

МИНОБРНАУКИ РОССИИ
Глазовский инженерно-экономический институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Ижевский государственный технический университет имени М.Т.Калашникова»
(ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

03.01. «Внедрение и поддержка компьютерных систем»

09.02.07 Информационные системы и программирование

Фонд оценочных средств разработан на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.07 "Информационные системы и программирование", утвержденного приказом Министерства образования и науки Российской Федерации 09 декабря 2016 г. № 1547 с изменениями и дополнениями (приказ Министерства просвещения Российской Федерации от 17.12.2020 № 747 «О внесении изменений в федеральные государственные образовательные стандарты среднего профессионального образования» (зарегистрирован 22.01.2021 № 62178), приказ Министерства просвещения Российской Федерации от 01.09.2022 № 796 «О внесении изменений в федеральные государственные образовательные стандарты среднего профессионального образования» (зарегистрирован 11.10.2022 № 70461))

**Организация
разработчик:**

ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Разработчик:

Савельева Татьяна Александровна, преподаватель СПО

Утверждено:

Протокол Ученого совета филиала № 3, от 20 мая 2025 г.

Руководитель образовательной программы



Т.А. Савельева

23 мая 2025 г.

Согласовано:

Начальник отдела по учебно-методической работе



И.Ф. Яковлева

23 мая 2025 г.

Оглавление

Паспорт фонда оценочных средств	4
Контрольно-измерительные материалы.....	6
Рисунок 1 - Основные понятия проектного менеджмента и их взаимосвязь.....	57

Паспорт фонда оценочных средств
по дисциплине **МДК. 03.01. «Внедрение и поддержка компьютерных систем»**
(наименование дисциплины)

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Тема 1.1. Основные методы внедрения и анализа функционирования программного обеспечения	ПК 4.1, ПК 4.3	тестовые задания, вопросы к зачету
2	Тема 1.2. Загрузка и установка программного обеспечения	ПК 4.1, ПК 4.3	тестовые задания, вопросы к зачету

Код	Наименование результата обучения
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
ОК 1	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес
ОК 2	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество
ОК 3	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность
ОК 4	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития
ОК 5	Использовать информационно-коммуникационные технологии в профессиональной деятельности
ОК 6	Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями
ОК 7	Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий
ОК 8	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации
ОК 9	Ориентироваться в условиях частой смены технологий в профессиональной деятельности

Уровни сформированности профессиональных компетенций

ПК 4.1. Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем	
Категории	Описание показателей
Уметь	- применять средства информатизации информационных технологий для реализации профессиональных задач; профессиональной деятельности, инструкции на государственном и иностранном языке; современную научную профессиональную

	терминологию;
	- выполнять инсталляцию, настройку и обслуживание программного обеспечения компьютерных систем; настройку отдельных компонентов программного обеспечения компьютерных систем. ;
	- использовать актуальную нормативно-правовую документацию по специальности. ;
Знать	- основные методы и средства эффективного анализа функционирования программного обеспечения;
	- основные виды работ на этапе сопровождения ПО.
	- методы организации работы в команде
ПК 4.3. Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика	
Уметь	Разрабатывать программы модуля программного продукта.
Знать	основные методы и средства эффективного анализа функционирования программного обеспечения.
	- модифицировать отдельные компоненты программного обеспечения соответствии с потребностями заказчика; отдельные виды работ на этапе поддержки программного обеспечения компьютерных систем.
	Использует выполнение отдельных видов работ на этапе поддержки программного обеспечения компьютерных систем

Зачетно-экзаменационные материалы

Контрольные вопросы для оценки усвоения знаний

Промежуточная аттестация проводится в виде дифференцированного зачета. Вопросы к зачету:

1. Какие из перечисленных условий входят в состав типичных факторов успешности проекта внедрения ИС?
2. Назначение и состав методологий внедрения.
3. Основные этапы внедрения информационных систем.
4. Стратегия, цель и задачи внедрения информационных систем.
5. Жизненный цикл и технология внедрения информационных систем.
6. Типовые функции инструментария для автоматизации процесса внедрения информационной системы.
7. Методы обеспечения качества функционирования информационных систем.
8. Входит ли "Управление снабжением" в перечень областей знаний управления проектами РМВОК?
9. Стратегии внедрения и использования ИС.
10. Планирование работ, ресурсов и контроль выполнения плана внедрения.
11. Порядок участие высшего руководства во внедрении системы.

12. Проведение работ по внедрению ИС специалистами по интегрированию систем совместно со специалистами предприятия.
13. Мониторинг качества выполняемых работ при внедрении ИС.
14. Оценка качества внедренных модулей ИС.
15. Организационное управление внедрением ИС;
16. Организационно-административное обеспечение внедрением ИС.
17. управление бизнес-процессами внедрения ИС;
18. Какие модели и каким образом используются при внедрении информационных систем?
19. Какие программные средства используются для моделирования процессов при разработке и внедрении информационных систем?
20. На основании каких данных и информации разрабатываются модели состояния AS IS и AS TO BE?
21. Порядок разработки, внедрения и развития ИС? Кто участвует в подготовке технического задания на разработку ИС?
22. Этапы проектирования и внедрения информационных систем.
23. Этапы жизненного цикла информационной системы.
24. На каком этапе разработки и внедрения ИС производится обучение персонала компании?
25. Перечислите основные фазы внедрения ИС.

Контрольно-измерительные материалы

Формы текущего контроля успеваемости:

Опрос - это основной вид устной проверки, может использоваться как фронтальный (на вопросы преподавателя по сравнительно небольшому материалу краткие ответы (как правило, с места) дают многие обучающиеся), так и индивидуальный (проверка знаний отдельных обучающихся). Комбинированный опрос - одновременный вызов для ответа сразу нескольких обучающихся, из которых один отвечает устно, один-два готовятся к ответу, выполняя на доске различные записи, а остальные выполняют за отдельными столами индивидуальные письменные или практические задания преподавателя.

Тестирование – задания, с вариантами ответов.

Критерии оценивания

Оценки «отлично» заслуживает студент, если он ответил правильно на 90% вопросов теста

Оценки «хорошо» заслуживает студент, если он ответил правильно на часть вопросов 75%-90%;

Оценки «удовлетворительно» заслуживает студент, если он правильно ответил часть вопросов 50%-75%;

Оценки «неудовлетворительно» заслуживает студент, если он правильно ответил менее чем на 50% вопросов.

Контрольная работа - письменная работа по теме. Состоит из нескольких задач различной степени сложности.

