e. тестирование.
23. Самый важный критерий качества программы:
- a. **работоспособность;**
  - b. надежность;
  - c. эффективность;
  - d. быстродействие;
  - e. простота эксплуатации.
24. Способы оценки качества:
- a. **сравнение с аналогами;**
  - b. наличие документации;
  - c. оптимизация программы;
  - d. структурирование алгоритма.
25. Существует ли связь между эффективностью и оптимизацией программы:
- a. **да;**
  - b. нет.
26. Наиболее важный критерий качества:
- a. **надежность;**
  - b. быстродействие;
  - c. удобство в эксплуатации;
  - d. удобный интерфейс;
  - e. эффективность.
27. Способы оценки надежности:
- a. **тестирование;**
  - b. сравнение с аналогами;
  - c. трассировка;
  - d. оптимизация.
28. Повышает ли качество программ оптимизация:
- a. **да;**
  - b. нет.
29. Существует ли связь между надежностью и быстродействием:
- a. **нет;**
  - b. да.
30. В каких единицах можно измерить надежность:
- a. **отказов/час;**
  - b. км/час;
  - c. Кбайт/сек;
  - d. операций/сек.

31. В каких единицах можно измерить быстродействие:
- отказов/час;
  - км/час;
  - Кбайт/сек;
  - операций/сек.**
32. Что относится к этапу программирования:
- написание кода программы;**
  - разработка интерфейса;
  - работоспособность;
  - анализ требований.
33. Последовательность этапов программирования:
- компилирование, компоновка, отладка;**
  - компоновка, отладка, компилирование;
  - отладка, компилирование, компоновка;
  - компилирование, отладка, компоновка.
34. Инструментальные средства программирования:
- компиляторы, интерпретаторы;**
  - СУБД (системы управления базами данных);
  - BIOS (базовая система ввода-вывода);
  - ОС (операционные системы).
35. На языке программирования составляется: Варианты ответа:
- исходный код;**
  - исполняемый код;
  - объектный код;
  - алгоритм.
35. Правила, которым должна следовать программа это:
- алгоритм;**
  - структура;
  - спецификация;
  - состав информации.
36. Можно ли внутри цикла поместить еще один цикл:
- да;**
  - нет.
37. Можно ли внутри условного оператора поместить еще одно условие:
- да;**
  - нет.
38. Можно ли одно большое (длинное) выражение разбить на несколько выражений:
- да;**
  - нет.
39. Если имеется стандартная функция, нужно ли писать собственную:
- нет;**
  - да.
40. Доступ, при котором записи файла читаются в физической последовательности, называется:
- последовательным;**
  - прямым;
  - простым;
  - основным.
41. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:
- прямым;**
  - последовательным;
  - простым;
  - основным.
42. Методы программирования (укажите НЕ верный ответ):
- логическое;**
  - структурное;

- c. модульное.
- 43. Что выполняется раньше:
  - a. **разработка алгоритма;**
  - b. выбор языка программирования;
  - c. написание исходного кода;
  - d. компиляция.
- 44. Можно ли переменным присваивать произвольные идентификаторы:
  - a. **да;**
  - b. нет.
- 45. Какие символы допускаются в именах переменных
  - a. **пробелы;**
  - b. цифры;
  - c. подчеркивание
- 46. Можно ли использовать имена, которые уже были использованы в другой программе (модуле):
  - a. **да;**
  - b. нет.
- 47. Можно ли ставить знак подчеркивания в начале имени:
  - a. **да, но не рекомендуется;**
  - b. да, без ограничений;
  - c. нет.
- 48. Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:
  - a. прямым указанием;
  - b. **венгерской нотацией;**
  - c. структурным программированием;
  - d. поляризацией.
- 49. Можно ли писать комментарии в отдельной строке:
  - a. **да;**
  - b. нет.
- 50. Транслируются ли комментарии:
  - a. да;
  - b. **нет.**
- 51. Наличие комментариев позволяет:
  - a. **быстрее найти ошибки в программе;**
  - b. быстрее писать программы;
  - c. быстрее выполнять программы.
- 52. Наличие комментариев позволяет:
  - a. легче разобраться в программе;
  - b. применять сложные структуры;
  - c. увеличить быстродействие.
- 53. Наличие комментариев позволяет:
  - a. **улучшить читаемость программы;**
  - b. улучшить эксплуатацию программы;
  - c. повысить надежность программы.
- 54. Что определяет выбор языка программирования:
  - a. **область приложения;**
  - b. знание языка;
  - c. наличие дополнительных библиотек.
- 55. Возможно ли комбинирование языков программирования в рамках одной задачи:
  - a. **да;**
  - b. нет.
- 56. Допустимо ли комбинирование языков программирования в рамках одной задачи:
  - a. **да;**
  - b. нет.

57. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:
- для экономических задач;**
  - для системных задач;
  - для инженерных задач.
58. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата:
- для инженерных задач;**
  - для системных задач;
  - для экономических задач.
59. На каком этапе производится выбор языка программирования:
- проектирование;**
  - программирование;
  - отладка;
  - тестирование.
60. Можно ли использовать комбинацию языков программирования в рамках одного проекта:
- да;**
  - нет.
61. Для решения экономических задач характерно применение:
- СУБД (систем управления базами данных);**
  - языков высокого уровня;
  - языков низкого уровня;
  - применение сложных математических расчетов.
62. Для решения инженерных задач характерно применение:
- САПР (систем автоматизированного проектирования);**
  - СУБД (систем управления базами данных);
  - ОС (операционных систем).
63. Причины синтаксических ошибок:
- плохое знание языка программирования;**
  - ошибки в исходных данных;
  - ошибки, допущенные на более ранних этапах;
  - неправильное применение процедуры тестирования.
64. Когда можно обнаружить синтаксические ошибки:
- при компиляции;**
  - при отладке;
  - при тестировании;
  - на этапе проектирования;
  - при эксплуатации.
65. Ошибки компоновки заключаются в том, что:
- указано внешнее имя, но не объявлено;**
  - неправильно использовано зарезервированное слово;
  - составлено неверное выражение;
  - указан неверный тип переменной.
66. Могут ли проявиться ошибки при изменении условий эксплуатации:
- да;**
  - нет.
67. Могут ли проявиться ошибки при изменении в предметной области: Варианты ответа:
- да;**
  - нет.
68. Возможно ли программирование с защитой от ошибок:
- да;**
  - нет.
69. Есть ли недостатки программирования с защитой от ошибок:
- да;**
  - нет.
70. Защитное программирование это:

- a. **встраивание в программу отладочных средств;**
  - b. создание задач, защищенных от копирования;
  - c. разделение доступа в программе;
  - d. использование паролей;
  - e. оформление авторских прав на программу.
71. Вид ошибки с неправильным написанием служебных слов (операторов):
- a. **синтаксическая;**
  - b. семантическая;
  - c. логическая;
  - d. символьная.
72. Вид ошибки с неправильным использованием служебных слов (операторов):
- a. **семантическая;**
  - b. синтаксическая;
  - c. логическая;
  - d. символьная.
73. Ошибки при написании программы бывают:
- a. **синтаксические;**
  - b. орфографические;
  - c. лексические;
  - d. фонетические;
  - e. морфологические.
74. Процедура поиска ошибки, когда известно, что она есть это:
- a. **отладка;**
  - b. тестирование;
  - c. компоновка;
  - d. транзакция;
  - e. трансляция.
75. Программа для просмотра значений переменных при выполнении программы:
- a. **отладчик;**
  - b. компилятор;
  - c. интерпретатор;
  - d. трассировка;
  - e. тестирование.
76. Отладка – это:
- a. **процедура поиска ошибок, когда известно, что ошибка есть;**
  - b. определение списка параметров;
  - c. правило вызова процедур (функций);
  - d. составление блок-схемы алгоритма.
77. Когда программист может проследить последовательность выполнения команд программы:
- a. **при трассировке;**
  - b. при тестировании;
  - c. при компиляции;
  - d. при выполнении программы;
  - e. при компоновке.
78. На каком этапе создания программы могут появиться синтаксические ошибки:
- a. **программирование;**
  - b. проектирование;
  - c. анализ требований;
  - d. тестирование.
79. Когда приступают к тестированию программы:
- a. **когда программа уже закончена;**
  - b. после постановки задачи;
  - c. на этапе программирования;
  - d. на этапе проектирования;
  - e. после составления спецификаций,

