

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
ФГБОУ ВО «Уральский государственный экономический университет»

Одобрено

на заседании педагогического совета
колледжа

23 апреля 2020 г.
протокол № 9

Директор колледжа



А.Э. Чечулин

Утверждено

советом по учебно-методическим вопросам
и качеству образования

20 мая 2020 г.
протокол № 9

Председатель



Д.А. Карх

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА МЕЖДИСЦИПЛИНАРНОГО КУРСА

Наименование междисциплинарного курса	Поддержка и тестирование программных модулей
Наименование специальности	09.02.07 Информационные системы и программирование
Форма обучения	Очно-заочная
Год набора	2020

Разработано

преподавателем

В.В. Плещевым

Екатеринбург
2020

1. ОБЩАЯ ХАРАКТЕРИСТИКА

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся.

КОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации.

Результатом освоения междисциплинарного курса является формирование у обучающихся следующих профессиональных компетенций:

ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода

В результате освоения междисциплинарного курса обучающийся должен иметь:

Практический опыт	Умения	Знания
<ul style="list-style-type: none">- в использовании инструментальных средств на этапе отладки программного продукта;- в проведении тестирования программного модуля по определенному сценарию;	<ul style="list-style-type: none">- выполнять отладку и тестирование программы на уровне модуля;- выполнять оптимизацию и рефакторинг программного кода;- оформлять документацию на программные средства	<ul style="list-style-type: none">- основные принципы отладки и тестирования программных продуктов;- способы оптимизации и приемы рефакторинга программного кода

2. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

Текущий контроль осуществляется в форме выполнения отладки предложенного программного модуля, а также в рамках тестирования.

Примерные тестовые задания

1. Delphi – это...
 - 1) Язык программирования низкого уровня
 - 2) Система объектно-ориентированного визуального программирования
 - 3) Программа обработки видео
 - 4) Программа для перевода кода программы на язык машинных кодов.
2. RAD – это...
 - 1) Программа записи видео
 - 2) Средство модульного программирования
 - 3) Среда быстрой разработки приложений
3. Язык программирования Delphi
 - 1) Object Pascal
 - 2) Java
 - 3) C#
4. IDE – это...
 - 1) История развития Delphi

- 2) Приложение, разрабатываемое программистом
- 3) **Интегрированная среда разработки**
- 5. IDE не включает в себя: (*несколько вариантов ответа*)
 - 1) Редактор кода
 - 2) Высокопроизводительный компилятор в машинный код
 - 3) Объектно-ориентированную модель компонент
 - 4) **Эффекты и переходы**
 - 5) Визуальное построение приложений
 - 6) **Сопровождение ПП**
 - 7) Средство для построения баз данных
- 6. Объектно-ориентированная программа - ...
 - 1) Программирование, основанное на объектах
 - 2) **Совокупность объектов и способов их взаимодействия**
 - 3) Структура среды программирования

7. Установить соответствие

1) Объект	a) Атрибуты (основные характеристики), которые описывают особенности объекта (цвет, ширина, положение и т.д.)
2) Событие	b) Совокупность данных (компонентов) и методов работы с ними
3) Свойство	c) Отклик на внешнее воздействие
1 – b, 2 – c, 3 – a	

- 8. Дополнить предложение. Основным окном разрабатываемого приложения является ...
 - 1) Код
 - 2) **Форма**
 - 3) Библиотека
 - 4) Объект
- 9. Дополнить предложение. Коды для стандартных окон диалога и кнопок в системе Delphi получены от ...
 - 1) Компилятора
 - 2) C++
 - 3) **Windows**
- 10. Названия процедур на определенное событие:
 - 1) Компоненты
 - 2) **Обработчики событий**
 - 3) Редактор кода
- 11. Назначение главного меню:
 - 1) **Осуществление функций управления при разработке программ**
 - 2) Сопровождение программных продуктов
 - 3) Автоматизирует процесс тестирования программ
- 12. Назначение панели инструментов:
 - 1) Обработка событий
 - 2) **Меню команд быстрого доступа к командам, содержащее набор кнопок, функции которых эквивалентны наиболее часто употребляемым командам Главного меню**
 - 3) Построение баз данных