Критерии оценивания

Оценки «отлично» заслуживает студент, обнаруживший глубокое знание материала, умение свободно выполнять задания, понимающий взаимосвязь основных понятий темы;

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала; успешно выполняющий предусмотренные задания; и допустивший незначительные ошибки: неточность фактов, стилистические ошибки;

Оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного материала в объеме, необходимом для дальнейшего изучения дисциплины. Справляющийся с выполнением заданий; допустивший погрешности в ответе, но обладающий необходимыми знаниями для их устранения под руководством преподавателя;

Оценки «неудовлетворительно» заслуживает студент, обнаруживший существенные пробелы в знании основного материала; не справляющийся с выполнением заданий, допустивший серьезные погрешности в ответах, нуждающийся в повторении основных разделов курса под руководством преподавателя.

Практическая/лабораторная работа – работа по теме. Состоит из нескольких задач различной степени сложности.

Критерии оценивания

Оценки «отлично» заслуживает студент, обнаруживший глубокое знание материала, умение свободно выполнять задания, понимающий взаимосвязь основных понятий темы;

Оценки «хорошо» заслуживает студент, обнаруживший полное знание материала; успешно выполняющий предусмотренные задания; и допустивший незначительные ошибки: неточность фактов, стилистические ошибки;

Оценки «удовлетворительно» заслуживает студент, обнаруживший знания основного материала в объеме, необходимом для дальнейшего изучения дисциплины. Справляющийся с выполнением заданий; допустивший погрешности в ответе, но обладающий необходимыми знаниями для их устранения под руководством преподавателя;

Оценки «неудовлетворительно» заслуживает студент, обнаруживший существенные пробелы в знании основного материала; не справляющийся с выполнением заданий, допустивший серьезные погрешности в ответах, нуждающийся в повторении основных разделов курса под руководством преподавателя.

Формы текущего контроля

Контрольные вопросы для проведения тестирования

Тема 4.1. Основные методы внедрения и анализа функционирования программного

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 BPwin 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 теневая стрелка
- Контрольная стрелка
- Стрелка механизм

23 Создать отчет в BPwin возможно с помощью:

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

24 В BPwin 4 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, написанный 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 объектноориентированном программировании

61 Компонентный подход:

- + Предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения
- + Предполагает взаимодействие между компонентами через стандартизованные 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 на 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 радиозлектроники и электротехники

92 Ядро знаний SWEBOK – 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 SWEBOK 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 SWEBOK 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 SWEBOK 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
- + Объем реализации и времени исполнения (реакции), надежности, устойчивости, безопасности
- + Меры автоматизма в работе реализации и инструментах разработки
- + Визуальности и тестируемости разработки

Тема 4.2. Загрузка и установка программного обеспечения

Контрольные вопросы для проведения тестирования

1 Какие программы можно отнести к системному программному обеспечению:

Варианты ответа:

- +операционные системы;**
- прикладные программы;
- игровые программы.

2 Какие программы можно отнести к системному ПО:

Варианты ответа:

+драйверы;

текстовые редакторы;

электронные таблицы;

графические редакторы.

3 Специфические особенности ПО как продукта:

+продажа по ценам ниже себестоимости (лицензирование);

низкие материальные затраты при создании программ;

возможность создание программ небольшими коллективом или даже одним человеком;

разнообразие решаемых задач с помощью программных средств.

4 Какие программы можно отнести к системному ПО:

Варианты ответа:

программа расчета заработной платы;

электронные таблицы;

+СУБД (системы управления базами данных).

5 Какие программы нельзя отнести к системному ПО:

Варианты ответа:

+игровые программы;

компиляторы языков программирования;

операционные системы;

системы управления базами данных.

6 Какие программы можно отнести к прикладному программному обеспечению:

Варианты ответа:

+электронные таблицы;

таблицы решений;

СУБД (системы управления базами данных).

7 Какие программы можно отнести к прикладному ПО:

Варианты ответа:

+программа расчета заработной платы;

диспетчер программ;

программа «Проводник» (Explorer).

8 Какие программы нельзя отнести к прикладному ПО:

Варианты ответа:

+компиляторы и (или) интерпретаторы;

текстовые и (или) графические редакторы;

электронные таблицы.

9 Можно ли отнести операционную систему к программному обеспечению:

Варианты ответа:

+да;

нет.

10 Можно ли отнести операционную систему к прикладному программному обеспечению:

Варианты ответа:

да;

+нет.

11 Специфические особенности ПО как продукта:

Варианты ответа:

+низкие затраты при дублировании;

универсальность;

простота эксплуатации;

наличие поддержки (сопровождения) со стороны разработчика.

12 Какие программы можно отнести к системному ПО:

Варианты ответа:

+утилиты;

экономические программы;

статистические программы;

мультимедийные программы.

13 Этап, занимающий наибольшее время, в жизненном цикле программы:

Варианты ответа:

+сопровождение;

проектирование;

тестирование;

программирование;

формулировка требований.

14 Этап, занимающий наибольшее время, при разработке программы:

Варианты ответа:

+тестирование;

сопровождение;

проектирование;

программирование;

формулировка требований.

15 Первый этап в жизненном цикле программы:

Варианты ответа:

+формулирование требований;

анализ требований;

проектирование;

автономное тестирование;

комплексное тестирование.

16 Один из необязательных этапов жизненного цикла программы:

Варианты ответа:

+оптимизация;

проектирование;

тестирование;

программирование;

анализ требований.

17 Самый большой этап в жизненном цикле программы:

Варианты ответа:

+эксплуатация;

изучение предметной области;

программирование;

тестирование;

корректировка ошибок.

18 Какой этап выполняется раньше:

Варианты ответа:

+отладка;

тестирование.

19 Какой этап выполняется раньше:

Варианты ответа:

отладка;

оптимизация;

+программирование;

тестирование.

20 Что выполняется раньше:

Варианты ответа:

+компиляция;

отладка;

компоновка;

тестирование.

21 Что выполняется раньше:

Варианты ответа:

+проектирование;

программирование;

отладка;

тестирование.

22 В стадии разработки программы не входит:

Варианты ответа:

+автоматизация программирования;

постановка задачи;

составление спецификаций;

эскизный проект;

тестирование.

23 Самый важный критерий качества программы:

Варианты ответа:

+работоспособность;

надежность;

эффективность;

быстродействие;

простота эксплуатации.

24 Способы оценки качества:

Варианты ответа:

+сравнение с аналогами;

наличие документации;

оптимизация программы;

структурирование алгоритма.

25 Существует ли связь между эффективностью и оптимизацией программы:

Варианты ответа:

+да;

нет.

26 Наиболее важный критерий качества:

Варианты ответа:

+надежность;

быстродействие;

удобство в эксплуатации;

удобный интерфейс;

эффективность.

27 Способы оценки надежности:

Варианты ответа:

+тестирование;

сравнение с аналогами;

трассировка;

оптимизация.

28 Повышает ли качество программ оптимизация:

Варианты ответа:

+да;

нет.

