



Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Калининградский государственный технический университет»
(ФГБОУ ВО «КГТУ»)

УТВЕРЖДАЮ
Начальник УРОПС

Фонд оценочных средств
(приложение к рабочей программе модуля)
«ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

основной профессиональной образовательной программы бакалавриата
по направлению подготовки

09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Профиль программы
**«АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И
УПРАВЛЕНИЯ»**

ИНСТИТУТ
РАЗРАБОТЧИК

Институт цифровых технологий
Кафедра систем управления и вычислительной техники

1 РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Таблица 1 – Планируемые результаты обучения по дисциплине, соотнесенные с установленными индикаторами достижения компетенций

Код и наименование компетенции	Индикаторы достижения компетенции	Дисциплины	Результаты обучения (владения, умения и знания), соотнесенные с компетенциями/индикаторами достижения компетенции
ПК-1: Способен разрабатывать требования, проектировать программное обеспечение (далее - ПО), документировать процедуру тестирования	ПК-1.1: Анализирует требования и проектирует программное обеспечение; ПК-1.2: Документально сопровождает процедуры тестирования программного обеспечения; ПК-1.3: Проектирует программное обеспечение	Программная инженерия	<p><u>Знать:</u> основные этапы создания программных средств и информационных технологий;</p> <ul style="list-style-type: none"> - структуру жизненного цикла программной системы, основные модели жизненного цикла, перечень стадий и основных этапов каждой стадии жизненного цикла; - отечественные и международные стандарты, регламентирующие процессы создания программных средств. <p><u>Уметь:</u> использовать на практике основные метрики программного обеспечения;</p> <ul style="list-style-type: none"> - формировать модели, описывающие различные аспекты предметной области проектируемой программной системы; <p><u>Владеть:</u> средствами планирования и управления процессом разработки программной системой.</p>

2 ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПОЭТАПНОГО ФОРМИРОВАНИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ (ТЕКУЩИЙ КОНТРОЛЬ) И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

2.1 Для оценки результатов освоения дисциплины используются:

- оценочные средства текущего контроля успеваемости;
- оценочные средства для промежуточной аттестации по дисциплине.

2.2 К оценочным средствам для текущего контроля успеваемости относятся:

- вопросы опроса после каждой лекции;
- задания и контрольные вопросы по лабораторным работам;

- тестовые задания.

2.3 К оценочным средствам для промежуточной аттестации по дисциплине, проводимой в форме экзамена, относятся:

- задания и контрольные вопросы по курсовой работе;
- экзаменационные вопросы.

3 ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

3.1 *Вопросы опроса после лекций.* Опрос после каждой лекции в виде ответа на пять вопросов по варианту. Перечень вопросов после каждой лекции:

Лекция 1

1. Сколько лет программирование носит массовый характер? (С какого времени программирование стало массовым?)
2. Что первично: алгоритм или программа?
3. Назовите основные причины кризиса программирования в 60-х годах 20-го столетия.
4. Алгоритм является компьютерно-ориентированным?
5. Что стоит слева при описании технологической операции?
6. Какое требование предъявляется к оформлению входной и выходной информации технологической операции?
7. Почему документация проектирования программного обеспечения должна оформляться на основе стандартов?
8. Что такое **метасреда** для программного изделия?
9. Как в настоящее время называется процесс создания больших программных систем коллективами разработчиков?
10. Назовите основные элементы методологии.
11. Что стоит справа при описании технологической операции?
12. Зачем проектирование программного обеспечения должно вестись в рамках стандартов?
13. Что такое «**стандарт**»?
14. В чем заключается двойственность методологии программирования?
15. Какие методологии используются для формирования листинга программ?
16. Какие методологии используются для проектирования программных приложений?
17. Что такое **нотация**?
18. Что является более общим: методология или технология?
19. Укажите примеры нотаций при разработке программного приложения.

20. Сколько групп стандартов используется при проектировании программных систем?
21. Кто (что) являются исполнителями технологической операции?
22. На основе чего оцениваются результаты выполнения технологических операций?
23. Что включает технология проектирования программного обеспечения?
24. Что такое «**критерий**»?
25. Какое требование предъявляется к входной и выходной информации технологической операции?
26. Что регламентирует процесс выполнения технологической операции?
27. Сколько групп стандартов используются при проектировании программного обеспечения?
28. В чем выразился *кризис программирования* в 60-х годах 20-го столетия?
29. Что такое CASE-средства?
30. В каком элементе проектирования программных систем должна указываться ответственность исполнителей?
31. Что указывается сверху у описания технологической операции?
32. Дайте определение системы.
33. Какие виды ограничений накладываются на программный проект?
34. Программа является компьютерно-ориентированной?
35. Какие расширения у исполняемых файлов программного приложения?
36. Чем программный продукт отличается от просто программы?
37. Приведите пример применения стандарта в повседневной жизни.
38. Какое программное изделие обязательно должно включать в себя документацию по системе?
39. С чего начинается проектирование программной системы при структурном подходе?
40. Что собой представляет программа в русле функциональной методологии?
41. Какие виды ресурсов нужны для создания программных продуктов?
42. Какие файлы, кроме исполняемых, сейчас обычно входят в состав программных систем?
43. Какой стандарт определяет вид компонентов пользовательского интерфейса?
44. Приведите пример CASE-средств, используемых при создании программных продуктов.
45. Что стоит снизу при описании технологической операции?
46. Какой подход используется для создания программной системы как элемента метасреды?
47. Почему терминология программной инженерии до сих пор не устоялась?
48. С какими научными дисциплинами наиболее тесно связана программная инженерия?

49. Какое явление привело к созданию программной инженерии?
50. В рамках одной методологии могут существовать несколько технологий, или наоборот, - в рамках одной технологии могут существовать несколько методологий?

Лекция 2

1. Как называется книга Брукса?
2. На какой стадии разработки программной системы наиболее актуально участие будущих пользователей?
3. Какие принципиальные отличия программных проектов от технических проектов?
4. Как расшифровывается название языка COBOL?
5. Какие стадии исходно входили в состав жизненного цикла программной системы?
6. На какие временные отрезки делится жизненный цикл программной системы?
7. Что дает программисту умение обобщать типичные ситуации?
8. На какой вопрос отвечает фаза анализа жизненного цикла программной системы?
9. В каком году Брукс опубликовал 1-й вариант своей книги?
10. В чем трудность контроля и управления ходом разработки программных проектов?
11. Какие ошибки наиболее трудно исправлять в ходе работы над программным проектом?
12. Какие стадии жизненного цикла входят в фазу синтеза программного обеспечения?
13. Какая из рассмотренных моделей жизненного цикла программной системы предполагает более узкую специализацию участников разработки?
14. Для решения каких задач предназначался язык COBOL?
15. Какой ГОСТ определял состав жизненного цикла программной системы в Советском Союзе?
16. Какой процент программного обеспечения в настоящее время подвергается модернизации?
17. На какой вопрос отвечает фаза синтеза жизненного цикла программной системы?
18. Что представляет стадия в рамках жизненного цикла программной системы?
19. Чему посвящена книга Брукса?
20. Кто разработал язык COBOL и в каком году?
21. В чем достоинства каскадной модели жизненного цикла программной системы?
22. Какие существуют варианты доступа пользователей к компьютеру?
23. Насколько может отличаться производительность разработчиков программных систем?
24. Как расшифровывается ТЗ?

25. Почему принципиально невозможно получить надежную программную систему?
26. Какой временной период охватывает жизненный цикл программной системы?
27. Какая нотация используется для отображения моделей жизненного цикла программной системы?
28. Когда определяется экономическая целесообразность затраченных средств при каскадной модели жизненного цикла?
29. Какая модель жизненного цикла программной системы была исторически первой?
30. Что позволило реализовать диалоговый режим пользователей к компьютеру?
31. В чем недостатки каскадной модели жизненного цикла программной системы?
32. Какой закон сформулировал Брукс?
33. Что такое «аудит»?
34. Укажите время появления первых моделей жизненного цикла программной системы.
35. Какие функции возлагаются в процессе разработки программной системы на будущих пользователей?
36. Как определяется общее количество пар в группе из N разработчиков?
37. Какое свойство является основным у программных систем?
38. Почему перестает расти производительность группы разработчиков программной системы при переходе через некоторый порог?
39. Какие виды «зрения» должны быть у профессионального программиста?
40. Кто создал язык COBOL?
41. В настоящее время возможности «железа» опережают возможности программного обеспечения или отстают?
42. Что такое «жизненный цикл» программной системы?
43. Для чего желательно разработчику ПО обобщать типовые ситуации?
44. Какая стадия является первой в советской структуре жизненного цикла программной системы?
45. К чему относится название Waterfall?
46. В чем достоинство подключения заказчика к программному проекту с первых стадий?
47. Почему в каскадной модели жизненного цикла можно достичь высокой специализации разработчиков?
48. Почему все компьютеры принципиально ненадежны?
49. Какие участники разработки программных проектов могут выполнять по две роли?
50. На какие временные фазы делится жизненный цикл программной системы?

Лекция 3

1. Какие участники отсутствуют при разработке коробочного ПО?
2. В какой модели жизненного цикла программной системы сложнее документировать этапы проектирования? Почему?
3. В какой модели жизненного цикла программной системы раньше можно определить экономическую целесообразность разработкой? Почему?
4. В какой модели жизненного цикла программной системы более широко можно использовать повторяющиеся проектные решения? Почему это выгодно?
5. Какая из рассмотренных моделей жизненного цикла программной системы предполагает более узкую специализацию участников разработки?
6. В какой модели жизненного цикла программной системы труднее реализовать контроль и управление разработкой? Почему?
7. Какая модель жизненного цикла программной системы наиболее полно соответствует рыночной экономике? Почему?
8. Какая модель жизненного цикла программной системы может учитывать интересы пользователя?
9. Что такое **системы реального времени**?
10. Какая стадия жизненного цикла программной системы является самой затратной?
11. Сколько стадий включает жизненный цикл программной системы по современным стандартам?
12. Содержимое плана сопровождения программной системы
13. Какая работа проводится с планами на более поздних этапах разработки?
14. Дайте определение плана
15. Какие задачи ставятся перед экспертами при оценке идеи программного проекта?
16. Как определяется потребность в заказном программном продукте?
17. На каких этапах разработки планирование проводится более интенсивно?
18. На какие вопросы должны ответить эксперты в процессе обсуждения идеи программного продукта?
19. По каким причинам эксперты могут отказать в реализации предлагаемому программному проекту?
20. Укажите содержимое плана по управлению персоналом
21. Какие ресурсы требуются для реализации программных проектов?
22. Какого размера должно быть описание предлагаемого программного проекта?