80. Тестирование бывает:
- автономное;**
  - инструментальное;
  - визуальное;
  - алгоритмическое.
81. Тестирование бывает:
- комплексное;**
  - инструментальное;
  - визуальное;
  - алгоритмическое.
82. Существует ли различие между отладкой и тестированием:
- да;**
  - нет.
83. При комплексном тестировании проверяются:
- согласованность работы отдельных частей программы;**
  - правильность работы отдельных частей программы;
  - быстродействие программы;
  - эффективность программы.
84. Чему нужно уделять больше времени, чтобы получить хорошую программу:
- тестированию;**
  - программированию;
  - отладке;
  - проектированию.
85. Процесс исполнения программы с целью обнаружения ошибок:
- тестирование;**
  - кодирование;
  - сопровождение;
  - проектирование.
86. Автономное тестирование это:
- тестирование отдельных частей программы;**
  - инструментальное средство отладки;
  - составление блок-схем;
  - пошаговая проверка выполнения программы.
87. Трассировка это:
- проверка пошагового выполнения программы;**
  - тестирование исходного кода;
  - отладка модуля;
  - составление блок-схемы алгоритма.
88. Локализация ошибки:
- определение места возникновения ошибки;**
  - определение причин ошибки;
  - обнаружение причин ошибки;
  - исправление ошибки.
89. Назначение тестирования:
- повышение надежности программы;**
  - обнаружение ошибок;
  - повышение эффективности программы;
  - улучшение эксплуатационных характеристик;
  - приведение программы к структурированному виду.
90. Назначение отладки:
- поиск причин существующих ошибок;**
  - поиск возможных ошибок;
  - составление спецификаций;
  - разработка алгоритма.
91. Инструментальные средства отладки (НЕ правильный ответ):
- компиляторы;**
  - отладчики;

- с. трассировка.
- 92. Отладка программ это:
  - а. **локализация и исправление ошибок;**
  - б. алгоритмизация программирования;
  - с. компиляция и компоновка.
- 93. Что выполняется раньше, автономная или комплексная отладка:
  - а. **автономная**
  - б. комплексная.
- 94. Что выполняется раньше, отладка или тестирование:
  - а. **отладка;**
  - б. тестирование.
- 95. Что такое автоматизация программирования:
  - а. **создание исходного кода программными средствами;**
  - б. создание исходного кода при помощи компилятора;
  - с. создание исходного кода без разработки алгоритма.
- 96. В чем сущность автоматизации программирования:
  - а. **создание программы без написания ее текста;**
  - б. получение готовой программы без выполнения компоновки;
  - с. в отсутствии компиляции.
- 97. Возможна ли автоматизация программирования:
  - а. **да;**
  - б. нет.
- 98. Создание исполняемого кода программы без написания исходного кода называется:
  - а. составлением спецификаций;
  - б. отладкой;
  - с. проектированием.
  - д. **автоматизацией программирования;**
- 99. Автоматизация программирования позволяет:
  - а. повысить надежность программы;
  - б. **сократить время разработки программы;**
  - с. повысить быстродействие программы.
- 100. Недостаток автоматизации программирования:
  - а. низкое быстродействие;
  - б. **большой размер программы;**
  - с. сложность программы.

### **Контрольные вопросы для проведения промежуточной аттестации**

1. Свойство, которое гарантирует, что информация не может быть доступна или раскрыта для неавторизованных личностей, объектов или процессов — это:  
*Ответ: конфиденциальность*
2. Удачная криптоатака называется:  
*Ответ: взломом*
3. Задачей анализа модели политики безопасности на основе анализа угроз системе является:  
*Ответ: минимизация вероятности преодоления системы защиты*
4. С доступом к информационным ресурсам внутри организации связан уровень ОС:  
*Ответ: сетевой*
5. Достоинством матричных моделей безопасности является:  
*Ответ: легкость представления широкого спектра правил обеспечения безопасности*
6. Конфигурация из нескольких компьютеров, выполняющих общее приложение, называется:  
*Ответ: кластером*