29 Существует ли связь между надежностью и быстродействием:

Варианты ответа:

+нет:

да.

30 В каких единицах можно измерить надежность:

Варианты ответа:

+отказов/час;

км/час;

Кбайт/сек;

операций/сек.

31 В каких единицах можно измерить быстродействие:

Варианты ответа:

отказов/час;

км/час;

Кбайт/сек;

+операций/сек.

32 Что относится к этапу программирования:

Варианты ответа:

+написание кода программы;

разработка интерфейса;

работоспособность;

анализ требований.

33 Последовательность этапов программирования:

Варианты ответа:

+компилирование, компоновка, отладка;

В) компоновка, отладка, компилирование;

отладка, компилирование, компоновка;

компилирование, отладка, компоновка.

34) Инструментальные средства программирования:

Варианты ответа:

+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.

36) Правила, которым должна следовать программа это:

Варианты ответа:

+алгоритм;

структура;

спецификация;

состав информации.

37) Можно ли внутри цикла поместить еще один цикл:

Варианты ответа:

+да;

нет.

38) Можно ли внутри условного оператора поместить еще одно условие:

Варианты ответа:

+да;

нет.

39) Можно ли одно большое (длинное) выражение разбить на несколько выражений:

Варианты ответа:

+да;

нет.

40) Если имеется стандартная функция, нужно ли писать собственную:

Варианты ответа:

+нет;

да.

41) Доступ, при котором записи файла читаются в физической последовательности, называется:

Варианты ответа:

+последовательным;

прямым;

простым;

основным.

42) Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:

Варианты ответа:

+прямым;
последовательным;
простым;
основным.

43 Методы программирования (укажите НЕ верный ответ):

Варианты ответа:

+логическое;
структурное;
модульное.

44 Что выполняется раньше:

Варианты ответа:

+разработка алгоритма;
выбор языка программирования;
написание исходного кода;
компиляция.

45 Можно ли переменным присваивать произвольные идентификаторы:

Варианты ответа:

+да;
нет.

46 Найдите НЕ правильное условие для создания имен:

Варианты ответа:

+имена могут содержать пробелы;
длинное имя можно сократить;
из имени лучше выбрасывать гласные;
можно использовать большие буквы.

47 Какие символы не допускаются в именах переменных:

Варианты ответа:

+пробелы;
цифры;
подчеркивание

48 Можно ли использовать имена, которые уже были использованы в другой программе (модуле):

Варианты ответа:

+да;
нет.

49 Можно ли ставить знак подчеркивания в начале имени:

Варианты ответа:

+да, но не рекомендуется;
да, без ограничений;
нет.

50 Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:

Варианты ответа:

прямым указанием;

+венгерской нотацией;
структурным программированием;
поляризацией.

51 Можно ли писать комментарии в отдельной строке:

Варианты ответа:

+да;
нет.

52 Транслируются ли комментарии:

Варианты ответа:

да;
+нет.

53 Наличие комментариев позволяет:

Варианты ответа:

+быстрее найти ошибки в программе;
быстрее писать программы;
быстрее выполнять программы.

54 Наличие комментариев позволяет:

Варианты ответа:

+легче разобраться в программе;
применять сложные структуры;
увеличить быстродействие.

55 Наличие комментариев позволяет:

Варианты ответа:

+улучшить читабельность программы;
улучшить эксплуатацию программы;
повысить надежность программы.

56 Что определяет выбор языка программирования:

Варианты ответа:

+область приложения;
знание языка;
наличие дополнительных библиотек.

57 Возможно ли комбинирование языков программирования в рамках одной задачи:

Варианты ответа:

+да;
нет.

58 Допустимо ли комбинирование языков программирования в рамках одной задачи :

Варианты ответа:

+да;
нет.

59 Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:

Варианты ответа:

+для экономических задач;
для системных задач;

для инженерных задач.

60 Для каких задач характерен большой объем вычислений, использование сложного математического

Варианты ответа:

+для инженерных задач;
для системных задач;
для экономических задач.

61 На каком этапе производится выбор языка программирования:

Варианты ответа:

+проектирование;
программирование;
отладка;
тестирование.

62 Можно ли использовать комбинацию языков программирования в рамках одного проекта:

Варианты ответа:

+да;
нет.

63 Для решения экономических задач характерно применение:

Варианты ответа:

+СУБД (систем управления базами данных);
языков высокого уровня;
языков низкого уровня;
применение сложных математических расчетов.

64 Для решения инженерных задач характерно применение:

Варианты ответа:

+САПР (систем автоматизированного проектирования);
СУБД (систем управления базами данных);
ОС (операционных систем).

65 Причины синтаксических ошибок:

Варианты ответа:

+плохое знание языка программирования;
ошибки в исходных данных;
ошибки, допущенные на более ранних этапах;
неправильное применение процедуры тестирования.

66 Когда можно обнаружить синтаксические ошибки:

Варианты ответа:

+при компиляции;
при отладке;
при тестировании;
на этапе проектирования;
при эксплуатации.

67 Ошибки компоновки заключаются в том, что:

Варианты ответа:

+указано внешнее имя, но не объявлено;

неправильно использовано зарезервированное слово;
составлено неверное выражение;
указан неверный тип переменной.

68 Могут ли проявиться ошибки при изменении условий эксплуатации:

Варианты ответа:

+да;

нет.

69 Могут ли проявиться ошибки при изменении в предметной области:

Варианты ответа:

+да;

нет.

70 Возможно ли программирование с защитой от ошибок:

Варианты ответа:

+да;

нет.

71 Есть ли недостатки программирования с защитой от ошибок:

Варианты ответа:

+да;

нет.

72 Защитное программирование это:

Варианты ответа:

+встраивание в программу отладочных средств;

создание задач защищенных от копирования;

разделение доступа в программе;

использование паролей;

оформление авторских прав на программу.

73 Вид ошибки с неправильным написанием служебных слов (операторов):

Варианты ответа:

+синтаксическая;

семантическая;

логическая;

символьная.

74 Вид ошибки с неправильным использованием служебных слов (операторов):

Варианты ответа:

+семантическая;

синтаксическая;

логическая;

символьная.

75 Ошибки при написании программы бывают:

Варианты ответа:

+синтаксические;

орфографические;

лексические;

фонетические;

морфологические.

76 Процедура поиска ошибки, когда известно, что она есть это:

Варианты ответа:

- +отладка;
- тестирование;
- компоновка;
- транзакция;
- трансляция.

77 Программа для просмотра значений переменных при выполнении программы:

Варианты ответа:

- +отладчик;
- компилятор;
- интерпретатор;
- трассировка;
- тестирование.

78 Отладка – это:

Варианты ответа:

- +процедура поиска ошибок, когда известно, что ошибка есть;
- определение списка параметров;
- правило вызова процедур (функций);
- составление блок-схемы алгоритма.