23. Кто принимает решение о переходе к реализации предлагаемого программного проекта для коробочного ПО?
24. Что такое аттестация программной системы? Содержимое плана аттестации
25. Кто принимает решение о переходе к реализации предлагаемого программного проекта для заказного ПО?
26. Дайте определение термину планирование
27. В какой последовательности решаются две основные задачи планирования?
28. В ходе какой стадии формируется график работ над программным проектом?
29. В ходе какой стадии производится оценка затрат на конкретные задачи и проект в целом?
30. В ходе какой стадии определяются требуемые ресурсы для выполнения проекта?
31. Что такое критический путь сетевого графика?
32. Что предполагает итерационность процесса планирования?
33. В ходе какой стадии строится список задач, которые необходимо решить в ходе проектирования программного изделия?
34. Какие по характеру планы рекомендуется составлять на стадии планирования?
35. Содержимое плана качества
36. Что такое независимый путь в сети?
37. Какой физический смысл у связей между вершинами в сети работ над проектом?
38. Когда проводится аттестация программной системы?
39. В чем заключается аттестация ПС?
40. Какая модель жизненного цикла ПС позволяет полнее учитывать пожелания будущих пользователей?
41. Почему для коробочного ПО более затратным является этап определения исходных требований?
42. Что такое «маркетинг рынка»?
43. Какая модель жизненного цикла ПС должна использоваться для систем реального времени?
44. Какая графическая структура называется сетью?
45. С какого момента разработки специалисты по сопровождению ПС подключаются к ее разработке?
46. Сколько стадий у жизненного цикла ПС, создаваемых фирмой Oracle?
47. Какие ограничения всегда существуют для программных проектов?

48. Укажите время появления спиральной модели жизненного цикла ПС.
49. Что должны предусмотреть эксперты при анализе одностороннего описания предлагаемого программного продукта?
50. Назовите основные направления работы фирмы Oracle

Лекция 4

1. Как называются специалисты по подготовке документации программной системы?
2. Какая информация нужна специалистам по сопровождению разработанной программной системы?
3. Как рассчитывается качество проекта на основе размерно-ориентированных метрик?
4. Для чего нужны исторические справки по предыдущим проектам?
5. Как рассчитывается удельная стоимость одной строчки кода проекта?
6. Что входит в расходы на персонал фирмы?
7. Какие метрики программного проекта были исторически первыми?
8. Когда рассчитывается ТЭО программного проекта?
9. Конечная цель создания разных моделей на этапе анализа стадии «Формирование и анализ требований»?
10. От чего зависит точность оценки стоимости программного проекта?
11. Дайте определение метрики программного проекта.
12. Какие длительности разработки программного проекта сейчас считаются невозможными?
13. Какие параметры программного проекта рассчитываются при оценке стоимости программного проекта?
14. В чем состоит цель стадии «формирование и анализ требований»?
15. Что означает вариант построения моделей «AS-IS»?
16. Какие данные используются для расчета предварительной стоимости разработки программного проекта?
17. Как расшифровывается КТС?
18. Какие длительности разработки программной системы сейчас считаются нерациональными?
19. Перечислите задачи стадии «формирование и анализ требований»
20. Какие модели создаются в процессе анализа предметной области разрабатываемой программной системы?
21. Что такое «предметная область» разрабатываемой программной системы?

22. Какое значение размера задачи рассчитывается на основе формул в ходе предварительной оценки стоимости программной системы?
23. Задачи системного анализа при проектировании программной системы?
24. Для чего конкретно используются сведения о предыдущих программных проектах при создании новых проектов?
25. Сколько вариантов размера задачи будущего проекта определяются на предварительном этапе?
26. Перечислите составляющие затрат на программный проект.
27. Как расшифровывается аббревиатура CASE?
28. Как меняются расходы на персонал с увеличением размера проекта?
29. Какой этап стадии «формирование и анализ требований» может отсутствовать?
30. Как меняются расходы на аппаратно-программные средства с увеличением размера проекта?
31. При каком условии может снижаться стоимость программного проекта?
32. Что такое **срок окупаемости** программного проекта?
33. Как расшифровывается FP? Что это такое?
34. Почему в настоящее время резко уменьшились расходы на командировки разработчиков проекта?
35. В каких программных проектах отсутствует этап анализа предметной области?
36. Почему нежелательно разработчикам резко преувеличивать расходы на разработку программного проекта?
37. Как расшифровывается LOC?
38. Что означает буква **L** в формуле трапеции?
39. Что означает буква **V** в формуле Симпсона?
40. Как расшифровывается ТЭО?
41. С какого момента специалисты по сопровождению подключаются к разработке программного проекта?
42. Что входит в конфигурацию программной системы?
43. Какие метрики программного проекта оперируют количеством строк кода?
44. Перечислите задачи, реализуемые в процессе управления конфигурацией.
45. Укажите виды документации, создаваемой в ходе проектирования программной системы.
46. Что входит в CASE-средства?
47. Что указывается в информации для управления программным проектом?

48. Укажите виды специалистов, задействованных в программном проекте.
49. Как рассчитывается производительность в проекте на основе размерно-ориентированных метрик?
50. Дайте определение CASE-средствам.

Лекция 5

1. Что отображается снизу дуги в нотации ДПС?
2. Что входит в функциональные требования к проектируемой ПС?
3. Как отображаются сущности в нотации Silverrun?
4. Кто является основоположником моделей ERD?
5. Какие специалисты используют в своей работе спецификацию проекта ПС?
6. Когда создается SRS?
7. Какая модель используется для описания поведения ПС при объектном подходе к проектированию?
8. Что располагается сверху в моделях Sequence diagrams?
9. Дайте определение связи как элемента ERD
10. Перечислите классификации требований.
11. Что предполагает парадигма «ТО-ВЕ» при системном анализе?
12. Кто является участниками процесса «Управление требованиями»?
13. Как называется второй этап стадии «Формирование и анализ требований»?
14. Что такое идентификатор сущности?
15. Какое требование предъявляется к языкам спецификаций?
16. Как называется язык, используемый для создания диаграмм для проектирования ПС при объектном подходе?
17. Какая модель описывает сервисы системы при структурном подходе к проектированию ПС?
18. Как выделяется идентификатор в нотации Silverrun?
19. Что является целью процесса «Управление требованиями»?
20. Какие требования предъявляются к содержимому спецификации?
21. К какому подходу к проектированию принадлежит модели SADT?
22. Какая модель описывает сервисы системы при объектном подходе к проектированию ПС?
23. У ДПС может быть несколько конечных состояний?
24. Дайте определение атрибута (свойства) как элемента ERD.
25. Укажите возможные виды языков спецификаций. Какие из них наиболее информативны?
26. Какой международный стандарт определяет структуру SRS?
27. Какая модель описывает режимы работы системы при структурном подходе?

28. Как расшифровывается SADT?
29. Какая модель в дальнейшем используется как основание для логической схемы базы данных?
30. Что такое «плавательная дорожка»?
31. Какой русский термин является синонимом термина «спецификация»?
32. Как отображается начальное состояние в нотации ДПС?
33. Что указывает стрелка сверху в модели SADT?
34. Какой раздел оформленных требований является наиболее важным?
35. Какие модели чаще всего строятся на стадии «Формирование и анализ требований»?
36. Кто может быть актором?
37. Вариантом какой математической абстракции являются ДПС?
38. Для чего заказчики используют спецификацию?
39. Что указывает стрелка снизу в модели SADT?
40. Какая модель называется концептуальной?
41. Что должно явиться итогом системного анализа сформированных моделей на стадии «Формирование и анализ требований»?
42. Как связываются между собой диаграммы разных уровней в моделях SADT?
43. Когда формируется спецификация ПС?
44. Как отображается конечное состояние в нотации ДПС?
45. Чем являются сформированные требования для разработчиков ПС?
46. Дайте определение сущности как элемента ERD
47. Что собой представляет SRS?
48. Для чего руководителям разработки нужна спецификация?
49. Что указывает стрелка слева в модели SADT?
50. Какая модель используется для описания сервисов ПС при объектном подходе к проектированию?

Лекция 6

1. Какое требование предъявляется к цели программного проекта?
2. Что происходит со спецификациями в процессе проектирования ПС?
3. Что дает распараллеливание проектирования разных модулей ПС?
4. Когда заканчивается процесс детализации спецификаций в процессе проектирования ПС?
5. Дайте определение понятию «проект»

6. У каких программных компонентов есть интерфейсы?
7. Какое проектирование не привязывается к конкретным средствам поддержки функционирования проектируемой ПС?
8. Может ли модуль входить в состав модуля?
9. Какой граф строится в результате пошаговой декомпозиции функционала проектируемой ПС?
10. Почему при модульном программировании легче контролировать ход проектирования?
11. Какой графической структурой представляется процесс разработки ПС?
12. Какие средства реализуют поддержку программному проекту?
13. Дайте определение понятию «архитектура ПС»
14. Какое программирование называется модульным?
15. Что является объектом пошаговой декомпозиции при разработке ПС?
16. Какие этапы проектирования ПС могут быть перенесены на стадию кодирования?
17. Что находится на входе структурного подхода к проектированию ПС?
18. На основании какой информации происходит разбивка проектируемой ПС на подсистемы?
19. Какие процессы декомпозируются в первую очередь при структурном подходе?
20. Что собой представляет логическое проектирование ПС?
21. Какие принципы лежат в основе декомпозиции функций проектируемой ПС?
22. Может ли модуль входить в состав подсистемы?
23. При корректном разбиении ПС на модули через что происходит общение модуля с другими модулями?
24. Дайте определение понятию «проектирование»
25. В чем достоинство разбивки проектируемой ПС на подсистемы?
26. На какой главный вопрос должно ответить проектирование ПС?
27. Какая функциональная модель может использоваться для описания декомпозиции функций проектируемой ПС?
28. Какой переход реализуется в процессе проектирования ПС?
29. В какой парадигме формируются функциональные модели в процессе проектирования ПС?
30. Для чего выделяются аналогичные блоки на каждом шаге детализации программного проекта?
31. Что должна представлять стандартная структура модуля?