7. Защита от форматирования жесткого диска со стороны пользователей обеспечивается:  
*Ответ: аппаратным модулем, устанавливаемым на системную шину ПК*
8. Битовые протоколы передачи данных реализуются на ... уровне модели взаимодействия открытых систем.  
*Ответ: физическом*
9. Регистрацией в системе Windows 2000 управляет:  
*Ответ: процедура winlogon*
10. Обеспечение взаимодействия удаленных процессов реализуется на...уровне модели взаимодействия открытых систем.  
*Ответ: транспортном*
11. Проверка подлинности пользователя по предъявленному им идентификатору - это:  
*Ответ: аутентификация*
12. Соответствие средств безопасности решаемым задачам характеризует:  
*Ответ: эффективность*
13. Как предотвращение возможности отказа одним из участников коммуникаций от факта участия в передаче данных определяется:  
*Ответ: причастность*
14. Формирование пакетов, данных реализуется на уровне модели взаимодействия открытых систем.  
*Ответ: канальном*
15. Для реализации технологии RAID создается:  
*Ответ: псевдодрайвер*
16. Недостатком модели политики безопасности на основе анализа угроз системе является  
*Ответ: изначальное допущение вскрываемой системы*
17. Нормативный документ, регламентирующий все аспекты безопасности продукта информационных технологий, называется:  
*Ответ: профилем защиты*
18. Преднамеренные дефекты, внесенные в программные средства для целенаправленного скрытого воздействия на ИС, называются  
*Ответ: программными закладками*
19. Содержанием параметра угрозы безопасности информации "конфиденциальность" является:  
*Ответ: несанкционированное получение*
20. "Уполномоченные серверы" были созданы для решения проблемы  
*Ответ: имитации IP-адресов*
21. Запись определенных событий в журнал безопасности сервера называется:  
*Ответ: аудитом*
22. Достоинствами программной реализации криптографического закрытия данных являются:  
*Ответ: практичность и гибкость*
23. Процесс имитации хакером дружественного адреса называется:  
*Ответ: "спуфингом"*
24. Первым этапом разработки системы защиты ИС является:  
*Ответ: анализ потенциально возможных угроз информации*
25. Система, позволяющая разделить сеть на две или более частей и реализовать набор правил, определяющих условия прохождения пакетов из одной части в другую, называется:  
*Ответ: брандмауэром*
26. Недостатком дискретных моделей политики безопасности является:

- Ответ: статичность*
27. Достоинством модели конечных состояний политики безопасности является:  
*Ответ: высокая степень надежности*
28. Защита исполняемых файлов обеспечивается:  
*Ответ: обязательным контролем попытки запуска*
29. Маршрутизация и управление потоками данных реализуются на уровне модели взаимодействия открытых систем.  
*Ответ: сетевом*
30. Недостатком многоуровневых моделей безопасности является:  
*Ответ: невозможность учета индивидуальных особенностей субъекта*
31. Поддержка диалога между удаленными процессами реализуется на уровне модели взаимодействия открытых систем.  
*Ответ: сеансовом*
32. Основу политики безопасности составляет:  
*Ответ: способ управления доступом*
33. "Троянский конь" является разновидностью модели воздействия программных закладок  
*Ответ: искажение*
34. Администратором базы данных является:  
*Ответ: любой пользователь, создавший БД*
35. Предоставление легальным пользователем дифференцированных прав доступа к ресурсам системы — это:  
*Ответ: авторизация*
36. Математические методы нарушения конфиденциальности и аутентичности информации без знания ключей объединяет:  
*Ответ: криптоанализ*
37. Степень защищенности информации от негативного воздействия на неё с точки зрения нарушения её физической и логической целостности или несанкционированного использования — это:  
*Ответ: безопасность информации*
38. Процесс определения риска, применения средств защиты для сокращения риска с последующим определением приемлемости остаточного риска, называется:  
*Ответ: управлением риском*
39. Выделения пользователем и администраторам только тех прав доступа, которые им необходимы это:  
*Ответ: принцип минимализации привилегий*
40. Проверка подлинности субъекта по предъявленному им идентификатору для принятия решения о предоставлении ему доступа к ресурсам системы — это:  
*Ответ: аутентификация*
41. Получение и анализ информации о состоянии ресурсов системы с помощью специальных средств контроля называется  
*Ответ: мониторингом*
42. Наукой, изучающей математические методы защиты информации путем ее преобразования, является  
*Ответ: криптология*
43. Защита от программных закладок обеспечивается:  
*Ответ: аппаратным модулем, устанавливаемым на системную шину ПК*
44. Список объектов, к которым может быть получен доступ, вместе с доменом защиты объекта называется:  
*Ответ: перечнем возможностей*
45. Сетевой службой, предназначенной для централизованного решения задач