79 Когда программист может проследить последовательность выполнения команд программы:

Варианты ответа:

- +при трассировке;
- при тестировании;
- при компиляции;
- при выполнении программы;
- при компоновке.

80 На каком этапе создания программы могут появиться синтаксические ошибки:

Варианты ответа:

- +программирование;
- проектирование;
- анализ требований;
- тестирование.

81 Когда приступают к тестированию программы:

Варианты ответа:

- +когда программа уже закончена;
- после постановки задачи;
- на этапе программирования;
- на этапе проектирования;
- после составления спецификаций,

82 Тестирование бывает:

Варианты ответа:

- +автономное;
- инструментальное;
- визуальное;

алгоритмическое.

83 Тестирование бывает:

Варианты ответа:

+комплексное;
инструментальное;
визуальное;
алгоритмическое.

84 Существует ли различие между отладкой и тестированием:

Варианты ответа:

+да;
нет.

85 При комплексном тестировании проверяются:

Варианты ответа:

+согласованность работы отдельных частей программы;
правильность работы отдельных частей программы;
быстродействие программы;
эффективность программы.

86 Чему нужно уделять больше времени, чтобы получить хорошую программу:

Варианты ответа:

+тестированию;
программированию;
отладке;
проектированию.

87 Процесс исполнения программы с целью обнаружения ошибок:

Варианты ответа:

+тестирование;
кодирование;
сопровождение;
проектирование.

88 Автономное тестирование это:

Варианты ответа:

+тестирование отдельных частей программы;
инструментальное средство отладки;
составление блок-схем;
пошаговая проверка выполнения программы.

89 Трассировка это:

Варианты ответа:

+проверка пошагового выполнения программы;
тестирование исходного кода;
отладка модуля;
составление блок-схемы алгоритма.

90 Локализация ошибки:

Варианты ответа:

+определение места возникновения ошибки;
определение причин ошибки;

обнаружение причин ошибки;
исправление ошибки.

91 Назначение тестирования:

Варианты ответа:

+повышение надежности программы;
обнаружение ошибок;
повышение эффективности программы;
улучшение эксплуатационных характеристик;
приведение программы к структурированному виду.

92 Назначение отладки:

Варианты ответа:

+поиск причин существующих ошибок;
поиск возможных ошибок;
составление спецификаций;
разработка алгоритма.

93 Инструментальные средства отладки (НЕ правильный ответ):

Варианты ответа:

+компиляторы;
отладчики;
трассировка.

94 Отладка программ это:

Варианты ответа:

+локализация и исправление ошибок;
алгоритмизация программирования;
компиляция и компоновка.

95 Что выполняется раньше, автономная или комплексная отладка:

Варианты ответа:

+автономная;
комплексная.

96 Что выполняется раньше, отладка или тестирование:

Варианты ответа:

+отладка;
тестирование.

97 Что такое автоматизация программирования:

Варианты ответа:

+создание исходного кода программными средствами;
создание исходного кода при помощи компилятора;
создание исходного кода без разработки алгоритма.

98 В чем сущность автоматизации программирования:

Варианты ответа:

+создание программы без написания ее текста;
получение готовой программы без выполнения компоновки;
в отсутствии компиляции.

99 Возможна ли автоматизация программирования:

Варианты ответа:

+да;
нет.

100 Создание исполняемого кода программы без написания исходного кода называется:

Варианты ответа:

составлением спецификаций;
отладкой;
проектированием.
+автоматизацией программирования;

101 Одно из преимуществ автоматизации программирования:

Варианты ответа:

+наглядное программирование с визуальным контролем;
получение стандартной программы;
создание программы с оптимальным кодом.

102 Один из методов автоматизации программирования:

Варианты ответа:

структурное программирование;
модульное программирование;
+визуальное программирование;
объектно-ориентированное программирование.

103 Влияет ли автоматизация программирования на эффективность программы:

Варианты ответа:

+нет;
да

104 Автоматизация программирования позволяет:

Варианты ответа:

повысить надежность программы;
+сократить время разработки программы;
повысить быстродействие программы.

105 Позволяет ли автоматизация программирования всегда создавать эффективные программы:

Варианты ответа:

да.
+нет;

106 Позволяет ли автоматизация программирования всегда создавать надежные программы:

Варианты ответа:

+нет;
да.

107 Недостаток автоматизации программирования;

Варианты ответа:

низкое быстродействие;
+большой размер программы;
сложность программы.

108 Возможны ли ошибки при автоматизации программирования:

Варианты ответа:

+да;

нет.

109 Нужно ли выполнять тестирование при автоматизации программирования:

Варианты ответа:

+да;

нет.

110 Выполняется ли процедура компиляции при автоматизации программирования:

Варианты ответа:

+да;

нет.

111 Что легко поддается автоматизации:

Варианты ответа:

+интерфейс;

работа с файлами;

сложные логические задачи;

алгоритмизация.

112 Относится ли визуальное программирование к средствам автоматизации:

Варианты ответа:

+да;

нет.

113 Нахождение наилучшего варианта из множества возможных:

Варианты ответа:

+оптимизация;

тестирование;

автоматизация;

отладка;

сопровождение.

114 Что такое оптимизация программ:

Варианты ответа:

+улучшение работы существующей программы;

создание удобного интерфейса пользователя;

разработка модульной конструкции программы;

применение методов объектно-ориентированного программирования.

115 Критерии оптимизации:

Варианты ответа:

+время выполнения или размер требуемой памяти;

размер программы и ее эффективность;

независимость модулей;

качество программы, ее надежность.

116 Критерии оптимизации:

Варианты ответа:

+эффективность использования ресурсов;

структурирование алгоритма;

структурирование программы.

117 Возможна ли оптимизация программ без участия программиста:

Варианты ответа:

+да;

нет.

118 Возможна ли оптимизация циклов:

Варианты ответа:

+да;

нет.

119 В чем заключается оптимизация условных выражений:

Варианты ответа:

+в изменении порядка следования элементов выражения;

в использовании простых логических выражений;

в использовании сложных логических выражений;

в использовании операций AND, OR и NOT.

120 Оптимизация циклов заключается в:

Варианты ответа:

+уменьшении количества повторений тела цикла;

просмотре задачи с другой стороны;

упрощение задачи за счет включения логических операций.

121 Оптимизация программы это:

Варианты ответа:

+модификация;

отладка;

повышение сложности программы;

уменьшение сложности программы.

122 Критерии оптимизации программы:

Варианты ответа:

+быстродействие или размер программы;

быстродействие и размер программы;

надежность или эффективность;

надежность и эффективность.

123 Результат оптимизации программы:

Варианты ответа:

+эффективность;

надежность;

машино-независимость;

мобильность.