32. В чем принципиальное различие подсистемы и модуля?
33. Дайте определение понятию «проектирование ПС»
34. Какие синонимы существуют у названия «блочно-иерархический подход»?
35. Как могут соотноситься во времени задачи программного проекта?
36. Почему каждый программный проект является уникальным?
37. Что предполагает итерационность процесса проектирования программной системы?
38. Процесс проектирования программной системы при структурном подходе идет «сверху-вниз» или «снизу-вверх»?
39. В чем суть принципа «разделяй и властвуй»?
40. Какие этапы включает архитектурное проектирование?
41. Может создаваться ПС без разбивки на подсистемы?
42. Что является более общим понятием: подсистема или модуль?
43. Дайте определение понятию «интерфейс»
44. Из каких двух частей должен состоять стандартно модуль?
45. Что является исходным для процесса проектирования ПС?
46. Что включает в себя модуль по современному определению?
47. Что собой представляет физическое проектирование ПС?
48. Почему при модульной декомпозиции надо стремиться к минимуму связей между программными модулями?
49. Чем программный проект принципиально отличается от технического проекта?
50. Может ли подсистема входить в состав модуля?

Лекция 7

1. Что возрастает с увеличением числа модулей программной системы?
2. На основании чего происходит выбор алгоритма разработчиком?
3. Какая исключительная ситуация может возникнуть во время работы рекурсии?
4. Дайте определение **структуры данных**.
5. Какой граф называется деревом?
6. Какие структуры данных называются физическими?
7. На основании какого компромисса, в конечном итоге, определяется количество модулей программной системы?
8. Дайте определение **итерации**?
9. Что такое **триггер**?
10. Что такое **компонентное программирование**?

11. На основе какой информации формируются структуры данных программной системы?
12. Укажите наиболее распространенные модели структурирования программных систем
13. Что такое **связность** модуля программной системы?
14. Дайте определение **рекурсии**?
15. Как на технологической схеме отображается клавиатура?
16. На какой стадии определяется информация для формирования структур данных программной системы?
17. Что такое **репозиторий**?
18. При какой связности модуль представляет «черный ящик»?
19. Возможно ли преобразовать рекурсивный алгоритм в итерационный?
20. Какой объем информации может хранить один триггер?
21. Как по-другому называются простые структуры данных?
22. Как на технологических схемах обозначается дисплей?
23. Что определяет характеристика **сцепление** модуля.?
24. Какие модели управления используются в программных системах?
25. Какой показателем описывается скорость работы алгоритма?
26. Какие структуры данных называются простыми?
27. Как на технологических схемах обозначается устройство вывода печатного документа?
28. Как должны соотноситься связность и сцепление для каждого модуля при корректной разбивке программной системы на модули?
29. Возможно ли преобразовать итерационный алгоритм в рекурсивный?
30. Что такое **рабочая станция**?
31. Чем отличается структура данных **стек** от структуры **очередь**?
32. Как на технологических схемах обозначается устройство долговременной памяти?
33. На кого возлагаются функции экспертного определения компромисса между стоимостью разработки одиночных модулей и стоимостью межмодульного интерфейса?
34. Показатель $O(n)$ характеризует максимальное, минимальное или среднее время поиска в неотсортированном списке?
35. На каких стадиях жизненного цикла программной системы может производиться алгоритмизация ее задач?
36. К какому виду структур данных относятся массивы?

37. Где хранится промежуточная информация во время работы рекурсии?
38. В чем особенность спецификаций самого низкого уровня при модульном разбиении программной системы?
39. Показатель $O(\log_2 n)$ характеризует максимальное, минимальное или среднее время поиска в отсортированном списке?
40. В чем сложность отладки программной системы при использовании модели управления по прерываниям?
41. Массив – это логическая или физическая структура данных?
42. Что собой представляет компонент при компонентном программировании?
43. Дайте определение **алгоритму**?
44. Определение модуля как **черного ящика** относится к связности или сцеплению модуля?
45. Какие компьютеры называются **серверами**?
46. Какие виды связей возможны в структурах данных?
47. Что собой представляет дисциплина работы с памятью LIFO?
48. Каким показателем характеризуется время сортировки массива методом пузырька?
49. Перечислите основные требования к алгоритму.
50. Может ли один компьютер одновременно использоваться и как рабочая станция, и как сервер?

Лекция 8

1. Какой алгоритм называется детерминированным?
2. Простой перебор вариантов при решении оптимизационной задачи обеспечивает выбор оптимального из них?
3. Когда останавливается решение при численном интегрировании?
4. Какие функции возлагаются на компьютер в пользовательском интерфейсе?
5. Для чего используется алгоритм Дейкстры?
6. Какой алгоритм называется недетерминированным?
7. Дайте определение **метода**.
8. «Метод ближайшей вершины» при решении «задачи коммивояжера» обеспечивает получение оптимального решения?
9. Когда останавливается решение при численном нахождении корней уравнения?
10. Для какого вида графа можно использовать алгоритм Дейкстры?
11. Что является более общим: метод или алгоритм?
12. Какой метод использует такое понятие как «текущий оптимальный маршрут»?

13. Какие периферийные устройства компьютера использует пользовательский интерфейс?
14. Чем определяется общее число вариантов в решении «задачи коммивояжера»?
15. Каким свойством должны обладать частные решения в методе «разделяй и властвуй»?
16. Чем отличается метод от алгоритма?
17. В чем суть метода «разделяй и властвуй»?
18. На какие фигуры разбивается площадь под графиком функции при численном интегрировании?
19. Какие алгоритмы называются **жадными**?
20. Приведите пример языка (языков), используемых для описания алгоритмов?
21. В чем специфика метода ветвей и границ?
22. В русле какого метода лежит алгоритм Дейкстры?
23. Какие задачи решает линейное программирование?
24. В чем задача диалога пользователя с системой?
25. Что предполагает концепция DWIM?
26. В каком случае меняется значение «текущего оптимального маршрута» в методе «ветвей и границ»?
27. Как определяется место корня при компьютерном решении уравнений?
28. Для какого типа графа можно использовать алгоритм Дейкстры?
29. В чем суть метода динамического программирования?
30. Какие функции возлагаются на пользователя в пользовательском интерфейсе?
31. Когда останавливается вычисление текущего маршрута в методе «ветвей и границ»?
32. Каким должен быть интерфейс пользователя с компьютером, чтобы можно было реализовать непосредственное манипулирование?
33. В каком случае количество маршрутов в методе простого перебора можно сократить в два раза?
34. В чем суть метода наискорейшего спуска?
35. Что такое **диалог**?
36. Что такое «реальный масштаб времени»?
37. Как расшифровывается WIMP?
38. По каким критериям может определяться оптимальный маршрут в «задаче коммивояжера»?
39. В чем суть метода последовательных приближений?

40. Каким был исторически первый интерфейс пользователя с персональным компьютером?
41. В чем заключается непосредственное манипулирование пользователем с компьютером?
42. За счет чего происходит сокращение анализируемых маршрутов на каждом шаге при динамическом программировании?
43. Дайте определение **пользовательскому интерфейсу**.
44. Приведите примеры современного использования «задачи коммивояжера»?
45. Какие периферийные устройства компьютера используются в пользовательском интерфейсе?
46. Как расшифровывается GUI?
47. Что такое **модальные** окна?
48. Что такое «симплекс-метод»?
49. Достоинства графического интерфейса?
50. Как соотносится во времени проектирование пользовательского интерфейса с остальными этапами стадии проектирования?

Лекция 9

1. Чем отличается командный язык общения с компьютером от естественного?
2. Какие данные должны отделяться друг от друга при выводе информации?
3. Как называется принцип проектирования пользовательского интерфейса, предполагающий, что поведение системы должно быть прогнозируемым?
4. Какой принцип должен быть положен в основу формирования документации программной системы?
5. Как должна распределяться информация по уровням справочной системы?
6. Какая форма ввода информации наиболее проста в освоении пользователем?
7. При быстрой смене выводимой информации какая форма: текстовая или графическая, - более предпочтительна?
8. Зачем целесообразно вести протокол всех действий пользователя в интерфейсе?
9. Какое время реакции системы на действия пользователя считается для него комфортным?
10. Какие виды точек входа должны быть в справочной системе?
11. Какая форма ввода информации наиболее сложна в освоении пользователем?
12. Каким образом можно отделять друг от друга разные виды информации при выводе на экране?
13. Как называется протокол всех действий пользователя в интерфейсе?

14. Что стандартно входит в справочную систему программного приложения?
15. Какие точки входа в справочную систему могут быть в множественном числе?
16. Какая форма ввода информации наиболее проста в программной реализации?
17. Какую информацию: текстовую или графическую, - легче анализировать в динамике?
18. Сколько стандартно включает разделов документация программной системы?
19. Какими должны быть сообщения системы пользователю?
20. Какой вид графа должна представлять справочная структура ПС?
21. Какая форма ввода информации наиболее сложна в программной реализации?
22. На основании какой информации проектируется пользовательский интерфейс?
23. Какая форма ввода информации минимизирует количество ошибок?
24. Что целесообразно включать в сообщение ПС об ошибке?
25. Что собой представляет компонент history справочной системы?
26. Какой раздел документации программной системы предназначен для потенциальных покупателей?
27. Кто должен помогать разработчикам программной системы в проектировании пользовательского интерфейса?
28. В чем состоит назначение первого раздела документация программной системы?
29. Какие специалисты должны формировать справочную систему программного приложения?
30. Какие средства должен содержать пользовательский интерфейс, если задержка реакции системы на действия пользователя слишком велика?
31. Какие специалисты должны заниматься проектированием пользовательских интерфейсов?
32. Какие есть этапы в разработке пользовательского интерфейса?
33. В каком разделе документация программной системы приведена инструкция по установке системы?
34. Зачем рекомендуется объединять сообщения системы в один модуль?
35. Приведите несколько форматов (не менее трех) справочных систем.
36. Какие цвета являются тревожными?
37. Какие термины и понятия должны использоваться в интерфейсе? Почему?
38. В каком разделе документация программной системы приведена инструкция по начальным шагам работы с системой?
39. Каким должен быть стиль общения программной системы с пользователем?