аутентификации и авторизации в крупных сетях, является

*Ответ: Kerberos*

46. Оконечное устройство канала связи, через которое процесс может передавать или получать данные, называется:

*Ответ: сокетом*

47. Достоинством модели политики безопасности на основе анализа угроз системе является

*Ответ: числовая вероятностная оценка надежности*

48. Присвоение субъектам и объектам доступа уникального номера, шифра и т.п. с целью получения доступа к информации — это:

*Ответ: идентификация*

49. Присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения, называется:

*Ответ: электронной подписью*

50. Обеспечением скрытности информации в информационных массивах занимается:

*Ответ: стеганография*

#### **Тема 4.1. Основные методы внедрения и анализа функционирования программного обеспечения**

- 1. Практическая работа № 1 «Разработка сценария внедрения программного продукта для рабочего места»** - Разработать сценарии внедрения программного продукта для рабочего места в соответствии с ГОСТ Р ИСО/МЭК 1207. Оформить план внедрения.
- 2. Практическая работа № 2. «Разработка руководства оператора»** - Разработать руководство оператора в соответствии со стандартом. Оформить руководство с учетом требований к технической документации.
- 3. Практическая работа № 3. «Разработка (подготовка) документации и отчетных форм для внедрения программных средств»** - Разработать (подготовить) документацию и отчетные формы для внедрения программных средств в соответствии со стандартом.

#### **Тема 4.2. Загрузка и установка программного обеспечения**

- 1. Практическая работа № 4. «Измерение и анализ эксплуатационных характеристик качества программного обеспечения»** - Разработать (подготовить) документацию отражающую анализ эксплуатационных характеристик качества программного обеспечения. Выявление причин возникновения проблем
- 2. Практическая работа № 5. «Выявление и документирование проблем установки программного обеспечения»** - Выявление причин возникновения проблем оформление сопровождающей документации, рекомендации по исправлению возникающих проблем.
- 3. Практическая работа № 6. «Устранение проблем совместимости программного обеспечения»** - Устранение проблем совместимости программного обеспечения, выявление причин проблем совместимости, рекомендации по исправлению проблем совместимости.
- 4. Практическая работа № 7. «Конфигурирование программных и аппаратных средств»** - Конфигурирование программных и аппаратных средств по заданному алгоритму.
- 5. Практическая работа № 8. «Настройки системы и обновлений»** - Настройка системы обновлений программного и аппаратного обеспечения с учетом требований заказчика

6. **Практическая работа № 9.** «Создание образа системы. Восстановление системы»- Создание образа системы. Восстановление системы, устранение проблем в работе системы. Создание рекомендаций по предотвращению сбоя системы.
7. **Практическая работа № 10.** «Разработка модулей программного средства» - Разработка модулей программного средства в соответствии с запросом заказчика, с учетом требований. Подготовка технической документации на разработку программного модуля
8. **Практическая работа № 11.** «Настройка сетевого доступа» - Настройка сетевого доступа пользователей к системе, разграничение доступа и обеспечение безопасности системы.

## 5 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

### Основные печатные издания

1. Федорова, Г.И. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие. – Москва: КУРС, 2021. – 336 с.

### Основные электронные издания

1. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. — Москва: КУРС: ИНФРА-М, 2021. — 336 с. — (Среднее профессиональное образование). - ISBN 978-5-906818-41-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1138896> (дата обращения: 25.12.2021). – Режим доступа: по подписке.

### Дополнительные источники

1. Гвоздева, В. А. Основы построения автоматизированных информационных систем: учебник / В. А. Гвоздева, И. Ю. Лаврентьева. — Москва: ФОРУ: ИНФРА-М, 2020. — 318 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0705-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1066509> (дата обращения: 13.05.2022). – Режим доступа: по подписке.
2. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. [http://real.tepkom.ru/Real\\_OM-СМ\\_A.asp](http://real.tepkom.ru/Real_OM-СМ_A.asp)

### Интернет-источники:

1. <http://www.ict.edu.ru> - Федеральный образовательный портал
2. <http://www.edu-it.ru> - ИТ-образование в России: сайт открытого е-консорциума