124 Сущность оптимизации циклов:

Варианты ответа:

+сокращение количества повторений выполнения тела цикла;

сокращение тела цикла;

представление циклов в виде блок-схем;

трассировка циклов;

поиск ошибок в циклах.

125 В чем сущность модульного программирования:

Варианты ответа:

- +в разбиении программы на отдельные функционально независимые части;
- в разбиении программы на отдельные равные части;
- в разбиение программы на процедуры и функции;

126 Можно ли сочетать модульное и структурное программирование:

Варианты ответа:

- +да;
- нет.

127 Может ли модуль включать несколько процедур или функций:

Варианты ответа:

- +да;
- нет.

128 Рекомендуемые размеры модулей:

Варианты ответа:

- +небольшие;
- большие;
- равные;
- фиксированной длины.

129 В чем заключается независимость модуля:

Варианты ответа:

- +в написании, отладке и тестировании независимо от остальных модулей;
- в разработке и написании независимо от других модулей;
- в независимости от работы основной программы.

130 При модульном программировании желательно, чтобы модуль имел:

Варианты ответа:

- большой размер;
- +небольшой размер;
- фиксированный размер;
- любой размер.

131 Модульное программирование это:

Варианты ответа:

- +разбиение программы на отдельные части;
- структурирование;
- использование стандартных процедур и функций.

132 Можно ли использовать оператор GO TO в модульном программах:

Варианты ответа:

- +можно;
- нельзя.

133 Разрешается ли использование циклов при модульном программировании:

Варианты ответа:

- +да;
- нет.

134 Разрешается ли использование условных операторов при модульном программировании:

Варианты ответа:

+да;

нет.

135 Сократится ли размер программы, если ее написать в виде набора модулей:

Варианты ответа:

+нет;

да.

136 Достоинство модульного программирования:

Варианты ответа:

+создание программы по частям в произвольном порядке;

не требует компоновки;

всегда дает эффективные программы;

снижает количество ошибок.

137 Недостаток модульного программирования:

Варианты ответа:

увеличивает трудоемкость программирования;

+усложняет процедуру комплексного тестирования;

снижает быстродействие программы;

не позволяет выполнять оптимизацию программы.

138 Достоинство модульного программирования:

Варианты ответа:

+возможность приступить к тестированию до завершения написания всей программы;

не требует комплексного тестирования;

уменьшает размер программы;

повышает надежность программы.

139 Допустимо ли использование оператора GO TO при структурном программировании:

Варианты ответа:

+нет;

да.

140 Можно ли сочетать структурное программирование с модульным:

Варианты ответа:

+можно;

нельзя;

только в особых случаях.

141 Любую ли программу можно привести к структурированному виду:

Варианты ответа:

+любую;

не все;

нельзя.

142 Можно ли использовать оператор GO TO в структурированных программах:

Варианты ответа:

можно;

+нельзя;

только в особых случаях.

143 Возможно, ли преобразовать неструктурированную программу к структурному виду:

Варианты ответа:

+да;

нет.

144 Возможно ли программирование без оператора GO TO:

Варианты ответа:

+да;

нет.

145 При структурном программировании задача выполняется:

Варианты ответа:

+поэтапным разбиением на более легкие задачи;

без участия программиста;

объединением отдельных модулей программы.

146 Разрешается ли использование оператора GO TO при структурном программировании:

Варианты ответа:

+нет;

да;

иногда.

147 Разрешается ли использование циклов при структурном программировании:

Варианты ответа:

+да;

нет.

148 Разрешается ли использование оператора IF при структурном программировании:

Варианты ответа:

+да;

нет.

149 Программирование без GO TO применяется. при:

Варианты ответа:

+структурном программировании;

модульном программировании;

объектно-ориентированном программировании;

все ответы верные.

150 Достоинство структурного программирования:

Варианты ответа:

+можно приступить к комплексному тестированию на раннем этапе разработки;

можно приступить к автономному тестированию на раннем этапе разработки;

нет необходимости выполнять тестирование;

можно пренебречь отладкой.

151 Достоинство структурного программирования:

Варианты ответа:

+облегчает работу над большими и сложными проектами;

повышает быстродействие программы;

снижает затраты на программирование.

152 Недостаток структурного программирования:

Варианты ответа:

- +увеличивает размер программы;
- снижает эффективность;
- уменьшает количество ошибок;
- не требует отладки.

153 Повышает ли читабельность программ структурное кодирование:

Варианты ответа:

- +да;
- нет.

154 Разрешается ли использование циклов при объектно-ориентированном программировании:

Варианты ответа:

- +да;
- нет.

155 Разрешается ли использование оператора IF при объектно-ориентированном программировании:

Варианты ответа:

- +да;
- нет.

156 Предусматривает ли объектно-ориентированное программирование использование стандартных процедур и функций:

- +да;
- нет.

157 Можно ли сочетать объектно-ориентированное и структурное программирование

Варианты ответа:

- +можно;
- нельзя.

158) Можно ли сочетать объектно-ориентированное и модульное программирование:

Варианты ответа:

- +можно;
- нельзя.

159 Что такое объект, в объектно-ориентированном программировании:

Варианты ответа:

- +тип данных;
- структура данных;
- событие;
- обработка событий;
- использование стандартных процедур.

160 Инкапсуляция это:

Варианты ответа:

- определение новых типов данных;
- определение новых структур данных;
- +объединение переменных, процедур и функций в одно целое;
- разделение переменных, процедур и функций;

применение стандартных процедур и функций.

161 Наследование это:

Варианты ответа:

передача свойств экземплярам;
передача свойств предкам;
+передача свойств потомкам;
передача событий потомкам.

162 Полиморфизм это:

Варианты ответа:

+изменение поведения потомков, имеющих общих предков;
передача свойств по наследству;
изменение поведения потомков на разные события;
изменение поведения экземпляров, имеющих общих предков;

163 Три "кита" объектно-ориентированного метода программирования:

Варианты ответа:

предки, родители, потомки;
+полиморфизм, инкапсуляция, наследование;
свойства, события, методы;
визуальные, не визуальные компоненты и запросы.

164 Какое утверждение верно:

Варианты ответа:

предки наследуют свойства родителей;
родители наследуют свойства потомков;
потомки не могут иметь общих предков;
+потомки наследуют свойства родителей.

165 Может ли дочерний элемент иметь двух родителей:

Варианты ответа:

да;
+нет;
только для визуальных элементов;
если их свойства совпадают.

166 Могут ли два визуальных компонента иметь общего предка:

Варианты ответа:

да;
+нет;
если их свойства совпадают;
если их методы совпадают.

167 Есть ли различие между объектом и экземпляром:

Варианты ответа:

+да;
нет;
если у них общий предок.