40. Разработка какого компонента ПС производится на основе принципа User-centered design?
41. Какие две проблемы надо решить, проектируя взаимодействие пользователя с программной системой?
42. На основе какой математической структуры создается меню?
43. Какие цвета являются спокойными?
44. В каком разделе документация программной системы приведена подробная инструкция по работе с системой?
45. Зачем разработчику интерфейса надо знать культурные особенности страны, где программная система будет эксплуатироваться?
46. Каким должен быть интерфейс пользователя с компьютером, чтобы можно было реализовать непосредственное манипулирование?
47. Что должно включать контекстное меню?
48. Почему не рекомендуется цветовое кодирование числовых значений?
49. Какие разделы документация программной системы предназначены для системного администратора?
50. Какой должна быть структура справочной системы? Почему именно такой?

Лекция 10

1. Дайте определения **класса**.
2. Какие модули программной системы кодируются в первую очередь при нисходящей разработке?
3. Что первично: объект или класс?
4. Назначение диаграмм классов, формируемых на втором этапе?
5. Кому отдается предпочтение при написании кода: писателю или читателю? Почему?
6. Как формируются классы?
7. Что такое **атрибут** класса?
8. Какие есть виды стереотипов поведения у объектов?
9. Что собой представляет пиктограмма класса-сущности? Какие секции в нее входят?
10. Зачем в листинг вводятся пустые строки?
11. Как по-другому называются операции класса?
12. Какие есть варианты видимости у атрибутов и методов класса?
13. Как задаются ограничения у класса?
14. Как расшифровывается UML?
15. Каким требованиям должен отвечать листинг программной системы?

16. На какой стадии жизненного цикла происходит определение рабочего окружения системы и разработка моделей ее использования при объектном подходе проектирования?
17. В чем назначения диаграмм вариантов использования при объектном подходе проектирования программной системы?
18. Что такое **объект**?
19. Для чего используется язык UML?
20. При каком методе разработки: восходящем или нисходящем, - максимально тестируются модули ядра системы?
21. Что такое принцип KISS?
22. Что описывает каждый прецедент использования при объектном подходе проектирования программной системы?
23. Что означает каждый пакет на диаграмме пакетов языка UML?
24. Назначение диаграмм классов, формируемых на третьем этапе?
25. Дайте определение процессу **кодирования**.
26. Какая диаграмма языка UML направлена на структуризацию проектируемой программной системы?
27. Что такое принцип EIBTI?
28. Что собой представляют **отношения** между классами?
29. Какой класс называется **супертипом**?
30. Что означает овал на прецеденте использования?
31. Какой процент листинга рекомендуется представлять пустыми строками?
32. Какое отношение между классами называется **ассоциацией**?
33. Какой класс называется **подтипом**?
34. Приведите пример отношения обобщения (наследования).
35. Что такое **таксономия**?
36. Для чего используется диаграмма последовательностей?
37. Что или кто может быть актором в прецеденте использования?
38. Как отображается время на диаграмме последовательностей?
39. Что отображают на диаграмме последовательностей горизонтальные линии?
40. Когда создается **таксономия** при объектно-ориентированном проектировании ПС?
41. Какая диаграмма описывает взаимодействие объектов разных классов в рамках конкретного сценария?

42. Какая диаграмма UML используется для описания алгоритмов, реализуемых методами класса?
43. Приведите пример отношения агрегации (целое-часть).
44. Какая диаграмма отображает проектируемую систему на физическом уровне?
45. Дайте определение **компонента** в рамках объектного подхода к проектированию программной системы.
46. Какие существуют виды связей между классами?
47. Сколько этапов по созданию диаграмм классов выделяют в рамках объектного подхода к проектированию программной системы?
48. Назначение диаграмм классов, формируемых на первом этапе?
49. Какая диаграмма стоит на последнем шаге проектирования программной системы при объектном подходе?
50. Какими видами сообщений обмениваются объекты классов проектируемой системы?

Лекция 11

1. Что такое **защитное программирование**?
2. Перечислите основные подходы к тестированию программных систем.
3. Сколько существует правил хорошего тона при поставке программного продукта?
4. Почему защитное программирование предполагает запрет на сравнение вещественных чисел в логических выражениях?
5. С каким «ящиком» ассоциируется функциональное тестирование?
6. Что собой представляет процесс разработки программной системы?
7. Почему защитное программирование предполагает запрет на деление больших чисел на малые?
8. С каким «ящиком» ассоциируется структурное тестирование?
9. Какой стандарт регламентирует номенклатуру технологических процессов разработки программной системы?
10. Какие языковые конструкции в языке Python предназначены для обработки исключительных ситуаций?
11. Какие технологически процессы продолжаются в течение всего жизненного цикла программной системы?
12. В каком подходе к тестированию внутренняя структура программной системы не рассматривается, а только анализируются системные характеристики?

13. Сколько технологических процессов предполагает современный стандарт разработки программной системы?
14. В чем недостатки парного программирования?
15. В каком подходе возможно исчерпывающее тестирование программной системы?
16. Какая группа технологических процессов разработки программной системы является самой многочисленной?
17. Почему значительно снижается количество ошибок при парном программировании?
18. Что собой представляет стандарт тестирования C1?
19. Какие варианты доставки программных продуктов покупателю существуют в настоящее время?
20. За счет чего растет квалификация программистов при парном программировании?
21. Какой процент ошибок в среднем обнаруживается при использовании стандарта C1?
22. При каких вариантах доставки программного продукта не предполагается печатная документация?
23. В каких программных системах вводятся жесткие ограничения на циклические конструкции?
24. Какой подход используется при интеграционном тестировании программной системы? Почему?
25. Какой вариант доставки программного продукта включает рекламные материалы?
26. Что такое стиль программирования?
27. Какие ошибки обнаруживаются на этапе компиляции программных модулей?
28. Почему в системах реального времени жесткие ограничения на циклические конструкции?
29. При каких вариантах доставки программного продукта печатная документация обязательна?
30. Дайте определение понятию тестирование.
31. Укажите классификацию программных ошибок по характеру проявления.
32. На какой стадии стоимость исправления ошибки в программной системе минимальна?
33. В чем заключается первое правило хорошего тона при доставке программного продукта?
34. Какие ошибки обнаруживаются на этапе компиляции программных модулей?
35. На какой стадии стоимость исправления ошибки в программной системе минимальна?
36. Что гласит принцип Биледи?

37. На какой стадии стоимость исправления ошибки в программной системе максимальна?
38. На какой стадии жизненного цикла стоимость исправления ошибки в программной системе принимается за единичную?
39. Сколько классов задач решается в ходе сопровождения программной системы?
40. Дайте определение понятию тест.
41. Основные характеристики теста
42. В чем заключается второе правило хорошего тона при доставке программного продукта?
43. Что характеризует детективность теста?
44. С какой стадии жизненного цикла специалисты по тестированию подключаются к разработке программной системы?
45. Что предполагает сопровождение программного продукта?
46. В чем заключается альфа-тестирование программной системы?
47. Что представляет собой первый класс задач, решаемых в ходе сопровождения программной системы?
48. В чем заключается бета-тестирование программной системы?
49. Для каких программных систем проводится бета-тестирование?
50. Дайте определению понятию отладка программной системы?

Лекция 12

1. Какие обязанности относятся к менеджеру программного проекта по методологии MSF?
2. Что гласит главные тезис специалистов по управлению проектами?
3. Что такое **гедонизм**? Как его может использовать менеджер при работе над программным проектом?
4. В каком случае программная система реализует **автоматизированное** управление объектом?
5. В чем заключаются функции администратора БД в группе разработчиков по модели MSF?
6. Дайте определения понятию **управление**.
7. Перечислите задачи менеджеров, руководящих программными проектами?
8. Что такое когнитивная теория? Как ее может использовать менеджер при работе над программным проектом?
9. Что является объектом методологии MSF?
10. Сколько этапов проходит каждая итерация программного проекта по методологии MSF?
11. В чем состоит задача управления программным проектом?

12. Что такое потребность? Виды существуют групп потребностей у человека?
13. Что предполагает показатель **наблюдаемость** программного проекта?
14. Какие виды объектов могут быть у методологий в программной инженерии?
15. С какого этапа начинается каждая итерация программного проекта по методологии MSF?
16. В чем отличие программных проектов от других видов проектов?
17. Какие потребности должен обеспечивать менеджер для повышения эффективности управления программным проектом?
18. Что предполагает показатель **управляемость** проекта?
19. Как называются документы, в которых изложены основы методологии MSF?
20. Какие задачи решаются на этапе планирования каждой итерации программного проекта по методологии MSF?
21. В чем была причина неудачного управления программными проектами в шестидесятые годы прошлого столетия?
22. Какие потребности могут проявляться у человека только в социуме?
23. Какие показатели существуют у проекта?
24. Сколько принципов лежит в основе методологии MSF?
25. В чем заключаются функции разработчика проекта в группе разработчиков по модели MSF?
26. Что могут представлять возмущающие воздействия для программного проекта?
27. Что повышает квалификацию членов группы разработчиков?
28. Дайте определение понятию программный **проект**.
29. В чем заключаются функции бизнес-аналитика в группе разработчиков по модели MSF?
30. Какие задачи решаются на этапе стабилизации каждой итерации программного проекта по методологии MSF?
31. Что представляют собой ограничения любого проекта?
32. Какие потребности человека должны удовлетворяться государством?
33. На какие ограничения проекта накладываются ограничения сверху?
34. В чем заключаются функции менеджера проекта в группе разработчиков по модели MSF?
35. Как называются аппаратно-программные средства поддержки разработки программного проекта?
36. Дайте определение понятию **проект**.