168 Есть ли различие в поведении объекта и экземпляра того же типа:

Варианты ответа:

да;

если у них есть общий предок;
+нет;
если у них нет общего предков.

169 Изменение свойств, приводит к изменению поведения экземпляра:

Варианты ответа:

нет;
только для визуальных;
только НЕ для визуальных ;
+да .

170 Можно ли свойствам присваивать значения:

Варианты ответа:

да (всегда);
+не всегда;
нет.

171 Можно ли переопределять методы:

Варианты ответа:

+да;
нет.

172 Можно ли переопределять свойства:

Варианты ответа:

да;
+нет.

173 Могут ли два различных объекта реагировать на событие по-разному:

Варианты ответа:

+да;
нет.

174 Могут ли два экземпляра одного объекта реагировать на событие по-разному:

Варианты ответа:

+да;
нет.

175 Какой методикой проектирования пользуются при структурном программировании:

Варианты ответа:

+сверху вниз;
снизу-вверх.

176 Какой этап проектирования может быть исключен:

Варианты ответа:

+эскизный проект;
технический проект;
рабочий проект.

177 Какие этапы проектирования можно объединять:

Варианты ответа:

+технический и рабочий;
эскизный и рабочий;
технический и эскизный.

178 Модульное программирование применимо при:

Варианты ответа:

проектировании сверху вниз;
+проектирование снизу-вверх;

179 Процесс преобразования постановки задачи в план алгоритмического или вычислительного решения

это:

Варианты ответа:

+проектирование;
анализ требований;
программирование;
тестирование.

180 Составление спецификаций это:

Варианты ответа:

+формализация задачи;
эскизный проект;
поиск алгоритма;
отладка.

181 Этап разработки программы, на котором дается характеристика области применения программы:

Варианты ответа:

+техническое задание;
эскизный проект;
технический проект;
внедрение;
рабочий проект.

182 Укажите правильную последовательность создания программы:

Варианты ответа:

+формулирование задачи, анализ требований, проектирование, программирование;
анализ требований, проектирование, программирование, тестирование, отладка;
анализ требований, программирование, проектирование, тестирование;
анализ требований, проектирование, программирование, модификация, трассировка;
формулирование задачи, анализ требований, программирование, проектирование, отладка.

183 Уточнение структуры входных и выходных данных, разработка алгоритмов, определение элементов интерфейса входят в:

Варианты ответа:

+технический проект;
рабочий проект;
эскизный проект.

184 Несуществующий метод проектирования:

Варианты ответа:

+алгоритмическое;
нисходящее;
восходящее.

185 Метод проектирования:

Варианты ответа:

- +нисходящее;
- алгоритмическое;
- логическое;
- использование языков программирования;
- составление блок-схем.

186 Нисходящее проектирование это:

Варианты ответа:

- +последовательное уточнение (детализация);
- составление блок-схем;
- разделение программы на отдельные участки (блоки);
- трассировка.

187 Признаки нисходящего программирования:

Варианты ответа:

- +последовательная детализация;
- наличие оптимизации;
- наличие тестирования;
- автоматизация программирования.

188 Какой методикой пользуются при структурном программировании:

Варианты ответа:

- +сверху вниз;
- снизу-вверх.

189 Проектирование сверху вниз это:

Варианты ответа:

- +последовательное разбиение общих задач на более мелкие;
- составление из отдельных модулей большой программы.

190 Проектирование снизу-вверх это:

Варианты ответа:

- +составление из отдельных модулей большой программы;
- последовательное разбиение общих задач на более мелкие.

191 Модульное программирование применимо при:

Варианты ответа:

- проектировании сверху вниз;
- проектирование снизу-вверх;
- +и в том, и другом случае;
- ни в коем случае.

192 Какой методикой проектирования пользуются при структурном программировании:

Варианты ответа:

- +сверху вниз;
- снизу-вверх.

193 В чем заключается иерархический подход в решении задачи:

Варианты ответа:

- +в последовательном разбиении задачи на более мелкие составные части;
- в выделении основных и второстепенных элементов;
- в возможности параллельного выполнения отдельных частей задачи.

194 Какой метод проектирования соответствует иерархическому подходу в решении задачи:

Варианты ответа:

+нисходящее (сверху вниз);
восходящее (снизу-вверх).

195 В каких единицах измеряются затраты на проектирование:

Варианты ответа:

+в человеко-днях;
в долларах;
в тенге;
в килобайтах.

196 Зависит ли трудоемкость разработки от сложности алгоритма:

Варианты ответа:

+да;
нет.

197 Зависит ли трудоемкость разработки от количества программистов:

Варианты ответа:

да;
+нет.

198 Зависит ли трудоемкость разработки от языка или системы программирования:

Варианты ответа:

+да;
нет.

199 Зависит ли трудоемкость разработки от количества обрабатываемой информации:

Варианты ответа:

да;
+нет.

200 Зависит ли трудоемкость разработки от вида информации:

Варианты ответа:

+да;
нет.

201 Если вы приобрели программу законным путем, являетесь ли вы собственником программы:

Варианты ответа:

+нет;
да.

202 Если вы приобрели программы законным путем, имеете ли вы право вносить в нее изменения:

Варианты ответа:

+нет;
да

203 Если вы приобрели программы законным путем, имеете ли вы право продать ее:

Варианты ответа:

+да;
нет.

204 Кому принадлежит право собственности на ПО:

Варианты ответа:

+разработчику;
продавцу;
покупателю.

205 Кому принадлежит авторское право на ПО:

Варианты ответа:

+разработчику;
продавцу;
покупателю.

206 Что охраняется законом:

Варианты ответа:

структура базы данных;
+содержание базы данных

Пример практических работ

Практическая работа по теме №3. «Управления проектами внедрения ИС на основе международных стандартов»

Цель работы: ознакомление с содержанием стандартов и приобретение навыков создания и управления проектами на основе международных стандартов.

Краткие теоретические и учебно-методические материалы

Управление проектом в соответствии со стандартом 54869-2011. (Проектный менеджмент. Требования к управлению проектом) включает совокупность процессов инициации, планирования, организации исполнения, контроля и завершения проекта. В рамках процессов управления проектом выполняются действия, относящиеся к следующим функциональным областям управления проектом:

- управление содержанием проекта;
- управление сроками проекта;
- управление затратами в проекте;
- управление рисками проекта;
- управление персоналом проекта;
- управление заинтересованными сторонами проекта;
- управление поставками проекта;
- управление качеством в проекте;
- управление обменом информацией в проекте;
- управление интеграцией проекта.

Последовательность процессов управления проектом определяется условиями конкретного проекта, при этом:
- проект должен начинаться с процесса инициации проекта;

- проект должен оканчиваться процессом завершения проекта;
- выполнение процессов организации исполнения и контроля проекта начинается не раньше процессов планирования.