37. Что собой представляет **управление** проектами?
38. Какие потребности человека должны удовлетворяться на месте его работы?
39. В чем заключаются функции архитектора в группе разработчиков по модели MSF?
40. Что является «сырьем» для технологических операций процесса разработки программного проекта?
41. Что находится на входе проекта как черного ящика?
42. Что собой представляет **управление конфигурациями** для программного проекта?
43. Почему все программные проекты уникальны?
44. В чем заключаются функции тестировщика в группе разработчиков по модели MSF?
45. Кто готовит документацию по каждой версии программного продукта по модели MSF?
46. Чем отличается автоматическое управление от автоматизированного?
47. Какие задачи решаются в ходе управления?
48. В чем заключаются функции релиз-менеджера в группе разработчиков по модели MSF?
49. Как называются контрольные точки проекта в терминологии MSF?
50. Для кого важны такие характеристики программного проекта как **наблюдаемость** и **управляемость**?

Лекция 13

1. Каким рискам уделяется наибольшее внимание при разработке ПО?
2. Какое количество ошибок допускает надежное ПО?
3. Как должно формироваться рабочее место программистов?
4. Сколько этапов насчитывает одна итерация в процессе управления готовностью?
5. В чем заключается работа с риском, если его нельзя предотвратить?
6. Кто в группе разработчиков ориентирован на сплочение группы в единую команду, устранение конфликтных ситуаций?
7. Какое количество ошибок допускает особо надежное ПО?
8. Какая работа проводится на третьем этапе управления готовностью в методологии MSF?
9. Какой стандарт определяет характеристики качества?
10. Сколько времени разработчик проводит в общении с другими членами команды?
11. Что собой представляет первый этап в управлении готовности?
12. Что повышает квалификацию членов группы разработчиков?
13. Какие проблемы решает сплоченность группы разработчиков программного обеспечения?
14. Какие способы рекомендуется использовать для сплочения группы разработчиков программного обеспечения?

15. Как распределяется время программиста по видам деятельности?
16. Что собой представляет второй этап в управлении готовности?
17. Что такое **качество** как общепринятое понятие?
18. Какие факторы учитываются при приеме разработчика ПО на работу?
19. Что входит в первую группу факторов, влияющих на качество программного продукта?
20. Что собой представляет третий этап в управлении готовности?
21. Что собой представляет «рыбий скелет» К. Исикавы?
22. Какие виды ущерба рассматриваются при управлении рисками?
23. Как распределяются факторы, влияющие на качество программного продукту по фазам жизненного цикла?
24. Какие виды повышения квалификации разработчиков предусматривает управление готовностью в методологии MSF?
25. Какие виды метрик вводятся для оценки качества программного продукта?
26. Какие факторы влияют на подбор специалистов по разработке программного обеспечения?
27. Что собой представляет четвертый этап в управлении готовности?
28. Какая характеристика качества определяется как отношение полезного эффекта к произведенным затратам?
29. Какие личные характеристики желательны для разработчика как члена коллектива?
30. Что собой представляют бизнес-риски для программного проекта?
31. Кто должен заниматься подбором персонала для работы над программным проектом?
32. Что такое качество программной системы?
33. Какой процент рабочего времени разработчик ПО тратит на самостоятельную работу?
34. Почему процесс управления готовностью должен протекать в течение всего программного проекта?
35. В каком случае общение в группе разработчиков программного обеспечения протекает более свободно?
36. Что такое LOC? Для оценки какой характеристики качества ПО используется LOC?
37. Что входит во вторую группу факторов, влияющих на качество программного продукта?
38. В каком случае отказываются от обеспечения высокой надежности ПО?
39. Какие форс-мажорные ситуации могут возникнуть в группе разработчиков?
40. Что должно предусматривать рабочая среда для того, чтобы разработчики могли заняться мандатной или созываемой деятельностью?

41. Что может представлять собой технологический риск?
42. Что собой представляет внутренне качество ПО?
43. Дайте определение надежности ПО.
44. Какова наиболее оптимальная численность команды разработчиков программных продуктов?
45. Что понимается под **готовностью** в методологии MSF?
46. К какому типу риска относится смена руководства компании-заказчика?
47. Перечислите основные подходы к повышению надежности программных средств.
48. Что расширяет профессиональную эрудицию разработчика ПО?
49. В чем проявляются недостатки слишком большой численности команды разработчиков программных продуктов?
50. На каком этапе управления готовности определяются компетенции, сценарии деятельности каждого разработчика?

Лекция 14

1. Почему фирмы SCL боятся брать на себя окончательное решения по вопросу сертификации программного продукта?
2. Какое основное требование предъявляется к требованиям, формируемым к разрабатываемому программному продукту?
3. С чего начинается обеспечение качества программного продукта?
4. Что говорится в рекомендациях по управлению программными проектами по поводу интерфейсов?
5. Какую цель преследует стандартизация?
6. Зачем фирмы SCL дополняют программные продукты программными закладками?
7. Какой вариант тестирования представляет собой сертификация на основе сбора информации программными закладками?
8. Сколько есть уровней зрелости фирм в области программной инженерии?
9. Какой режим доступа к информации должен обеспечиваться для разработчиков согласно рекомендациям по управлению программными проектами?
10. Чем доказывается качество программного продукта?
11. Что является признаком хорошего управления программным проектом?
12. Перечислите основные функции основополагающих стандартов.
13. Как определяется принадлежность фирмы – производителя программной продукции – к тому или иному уровню по развитию персонала?

14. Дайте определение **сертификации** программного продукта.
15. Абсолютное число оставшихся ошибок в программной системе может дать характеристику надежности продукта?
16. На основании чего должна оформлять документация программного проекта?
17. Каким должен быть доступ к информации о программном проекте у его разработчиков?
18. Какая шкала используется для оценки уровня подготовки персонала по модели Р-СММ?
19. В каких стандартах рассматриваются вопросы безопасности продукции?
20. В какой области программной инженерии используется метод Миллса?
21. Кто должен осуществлять контроль качества программной продукции?
22. Какие процессы являются итерационными в работе менеджеров программного проекта?
23. Какое минимальное количество баллов по каждому показателю обеспечивает нахождение фирмы – производителя программной продукции – на определенном уровне модели Р-СММ?
24. Какие стандарты определяют методы контроля продукции?
25. В какой области программной инженерии используется метод Руднера?
26. Каким образом фирмы по сертификации ПО чаще всего собирают информацию о надежности конкретного программного продукта?
27. Как эффективней всего представлять рабочие графики программного проекта?
28. Какое максимальное количество баллов можно набрать по каждому показателю в модели Р-СММ?
29. Что должны устанавливать стандарты на процессы?
30. В каком методе определения числа оставшихся в программной системе ошибок используется несколько бригад?
31. Как называются фирмы по сертификации программной продукции?
32. Какие действия обязательны при работе с ошибками в соответствии с рекомендациями по управлению программными проектами?
33. На каких уровнях модели Р-СММ результат работы зависит от конкретных разработчиков?
34. Какие две функции должна обеспечивать для потребителя стандартизация?
35. Что собой представляет число m_{12} в методе Руднера?
36. Сертификации должны подвергаться только программные продукты на заказ, только корпоративное ПО или оба вида продукции?

37. Какое требование накладывается на разработчика при создании и модификации документации программной продукции?
38. Какой уровень зрелости персонала присущ наибольшему количеству фирм в программной инженерии?
39. Все фирмы в области программной инженерии должны обязательно соблюдать международные стандарты или могут формировать свои внутрифирменные стандарты?
40. Какие стандарты могут служить основой для разработки других стандартов?
41. Чем занимаются фирмы SCL?
42. Что находится по формуле $M = m_1 * m_2/m_{12}$?
43. Что представляет показатель **correct first-try rate**?
44. На каком этапе жизненного цикла ПО проводится сертификация?
45. Что может быть объектом стандартизации?
46. Сколько вариантов как минимум для решения конкретной задачи должно предлагать профессиональное руководство программным проектом?
47. Что собой представляют «меченые рыбки» Миллса?
48. Какая фирма ввела внутрифирменную метрику «процент пользователей, которым удалось без обращения к руководству по ПС выполнить некоторое осмысленное действие»?
49. Что может быть результатом стандартизации?
50. Какой стандарт перечисляет сущность каждого атрибута характеристики надежности и наборы метрик для его оценки?

Лекция 15

1. Какие стандарты регламентируют правила общения с клиентами фирм, занимающихся разработкой ПО?
2. Какая организация: МЭК или ИСО, - была образована раньше?
3. Какие стандарты регламентируют контроль качества и подтверждения соответствия продукции?
4. В каких двух вариантах может быть представлен стандарт?
5. В какую группу стандартов входят стандарты, регламентирующие правила хранения проектно-аналитической документации, правила кодирования имен файлов?
6. Какие стандарты регламентируют правила общения сотрудников фирм, занимающихся разработкой ПО?
7. Какие вопросы в области стандартизации, рассматриваются МЭК и ИСО совместно?
8. Какие виды регионов предусматриваются региональными стандартами?

9. В чем заключается основная задача стандартизации в области разработки ПО?
10. В какую группу стандартов входят стандарты, регламентирующие правила описания бизнес-процессов программными средствами?
11. Как расшифровывается ЕСПД?
12. Какой орган в России должен заниматься организацией подготовки и повышения квалификации специалистов в области стандартизации?
13. Как называется совместное образование МЭК и ИСО?
14. К какому виду стандартов относятся стандарты, обязательные или рекомендуемые в пределах Калининградской области?
15. Дайте определение стандартизации в области информационных технологий.
16. В какую группу стандартов входят стандарты, регламентирующие формирование наименований?
17. Что такое **сертификация**?
18. Как называется чисто европейская организация по стандартизации?
19. Какого уровня стандарты делятся на несколько подуровней?
20. Каким образом происходит «рождение» новых стандартов?
21. В какую группу стандартов входят стандарты, регламентирующие структуру директорий разработки?
22. Что такое **метрология**?
23. Разработкой каких стандартов занимается ANSI?
24. Чем отличается стандарт **де-факто** от стандарта **де-юре**?
25. Кто является главным государственным инспектором в РФ по надзору за соблюдением стандартов?
26. В какую группу стандартов входят стандарты, регламентирующие документирование исходного кода?
27. Какая организация в РФ уполномочена на сотрудничество с национальными органами других стран по вопросам стандартизации?
28. Как называется головная организация по разработке и утверждению стандартов в России?
29. Дайте определение **стандарта**.
30. Какая организация является лидером в разработке стандартов в области ПО?
31. Что входит в группу общего программного обеспечения?

32. Какая организация в РФ представляет Россию в международных организациях, занимающихся стандартизацией?
33. Перечислите международные организации по разработке стандартов в области информационных технологий.
34. В каком случае стандарт может играть отрицательную роль?
35. Кто уполномочен в РФ определять порядок и правила проведения работ по стандартизации, сертификации и метрологии?
36. Какое программное обеспечение называется прикладным?
37. Какая организация в РФ утверждает государственные стандарты?
38. Как называется первый стандарт по разработке ПО, принятый еще в СССР?
39. Какую процедуру должны периодически проходить стандарты?
40. Для каких разработчиков программного обеспечения стандартами предусматриваются свои рабочие места?
41. Какое программное обеспечение называется системным?
42. На каких правах Россия входит в ИСО?
43. Какие стандарты устанавливают методы способы, приемы, методики выполнения испытаний?
44. Какая разновидность стандарта используется путем сравнения?
45. Что собой предусматривает рабочее место разработчика ПО в соответствии со стандартами?
46. Какие стандарты регламентируют вспомогательные процессы программного процесса?
47. Какие предметные области относятся к компетенции МЭК?
48. Цель стандартов на процессы?
49. Что должен в себя включать стандарт как документ?
50. В какую группу стандартов входят стандарты, регламентирующие применение методик структурного анализа или методов объектно-ориентированного применения?

3.1.2 Критерии оценки опросов:

В ходе опроса каждый студент должен ответить на пять вопросов, последняя цифра номера которых совпадает с последней цифрой номера студента в общем списке потока, выложенном в ЭИОС. Т.е. студенты с номерами 3, 13, 23, 33, 43 должны ответить на вопросы с такими же номерами. На ответы студентам выделяется 1,5 часа.

Ответ на каждый вопрос оценивается в 1 балл по системе «правильный/неправильный», т.е. минимально студент может получить за один опрос 0 баллов, максимально – 5 баллов.

После окончания лекционного курса к перечню полученных оценок добавляется оценка за курсовую работу, после чего определяется среднее значение оценок, которое округляется до ближайшего целого. Если полученное значение не менее 3 «удовлетворительно», то эта оценка предлагается студенту в качестве итоговой по дисциплине.

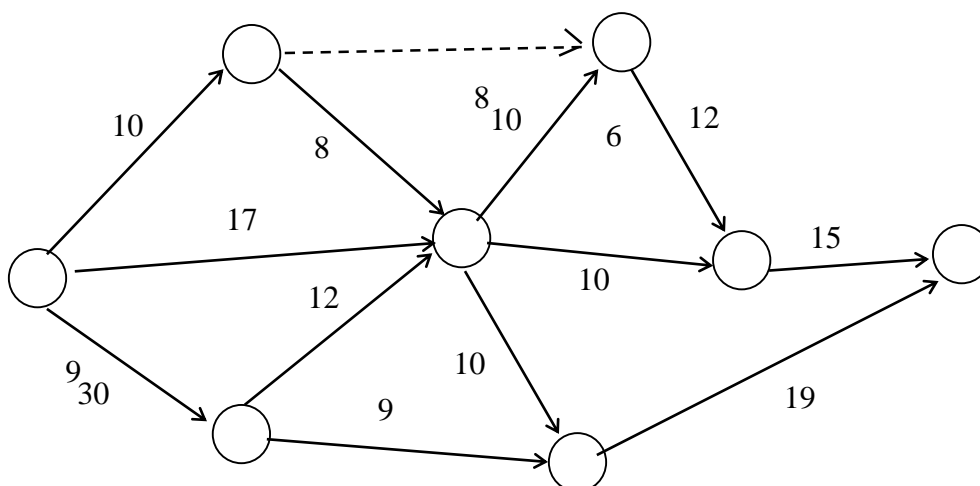
3.2 Задания и контрольные вопросы по лабораторным работам

3.2.1 Задания и контрольные вопросы

Лабораторная работа № 1 Планирование разработки программной системы

Пример задания

Для сети работ:



1. Преобразовать сеть «дуга-работа» в сеть «узел-работа»;
2. Рассчитать все временные параметры работ.
3. Выделить в сети критический путь.
4. Произвести оптимизацию календарного плана:
 - а) по максимизации запасов у работ;
 - б) по минимизации количества бригад разработчиков.
5. Построить в масштабе календарные планы выполнения работ в двух рассмотренных выше вариантах оптимизации.

Контрольные вопросы по лабораторной работе:

1. В чём достоинство сети "узел-работа"?

2. Как определить вершины, лежащие на критическом пути, если плановый срок окончания проекта $T > EF_{\text{окончания}}$?
3. Какого типа матрицы можно использовать для описания сети "узел-работа"?
4. Какой граф называется сетью?
5. Почему множество работ по созданию ИС является частично-упорядоченным?
6. Какие разработчики назначаются в бригаду «критического пути»?

Лабораторная работа № 2 Использование метода экспертных оценок при разработке программных систем

Пример задания

Произвести обработку экспертного опроса на основе данных, выбранных из таблицы с заданиями, и сделать вывод о согласованности мнений экспертов.

№ задач	Эксперты								
	1	2	3	4	5	6	7	8	9
2	2	1	1	2	1	2	2	1	2
3	5	5	6	5	6	6	5	6	6
4	4	4	2	2	2	4	4	4	5
5	4	4	5	5	5	6	5	6	4
6	5	5	6	5	5	5	5	6	5
8	3	2	3	2	3	3	2	4	2
9	3	3	4	4	4	2	3	4	2
10	3	3	4	3	4	4	3	5	3
12	4	2	5	5	1	5	5	3	2
13	2	1	2	2	2	2	1	3	1
14	2	4	5	5	5	2	5	6	2
15	2	2	3	2	2	3	3	2	2

Контрольные вопросы по лабораторной работе:

1. В каких случаях используется экспертный опрос?
2. Кто может быть экспертом?
3. Каким может быть максимальное значение коэффициента конкордации?
4. Для чего вводятся весовые коэффициенты при экспертном опросе?
5. Чему равна сумма весов нормированных коэффициентов?

Лабораторная работа № 3 Формирование информационной модели предметной области

Пример задания

Необходимо построить информационную модель в соответствии с заданной предметной областью и на основе заданной нотации.

Предметная область	Вид нотации
Студенческая сессия	Silverrun

Контрольные вопросы по лабораторной работе:

1. На какой стадии жизненного цикла программной системы формируется информационная модель предметной области?
2. Какая связь называется идентифицирующей?
3. Может ли быть сущность без атрибутов?
4. Что означает связь M:M?
5. Может ли быть сущностью абстрактный объект?

Лабораторная работа №4 Средства формирования поведенческой модели

Пример задания

Сформировать поведенческую модель - алгоритм распознавания цепочки символов вида «Дуга с числовыми значениями координат и параметров **start** и **extent**» - в виде граф-схемы автомата.

Контрольные вопросы по лабораторной работе:

1. На какой стадии жизненного цикла программных систем формируются ДПС?
2. На основании чего определяется следующее состояние автомата?
3. Какими формальными средствами описывается переходная функция конечного автомата?
4. Может ли быть у автомата несколько начальных вершин?
5. Может ли быть у автомата несколько конечных вершин?

Лабораторная работа №5 Алгоритмизация «задачи коммивояжера»

Пример задания

В таблице представлены на двумерной плоскости координаты узлов некоторой транспортной системы, для которой требуется минимизировать длину маршрута в соответствии с «задачей коммивояжера» на основе заданных алгоритмов.

Координаты узлов

X ₁ ,Y ₁	X ₂ ,Y ₂	X ₃ ,Y ₃	X ₄ ,Y ₄	X ₅ ,Y ₅	X ₆ ,Y ₆	X ₇ ,Y ₇	X ₈ ,Y ₈	X ₉ ,Y ₉	X ₁₀ ,Y ₁₀
56,34	9,60	60,89	13,77	52,35	55,70	39,88	90,54	57,21	11,95

Контрольные вопросы по лабораторной работе:

1. Используемые методы можно использовать для расчета оптимальной топологии ЛВС в виде кольца. А как на основе полученного результата определить оптимальную топологию вида «общая шина»?
2. В каком случае количество вариантов метода ветвей и границ определяется выражением $(n-1)!$, а в каком $(n-1)!/2$?
3. Почему матрицы расстояний в проводимых расчетах являются треугольными?
4. Какие узлы исходной сети являются граничными, а какие внутренними?
5. Какой граф называется деревом?

Лабораторная работа № 6 Разработка спецификаций программного обеспечения

Пример задания

Программируемая задача: «Расчет параметров сетевого графика с графической иллюстрацией и указанием критического пути».

Задание: Произвести разбивку заданной программной системы на модули и определить корректность данного разбиения.

Контрольные вопросы по лабораторной работе:

1. Что такое спецификация?
2. В чем достоинства модульного подхода при разработке программ?
3. Какая структура программы: горизонтальная или вертикальная (рис. 6.7) - чаще реализуется на практике, и почему?
4. Какое условие должно выполняться для корректного разбиения программы на модули?
5. В каком случае имеет место логическая связность модулей?

Лабораторная работа № 7 Изучение стандартов

Пример задания

Для стандарта IEEE 1074-1995 создать отчет объемом 3, 4 страницы, содержащий краткое описание стандарта и его назначение по всем рассмотренным в разделе «Теоретическое введение» метод. указания по лабораторным работам классификациям.