Процесс инициации проекта. Цель процесса: формальное открытие проекта. Выходы процесса определяются и документируются следующими параметрами проекта:

- наименование проекта;
- причины инициации проекта;
- цели и продукты проекта;
- дата инициации проекта;
- заказчик проекта;
- руководитель проекта;
- куратор проекта.

Процессы планирования внедрения проекта

Процесс планирования содержания проекта. Цель процесса: определение требований проекта и состава работ проекта. Выходы процесса:

а) требования к проекту со стороны заказчика, других заинтересованных сторон проекта, а также законодательства и нормативных актов определены, проанализированы на предмет возможности их выполнения, согласованы с заказчиком проекта и документированы;

б) определены, согласованы с заказчиком и документированы ключевые данные по продукту проекта, а именно:

1. Назначение, свойства и характеристики продукта.
2. Критерии и методы приемки продукта проекта и его составных частей.
3. Допущения и исключения, касающиеся продукта проекта.
4. Определены, согласованы с заказчиком и документированы работы проекта, а также допущения и исключения, касающиеся работ проекта.

Процесс разработки расписания. Цель процесса: определение дат начала и окончания работ проекта, ключевых событий, этапов и проекта в целом. Выходы процесса:

- определены взаимосвязи между работами проекта;
- проведена оценка длительности работ проекта;
- определен и утвержден график привлечения ресурсов, необходимых для выполнения проекта в срок;
- определено и документировано расписание проекта;
- утвержден базовый календарный план проекта.

Процесс планирования бюджета проекта. Цель процесса: определение порядка и объема обеспечения проекта финансовыми ресурсами. Выходы процесса:

- определена и документирована структура статей бюджета проекта, позволяющая контролировать затраты на проект в ходе его реализации;
- определена плановая стоимость всех ресурсов проекта (материальных и людских) с учетом всех известных ограничений на их использование;
- определена стоимость выполнения работ проекта;

- утвержден базовый бюджет проекта;
- определен и документирован порядок поступления денежных средств в проект.

Процесс планирования персонала проекта. Цель процесса: определение порядка обеспечения проекта человеческими ресурсами. Выходы процесса:

- определены и документированы роли участников проекта, их функции и полномочия;
- определен численный и квалификационный состав команды проекта, а также требования к условиям труда;
- персонально определены основные члены команды проекта.

Процесс планирования закупок в проекте. Цель процесса: определение порядка и объема обеспечения проекта продукцией и услугами, приобретаемыми у сторонних организаций. Выходы процесса:

- а) проведен анализ необходимости закупки продукции и услуг для достижения целей проекта;
- б) в случае если по результатам анализа принято решение о целесообразности закупок продукции и/или услуг в проекте, то:
 - определены требования к закупаемой продукции (услугам), в том числе ограничения по стоимости и срокам поставки;
 - определены требования к приемке закупаемой продукции (услугам);
 - запланированы мероприятия по выбору и оценке поставщиков на основе определенных критериев.

Процесс планирования реагирования на риски. Цель процесса: определение основных рисков проекта и порядка работы с ними. Выходы процесса:

- выявлены и документированы риски проекта;
- проведены оценка и ранжирование по вероятности и степени влияния на результат проекта всех идентифицированных рисков;
- разработаны мероприятия по изменению вероятности и степени влияния наиболее значимых рисков, а также созданы планы реагирования на случай возникновения таких рисков;
- учтены результаты разработки упреждающих мероприятий по реагированию на риски в связанных с ними планах.

Процесс планирования обмена информацией в проекте. Цель процесса: определение порядка обмена информацией между лицами, участвующими в реализации проекта и заинтересованными в результатах проекта. Выходы процесса:

- определены все участники информационного обмена, а также их потребности в информации;
- определены методы и средства распространения информации по проекту;
- определена процедура разработки, согласования, утверждения, распространения проектных документов;
- определены место и правила хранения информации по проекту.

Процесс планирования управления изменениями в проекте. Цель процесса: определение порядка работы с изменениями в проекте. Выходы процесса:

1. Выявление изменений.
2. Согласование и утверждение изменений.
3. Организация учета версий документов и продуктов проекта.
4. Доведение информации об изменениях до заинтересованных сторон.

Процесс организации исполнения проекта. Цель процесса: организация выполнения проекта согласно разработанным планам. Выходы процесса:

- выполнены запланированные работы;
- получены продукты проекта;
- осуществлены изменения согласно принятым в проекте правилам;
- выполнены намеченные корректирующие и предупреждающие действия;
- актуализированы документы по управлению проектом.

Процесс контроля исполнения проекта. Цель процесса: проверка соответствия процессов и продукта проекта установленным требованиям. Выходы процесса:

- документированы результаты регулярной проверки состояния проекта, в частности отклонения от планов, и проанализированы с целью определения причин отклонений;
- произведена оценка соответствия продукта проекта требованиям к нему;
- сформированы корректирующие и предупреждающие действия по результатам проверки;
- отчеты о выполнении работ проекта соответствуют утвержденной системе отчетности по проекту.

Процесс завершения проекта. Цель процесса: формальное закрытие проекта. Выходы процесса:

- проведена и документально оформлена приемка продукта проекта заказчиком;
- проведено закрытие всех договоров по проекту (в случае их наличия);
- документировано окончание проекта;
- сформирован архив проекта;
- команда проекта и основные заинтересованные стороны проинформированы об окончании проекта.



Рисунок 1 - Основные понятия проектного менеджмента и их взаимосвязь

Задания на практическую работу:

1. Провести формальное открытие проекта по одной из предложенных тем:
 - Создание и внедрение сайта компании.
 - Создание и внедрение интернет-магазина.
 - Разработка мобильного приложения.
2. Определить наименование проекта, причины инициации проекта, цели и продукты проекта, дату инициации проекта, заказчика проекта, руководителя проекта и куратора проекта.
3. Осуществить планирование персонала проекта. Определить роли участников проекта, их функции и полномочия, численный, персональный и квалификационный состав команды проекта, а также требования к условиям труда.
4. Разработать планы реагирования на риски проекта. Выявить риски проекта, провести оценку и ранжирование по вероятности и степени влияния на результат проекта всех идентифицированных рисков, создать план реагирования на случай возникновения рисков, разработать упреждающие мероприятия по реагированию на риски.

Содержание отчета:

1. Отчет об открытии проекта: наименование проекта, причины инициации проекта, цели и продукты проекта, дату инициации проекта, заказчика проекта, руководителя проекта и куратора проекта.
2. Перечень участников проекта, их функции и полномочия, численный, персональный и квалификационный состав команды проекта.
3. Выявленный перечень рисков проекта.
4. Разработанный план реагирования на риски проекта.
5. Перечень упреждающих мероприятий по реагированию на риски.