Контрольные вопросы по лабораторной работе:

1. Для чего нужна стандартизация?
2. Какие международные организации занимаются разработкой стандартов в области создания ПО?
3. Как распределяются стандарты по областям применения?
4. В каком случае применение стандартов является обязательным?
5. Что может явиться объектом стандартизации?

Лабораторная работа № 8 Управление рисками для программной системы

Пример задания

Для уменьшения риска несанкционированного доступа к документом разработки программной системы создать парольную защиту вида:

Простой пароль длиной не менее 6 символов, содержащий не менее 1 цифры и буквы в разных регистрах. Трехкратный неправильный ввод приводит к предупреждению на экране, что за компьютером злоумышленник, и выдаче звукового сигнала длиной 15 сек. После шести запусков программа требует и обязательно производит смену пароля. Пароль хранится в закодированном виде.

Контрольные вопросы по лабораторной работе:

1. Для чего применяются пароли?
2. Какие рекомендации целесообразно соблюдать при использовании паролей?
3. Как можно реализовать разграничение доступа к программе?
4. Чем отличается логическая бомба от вируса?
5. Что такое НСД?

Лабораторная работа № 9 Тестирование программного обеспечения

Пример задания

Для программируемой задачи «Вводятся координаты левого верхнего и правого нижнего углов двух прямоугольников. Определить площадь пересечения прямоугольников»:

1. Разработать блок-схему программы.
2. Сформировать граф потока управления программы и определить его цикломатическое число

3. Разработать программный проект, реализующий задачу. Проект должен графически иллюстрировать реализуемый вариант решения задачи. Все графические построения реализуются на поле размером 600x600 пикселей. Центр координатной системы находится в центре поля, диапазон значений по каждой оси – от –200 до 200.
4. Разработать для каждой ветви графа потока управления свой тест с графической иллюстрацией. Произвести проверку всех возможных решений, уделяя особое внимание граничным значениям входных данных.

Контрольные вопросы по лабораторной работе:

1. Какая разновидность тестирования изучается в данной лабораторной работе?
2. Что включает в себя сценарий тестирования?
3. Какую технологию программирования предполагает тестирование ветвей?
4. Какие элементы блок-схемы не включаются в граф потока управления?
5. Чему равно минимальное число тестов при тестировании ветвей?

3.2.2 Критерии оценки лабораторной работы:

Оценка результатов выполнения задания по каждой лабораторной работе производится при представлении студентом отчета по лабораторной работе, демонстрации преподавателю исполнения разработанной им программы и на основании ответов студента по тематике лабораторной работы.

- оценка «зачтено» выставляется обучающемуся, если он демонстрирует способность к полной самостоятельности (допускаются консультации с преподавателем по сопутствующим вопросам) в выборе способа решения неизвестных или нестандартных заданий в рамках учебной дисциплины с использованием знаний, умений и навыков, полученных как в ходе освоения данной учебной дисциплины, так и смежных дисциплин;

- оценка «незачтено» выставляется, если выявляется неспособность обучаемого самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения, отсутствие самостоятельности в применении умения к использованию методов освоения учебной дисциплины и неспособность самостоятельно проявить навык повторения решения поставленной задачи по стандартному образцу, что свидетельствует об отсутствии сформированной компетенции.

3.3 Тестовые задания

3.3.1 Тестовые задания представлены в Приложении № 1.

3.3.2 Целью тестирования является закрепление, углубление и систематизация знаний студентов, полученных на лекциях и в процессе самостоятельной работы; проведение тестирования позволяет ускорить контроль за усвоением знаний и объективизировать процедуру оценки знаний студента. Оценка определяется количеством допущенных в ответах ошибок.

Оценка «5» («отлично») ставится, если студент ответил правильно на 81% - 100% тестовых заданий.

Оценка «4» («хорошо») ставится, если студент ответил правильно на 61% - 80% тестовых заданий.

Оценка «3» («удовлетворительно») ставится, если студент ответил правильно на 41% - 60% тестовых заданий.

Оценка «2» («неудовлетворительно») ставится, если студент ответил правильно не более, чем на 40% тестовых заданий.

4 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

4.1 Задания и контрольные вопросы по курсовой работе

4.1.1 Пример задания

Создать информационную систему «Товарный магазин» на базе таблицы **Список товаров** с возможностью ее редактирования, а также проверочные таблицы по товарам, поставщикам и покупателям. Каждый день создается по каждому товару фиксируется поступление товара и его покупка. Должна быть возможность просмотра продаж по дням и по товарам с расчетом итоговых показателей по дням, месяцам, диапазонам дат, поставщикам и покупателям.

4.1.2 Контрольные вопросы к защите курсовой работы

1. Какие таблицы включает база данных разработанного приложения?
2. Как называется язык, на котором сформированы запросы к базе данных?
3. Как обеспечивается разделение доступа к данным в курсовой работе?
4. Какие разделы должны входить в справочную систему программного приложения?
5. Какие вопросы должны отражаться в руководстве пользователя?

4.1.3 Оценочные средства курсовой работы

По результатам защиты курсовой работы (студент представляет результаты программирования в виде пояснительной записки заданной структуры, демонстрирует работу программы в различных режимах и отвечает на вопросы преподавателей) выставляется экспертная оценка («отлично», «хорошо», «удовлетворительно» или «неудовлетворительно»), которая учитывается при итоговой аттестации по дисциплине (на экзамене).

4.2 Промежуточная аттестация по дисциплине проводится в форме экзамена.

К экзамену допускаются студенты, положительно аттестованные по результатам текущего контроля.

4.2.1 Экзаменационные вопросы по дисциплине

1. Терминология программной индустрии
2. Методология и технология программирования, классификация, определения, требования к технологии,
3. Виды используемых в технологии программирования стандартов, их назначение
4. Основные проблемы программной инженерии на современном этапе
5. Основные участники программного проекта
6. Требования к профессиональному программисту
7. Понятие жизненного цикла (ЖЦ), основные стадии ЖЦ программной системы (ПС)
8. Основные модели ЖЦ, их достоинства и недостатки, распределение затрат по стадиям ЖЦ
9. Стадия ЖЦ ПС "Возникновение и исследование идеи"
10. Стадия ЖЦ ПС "Планирование проекта": определения, цели, решаемые задачи, виды ограничений
11. Виды дополнительных планов при разработке ПС
12. График разработки ПС: критический путь, резервы, виды оптимизации
13. Документация программного проекта: классификация, функции
14. Структура затрат на разработку программного проекта, оцениваемые параметры проекта
15. Виды метрик, используемые при оценивании программного проекта
16. Методика предварительного оценивания параметров программного проекта
17. Стадия ЖЦ ПС "Формирование и анализ требований": задачи системного анализа проблемы
18. Поведенческие модели, используемые при анализе проблемы, решаемой проектируемой ПС.
19. Функциональные модели, используемые при анализе проблемы, решаемой проектируемой ПС
20. Модели ERD
21. Спецификации. Языки описания требований к программному проекту. Пользователи спецификаций

22. Понятие проекта и проектирования ПС, характеристики проекта, классификация
23. Сущность и структура проектирования ПС при структурном подходе
24. Архитектурное проектирование ПС, основные этапы, понятие подсистемы и модуля
25. Модели структуры, модели управления в программной системе
26. Модульная декомпозиция, понятие модуля, формирование модульной структуры ПС
27. Структуры данных ПС: определение, классификация
28. Алгоритмы: определение, классификация, рекомендации по проектированию
29. Модели UML на этапе проектирования ПС при объектном подходе
30. Модели UML на этапе реализации ПС при объектном подходе
31. Графический интерфейс пользователя: структура, основные элементы, достоинства
32. Структура проектирования интерфейса пользователя, принципы проектирования
33. Ввод информации: основные формы ввода, рекомендации по организации компьютерного ввода/Я буду со второй пары вывода
34. Средства поддержки пользователя. Принципы, закладываемые в сообщения системы и справочную систему
35. Справочная система ПС
36. Документация пользователя, структура, пользователи
37. Стадия ЖЦ ПС "Кодирование": основные рекомендации по формированию листинга
38. Защитное и парное программирование
39. Стадия ЖЦ ПС "Тестирование": задачи, виды ошибок, понятие теста, отладки
40. Структура отладки программной системы и работы с ошибкой
41. Функциональное тестирование: задачи, достоинства, недостатки
42. Структурное тестирование: задачи, стратегии, достоинства, недостатки
43. Этапы тестирования ПС
44. Принципы тестирования программной системы
45. Ввод программной системы в действие
46. Стадия ЖЦ ПС "Эксплуатация и сопровождение": виды сопровождения, решаемые задачи
47. Технологические процессы ЖЦ ПС: определение, структура в соответствии с международными стандартами

48. Методология MSF: основные принципы
49. Методология MSF: модель команды
50. Методология MSF: модель процесса
51. Управление проектом: определение, основные задачи и проблемы
52. Руководство проектом: участники, задачи, виды менеджмента
53. Руководство проектом: подбор персонала разработки ПС, его рабочая среда
54. Оценивания уровня развития персонала разработки ПС
55. Управление готовностью: определение, решаемые задачи, основные этапы
56. Риски: определение, классификации, структура управления
57. Риски: определение рисков
58. Риски: анализ рисков
59. Риски: планирование рисков
60. Риски: мониторинг рисков
61. Качество разрабатываемого ПС: определение, влияющие факторы
62. Основные характеристики качества ПС
63. Надежность программных продуктов: определение, подходы к обеспечению, методы определения надежности
64. Формирование требований к качеству ПС
65. Основные подходы к достижению качества ПС
66. Сертификация программных продуктов
67. Рекомендации по эффективному управлению программными проектами
68. Оценка зрелости персонала
69. Стандартизация: основные понятия
70. Классификация стандартов
71. Стандартизация в области программного обеспечения

4.2.2 Оценочные средства экзамена

Экзаменационная оценка («отлично», «хорошо», «удовлетворительно» или «неудовлетворительно») является экспертной и зависит от уровня освоения студентом тем дисциплины (наличия и сущности ошибок, допущенных студентом при ответе на экзаменационные вопросы в билете, и дополнительные вопросы теоретического курса и лабораторных занятий).

Оценка **“отлично”** на экзамене выставляется студенту, который:

- дал полный ответ на два вопроса.

- при ответе на дополнительные вопросы показал знание всех разделов курса.

Оценка **“хорошо”** на экзамене выставляется студенту, который:

- дал ответ на два вопроса, за исключением наиболее трудных. Допускает незначительные неточности в доказательствах.

- при ответе на дополнительные вопросы показал знание всех разделов курса.

Оценка **“удовлетворительно”** на экзамене выставляется студенту, который:

- дал ответ на два вопроса. Допускает неточности и пробелы в формулировках, не нарушающие общей логики рассуждений.

- при ответе на дополнительные вопросы показал знание основных понятий и наиболее важных законов программы курса.

Оценка **“неудовлетворительно”** выставляется студенту, который:

- при ответе на вопросы допускает грубые ошибки.
- отвечая на дополнительные вопросы, демонстрирует существенные пробелы в знаниях.

5 СВЕДЕНИЯ О ФОНДЕ ОЦЕНОЧНЫХ СРЕДСТВ И ЕГО СОГЛАСОВАНИИ

Фонд оценочных средств для аттестации по дисциплине «Программная инженерия» представляет собой компонент основной профессиональной образовательной программы бакалавриата по направлению подготовки 09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления».

Фонд оценочных средств рассмотрен и одобрен на заседании кафедры систем управления и вычислительной техники 25.04.2022 г. (протокол № 5).

Заведующий кафедрой



В.А. Петрикин

ТЕСТОВЫЕ ЗАДАНИЯ ПО ДИСЦИПЛИНЕ

Вариант 1

Задание 1

Головной международной организацией в разработке стандартов является ...

Задание 2

В жизненном цикле программной системы третьей является фаза ...

Задание 3

Нотации являются составной частью ... создания программных систем

Задание 4

Результаты технологической операции обязаны быть в ... представлении

Задание 5

Расставьте последовательно стадии жизненного цикла программной системы:

- A. возникновение и исследование идеи;
- B. формирование и анализ требований к ПС;
- C. планирование разработки ПС;
- D. проектирование ПС;
- E. программирование ПС;
- F. тестирование и отладка ПС;
- G. ввод в действие ПС;
- H. эксплуатация и сопровождение ПС;
- I. завершение эксплуатации ПС.

Задание 6

План – это средство снятия неопределенности относительно требуемого состояния системы, поведения системы и/или ... среды

Задание 7

Минимальное время разработки программной системы определяется длиной ... в сетевом графике разработки

Задание 8

LOC –lines of ...

Задание 9

Для систем реального времени наиболее подходящей является ... модель жизненного цикла программных систем

Задание 10

КТС – комплекс ... средств

Задание 11

Главное отличие программного проекта от других видов проектов это ...

Задание 12

Модель предметной области, которая строится для описания текущего состояния, называется модель ...

Задание 13

Диаграмма переходов состояний (ДПС) является графической формой предоставления ... автомата.

Задание 14

... – реальный или абстрактный объект, имеющий определяющее значение при формировании информационной модели ПрО

Задание 15

Подход, с помощью которого ищутся пути решения определенной проблемы, обеспечивая переход от требований к их исполнению, называется ...

Задание 16

... алгоритм —алгоритм, заключающийся в принятии локально оптимальных решений на каждом этапе, допуская, что конечное решение также окажется оптимальным

Задание 17

ООП - ...- ориентированное программирование

Задание 18

... в ООП– отношение между классами, при котором объект одного класса (подтипа) обязательно является также и объектом другого класса, называемого супертипом

Задание 19

... – список операций, которые определяют услуги класса или компонента при объектном проектировании

Задание 20

Системы реального времени управляются ...

Задание 21

... - регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи в виде обмена информацией и/или координации действий

Задание 22

Не рекомендуется использовать в листинге цикл `While` в системах ...

Задание 23

Защитное программирование требует в логических условиях не вычитать близкие ... числа

Задание 24

... - способ разработки и написания программной системы, позволяющий с минимальными затратами ее адаптировать для дальнейшего использования на международных рынках (локализовать).

Задание 25

Тестирование программной системы для оценки соответствия исходным требованиям и основным критериям качества называется ...

Задание 26

Ошибка в программной системе называется ..., если ее появление не зависит от значений входных данных

Задание 27

... - мера взаимодействия модулей программной системы по данным и управлению (или наоборот, мера относительной независимости модулей).

Задание 28

Тестирование по стандарту ..., который предполагает прохождение каждой ветви управляющего графа программной системы хотя бы один раз

Задание 29

Три правила «хорошего тона» используются на стадии ... жизненного цикла программной системы

Задание 30

... - все действия по повышению надежности (качества) программного продукта после завершения отладки и разработке усовершенствованных версий (развитию функциональных возможностей)

Вариант 2

Задание 1

В жизненном цикле программной системы первой является фаза ...

Задание 2

Нотация SADT используется для построения ... моделей проектируемой программной системы

Задание 3

Язык ... используется для построения набора моделей при объект-ориентированной методологии создания программных систем

Задание 4

Структурное тестирование рассматривает программу как ... ящик

Задание 5

с ...

Задание 6

Первая задача, решаемая при планировании, это задача принятия решений по ...

Задание 7

LOC – ... of code

Задание 8

Оценка выполнимости системы оформляется в виде ...

Задание 9

Модель предметной области, которая строится для описания будущего состояния после автоматизации, называется модель ...

Задание 10

ERD - ...-Relationship Diagrams

Задание 11

Диаграмма переходов состояний является ... формой предоставления конечного автомата

Задание 12

Сущность – ... или абстрактный объект, имеющий определяющее значение при формировании информационной модели ПрО

Задание 13

Актор является элементом диаграммы вариантов ... в языке UML

Задание 14

«Плавательные дорожки» являются осями времени в диаграммах ... UML

Задание 15

Формализованные требования к программной системе, зафиксированные на некотором формальном языке и, предполагающем однозначность трактовки описания называются ...

Задание 16

Принцип "разделяй и властвуй" используется ... подходом к проектированию программных систем

Задание 17

При модульном программировании сокращение времени разработки программной системы происходит из-за возможности ... работ

Задание 18

Мера прочности соединения функциональных и информационных объектов внутри одного модуля называется ...

Задание 19

... – множество элементов данных, множество связей между ними, а также характер их организованности

Задание 20

... - это "конечный набор правил, позволяющих чисто механически решать любую конкретную задачу из некоторого класса однотипных задач".

Задание 21

... – совокупность программно-аппаратных средств, обеспечивающих взаимодействие пользователя с компьютером, т.е. это оболочка программного продукта, осуществляющая взаимосвязь между пользователем и ядром программы

Задание 22

В системах реального времени не рекомендуется использовать в листинге цикл ...

Задание 23

Документация пользователя программной системы должна включать ... разделов

Задание 24

Для комфортной работы скорость реакции программной системы на действия ...

Задание 25

Принцип KISS (Keep It Simple Stupid – будь проще, дурачок) используется на стадии ... программного проекта

Задание 26

... - процесс адаптации ПС для выбранного региона со своим языком, традициями и культурной средой

Задание 27

Тестирование отдельных компонентов программной системы называется ...

Задание 28

... - набор исходных данных и условий для запуска программы, а также набор ожидаемых результатов работы программы или известных правил поведения этих программ

Задание 29

Документ по инсталляции системы это ... раздел документации пользователя

Задание 30

Минимальная стоимость исправления ошибки в программной системе на стадии ...

Вариант 3

Задание 1

В жизненном цикле программной системы второй является фаза ...

Задание 2

Модели SADT строятся при ... методологии проектирования программных систем

Задание 3

Программа рассматривается как ... ящик при функциональном подходе к тестированию

Задание 4

Для тестировщиков программного обеспечения самая последняя выявленная ошибка является ...

Задание 5

Головной организацией в России по стандартизации является ...

Задание 6

Исторически первой была ... модель жизненного цикла программной системы

Задание 7

Вторая задача, решаемая при планировании, это задача принятия решений по ...

Задание 8

Документ, определяющий зависимость между этапами создания программной системы, оценивающий время выполнения и распределение разработчиков по этапам называется ...

Задание 9

... программного обеспечения — мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций

Задание 10

ТЭО – технико-... обоснование выполнимости программной системы

Задание 11

ERD - Entity-<...> Diagrams

Задание 12

После модели предметной области “AS-IS”, которая строится для описания текущего состояния, для описания будущего состояния строится модель ...

Задание 13

... переходов состояний является графической формой предоставления конечного автомата

Задание 14

Сущность – реальный или абстрактный объект, имеющий определяющее значение при формировании ... модели предметной области

Задание 15

Первой при объектном подходе к проектированию программных систем строится ...

Задание 16

... – это описание функциональных возможностей и ограничений, накладываемых на проектируемую программную систему

Задание 17

Использует спецификацию для лучшего понимания работы системы ... системы

Задание 18

В русскоязычной терминологии спецификация разрабатываемой программной системы называется ...

Задание 19

В процессе пошаговой детализации при структурном подходе к проектированию программных систем в первую очередь детализируются ... процессы

Задание 20

Подсистема – программное приложение, способное функционировать ...

Задание 21

Концепция DWIM (Do What I Mean – делай то, что я имею в виду) используется при проектировании ... программной системы

Задание 22

Документацию по программной системе должен готовить ...

Задание 23

Документ по установке системы представляет ... раздел документации пользователя

Задание 24

Принцип EIBTI (Explicit Is Better Than Implicit – явное лучше неявного) используется на стадии ... программного проекта ...

Задание 25

Исследование показало, что в листинге программе в диапазоне от 8 до 16% его объема должны составлять ... строки

Задание 26

... – это регулярная деятельность по переписыванию кода для улучшения его структуры

Задание 27

Ввод информации в компьютер в диалоговом режиме, при котором информация не отображается на экране, называется ...

Задание 28

За единицу принимается стоимость исправления ошибки в программной системе на стадии

...

Задание 29

Метрикой характеристики теста ... является процент выявляемых ошибок

Задание 30

... - мера прочности соединения функциональных и информационных объектов внутри одного модуля