Контрольные вопросы:

1. Основные международные стандарты по управлению проектами?
2. Функциональные области управления проектом.
3. Содержание процесса инициации проекта.
4. Содержание процесса планирования бюджета проекта.
5. Содержание процесса планирования содержания проекта.

Практическая работа № 4 «Настройка и конфигурирование DNS сервера»

Цель работы:

- Получить представление о работе DNS сервера.
- Получить практические навыки использования утилит работы с серверами системы DNS и конфигурирования системы.

Краткие теоретические и учебно-методические материалы

Система DNS – распределенная база данных хранящая соответствие между **IP адресом** и **доменным именем** компьютера.

Система DNS – **клиент - серверная**. DNS-клиент получает в качестве конфигурационного параметра IP адрес обслуживающего DNS-сервера и получает к нему доступ напрямую.

На сервере DNS могут присутствовать множество записей разных типов и назначения.

Диагностику работы DNS с клиента можно выполнять с помощью команд `ping` (формальная проверка разрешения имени) и с помощью консольной утилиты **nslookup** (работа с DNS сервером в режиме запрос-ответ).

Задания для практической работы:

Необходимое обеспечение:

- Установленная система виртуализации.
- Виртуальные машины Windows Server.

Часть 1. Освоение утилиты nslookup

1. Используя встроенную справку и доступные материалы в Web выяснить:
 - Назначение и формат следующих типов записей DNS: **SOA, A, NS, MX, CNAME**;
 - Значение и взаимосвязь терминов «**домен**» и «**доменная зона**»;
 - Значение термина «**зона обратного просмотра**»;
 - Значение термина «**делегирование домена**».
2. С помощью консольной утилиты **nslookup**:
 - Определить адреса хостов, обслуживающих почтовый домен `yandex.ru`
Примечание: запрос необходимо выполнить к NS северу сети RunNet (домен `runnet.ru`), для чего необходимо выяснить имена или адреса DNS серверов зоны `runnet.ru`.
 - Определить каноническое имя (CNAME) для хоста `www.ifmo.ru`.
 - Определить e-mail администратора DNS сервера зоны `ifmo.ru` (запрос можно к DNS серверу зоны `ifmo.ru`).

Часть 2. Управление и настройка DNS-сервера под Windows Server

1. Подготовить два (Б и Д) компьютера с **Windows Server**. Согласовать настройку сети с преподавателем. Проброс сети в виртуальной машине должен быть настроен на режим «**сетевой мост**».
2. Установить пакет **support tools** (он содержит необходимую для работы утилиту `dnscmd.exe`). В конфигурации TCP/IP установить согласованный с преподавателем **IP адрес и адрес DNS** равный IP.
3. Разработать план доменного дерева со следующими условиями:
 - **Сервер Б** должен содержать зону, поддерживающую домен **инициалы.local** (например `adb.local`);
 - **Сервер Б** должен содержать зону **обратного просмотра** для IP сети, в которой будут находиться сервера Б и Д;
 - В зоне **прямого просмотра сервера Б** должна быть заведена запись **типа А** для сервера Б;
 - В зоне **прямого просмотра сервера Б** должен быть создан поддомен **sub1.инициалы.local**, все записи которого хранятся в зоне сервера Б;
 - В зоне **прямого просмотра сервера Д** должен быть создан поддомен **sub2.инициалы.local**;
 - В зоне **прямого просмотра сервера Б** должно быть назначено **делегирование** домена **sub2.инициалы.local** в зону сервера Д;

- Все ссылки в **SOA** на **DNS** серверах должны быть сделаны через **псевдонимы** с именем **ns**;
 - **Сервер Д** должен содержать **дополнительную** зону обратного просмотра для зоны обратного просмотра с сервера Б, должно быть включено **уведомление** об изменениях и **ограничено** предоставление копии зоны только для сервера Д;
 - В доменах **инициалы.local**, **sub1.инициалы.local** и **sub2.инициалы.local** должны быть **A записи** на хосты с именами **srv** и **ip** равными **ip-адресам** сервера, поддерживающего домен, в котором создается запись.
4. Установить и настроить **DNS сервера** на компьютерах Д и Б согласно **п.5**.
 5. Установить, на каких **номерах портов** и по каким **протоколам транспортного уровня** работает DNS сервер.
 6. Изучить **содержимое** файлов зон (сохранить их для отчета).
 7. С помощью утилит **dnscmd** получить **список всех зон** на обоих серверах, и **содержимого** зоны **инициалы.loc** (сохранить их для отчета).
 8. Разобраться в назначении **других ключей** утилиты **dnscmd**.
Убедиться, что на сервере Б корректно разрешается имена:
 - **srv.инициалы.local**;
 - **srv.sub1.инициалы.local**;
 - **srv.sub2.инициалы.local**.
 Сохранить для отчета вывод команд.

Часть 3. Рекурсивный поиск по дереву DNS

1. Перенастроить **DNS сервер Б**, поменяв IP адрес по указаниям преподавателя и переключив проброс сети в виртуальной машине на **режим «NAT»**.
2. Настроить DNS-сервер так, чтобы он запрашивал **внешний сервер** с адресом **194.85.32.18** в случаях, когда сам **не способен** разрешить имена. (Параметр **Forwarders** в Свойствах сервера).
3. Проверить **корректность** разрешения имени **www.google.ru** при работе через DNS.
4. **Удалить** настройку **Forwarders** и **очистить кэш** сервера не перезагружая его.
5. Проверить корректность разрешения имени **www.google.ru** при работе через DNS в **новой конфигурации**.
6. С помощью любой программы анализатора трафика (например, **wireshark**) установить этапы работы алгоритма разрешения имени в п.3 и п. 5. **Сохранить** перехваченные сообщения для отчета.

Содержание отчета:

1. Консольный вывод команды **nslookup** части 1 п. 2.
2. Файлы зон с серверов Б и Д из части 2 п.6.
3. Вывод команд из части 2 п. 7, 8.
4. Перехваченные сообщения разрешения имени из части 3 п. 6.

Для выполнения работы в иной ОС (например, Linux) следует готовить этот отчет в терминах и применительно к другой ОС.

Контрольные вопросы:

2. Для чего предназначены основные типы записей DNS?
3. В каком режиме работал DNS-сервер в части 3 п. 3 и в п. 5 (рекурсивном или нет)?
4. Что такое корневые ссылки? Привести несколько адресов корневых DNS серверов «известных» созданному DNS-серверу по умолчанию.
5. Разрешение имени в части 3 п. 3 и п. 5 происходило с разной скоростью. Почему?

6. В чем назначение зоны обратного просмотра?
7. Как определить, какие хосты обрабатывают почту, направленную в домен yandex.ru?