13. Назначение палитры компонентов:
- 1) **содержит пиктограммы, которые представляют компоненты VCL , которые можно включить в приложение**
 - 2) Подделка подписей
 - 3) Управление Paint
14. Дополнить предложение. Главное окно управляет окнами ... *(несколько вариантов ответа)*
- 1) **Инспектор объектов**
 - 2) Язык ассемблера
 - 3) **Редактора кода**
 - 4) **Проектировщика форм**
 - 5) Трансляции программы
15. Заготовка разрабатываемого приложения - ...
- 1) Компилятор
 - 2) Свойство
 - 3) **Окно проектировщика форм**
 - 4) Главное окно
16. Действия которые нельзя выполнить с помощью проектировщика форм
- 1) Добавить компоненты в форму
 - 2) **Сменить язык программирования**
 - 3) Модифицировать форму и её компоненты
 - 4) Связать обработчики событий компонента с программой на Object Pascal, содержащейся в редакторе кода
 - 5) **Документирование программы**
17. Дополнить предложение. Инспектор объектов позволяет ... *(несколько вариантов ответа)*
- 1) **Устанавливать свойства объектов**
 - 2) Изменять структуру программного кода
 - 3) Изменять наследование классов объектов
 - 4) **Назначать методы обработки событий**
18. Дополнить предложение. Инспектор объектов состоит из ... *(несколько вариантов ответа)*
- 1) Unit – программного модуля
 - 2) **Properties – списка свойств,**
 - 3) **Events – списка событий.**
 - 4) Begin...end.
19. Окно содержащее текст программы на языке Object Pascal, связанный с каждой формой в приложении:
- 1) Окно компилятора
 - 2) **Окно редактора кода**
 - 3) Окно проектировщика
 - 4) Главное окно
20. Задание метки текста из редактора ввода:
- 1) Memo1.Caption := Edit1.Text;
 - 2) **Label1. Caption := Edit1.Text;**
 - 3) Form1. Caption := Edit1.Text;
21. Обнуление строки ввода:
- 1) **Edit1.Text :='';**
 - 2) Edit1.Text := TMemo;
 - 3) Edit1.Text :='Закругляемся'

22. Передача фокуса ввода на редактор ввода
- 1) Label1.SetFocus;
 - 2) Edit1.Add;
 - 3) **Edit1.SetFocus;**
23. Открыть файл проекта Project1 можно, нажав:
- 1) CTRL+F4
 - 2) CTRL+ALT+F10
 - 3) **CTRL+F12**
 - 4) ALT+F12
24. Для обозначения комментария не используются:
- 1) (*комментарий*)
 - 2) \комментарий/
 - 3) {комментарий}
 - 4) //комментарий
25. Зарезервированное слово, объявляющее блок подключаемых к проекту модулей:
- 1) Program
 - 2) Begin..end
 - 3) **Uses**
 - 4) Forms
26. Дополнить предложение. Объект Application определяет ...
- 1) **Свойства и методы программы-приложения для Windows**
 - 2) Отображение размещения формы
 - 3) Перемещение по программному коду
27. Метод Application.Run:
- 1) Создает окно формы, регистрирует входящие в него компоненты
 - 2) **Подключает автоматический цикл обработки сообщений Windows к приложению**
 - 3) Выполняет подпрограммы раздела инициализации модулей приложения в случае их наличия
28. Контейнер, содержащий все элементы программы:
- 1) Unit1
 - 2) **Form1**
 - 3) Edit1
 - 4) Memo1
29. Дополнить предложение. Свойство **Position** определяет ...
- 1) **Размещение и размер формы**
 - 2) Подключение модулей
 - 3) Цвет шрифта
30. **poDesigned:**
- 1) Форма выводится в центр экрана
 - 2) Windows автоматически определяет начальную позицию и размеры формы
 - 3) **Форма отображается в той позиции и с теми размерами, которые были установлены при её конструировании**
31. Определяет способ выравнивания компонента внутри контейнера:
- 1) Свойство Position
 - 2) **Свойство Align**
 - 3) Свойство Memo
 - 4) Свойство ScrollBars
32. **alClient:**

- 1) **Компонент занимает всю поверхность контейнера**
 - 2) Выравнивание не используется
 - 3) Компонент помещается в нижнюю часть контейнера
33. **alCustom:**
- 1) Компонент помещается в верхнюю часть контейнера
 - 2) Компонент помещается в нижнюю часть контейнера
 - 3) **Размеры и положение компонента устанавливаются разработчиком**
34. Объект **Memo1** - ...
- 1) Однострочный редактор
 - 2) **Многострочный редактор, содержащий несколько строк текста**
 - 3) Текстовая надпись
35. Свойство **ScrollBars:**
- 1) **Задаёт в поле редактирования полосы прокрутки**
 - 2) Определяет способ выравнивания компонента внутри контейнера
 - 3) Задаёт имя объекта
36. Функциональная кнопка с изображением устанавливается из:
- 1) Палитра компонентов -> Standart -> Button
 - 2) Палитра компонентов -> Standart -> Label
 - 3) **Палитра компонентов -> Additional -> BitBtn**
37. Дополнить предложение. За стиль оформления внешней и внутренней рамок отвечают свойства ... (несколько вариантов ответа)
- 1) Align
 - 2) **BevelOuter**
 - 3) **BevelInner**
 - 4) ssBoth
 - 5) Caption
38. Установить соответствие
- | | |
|--|--|
| 1) bvLowered | a) Компонент помещается в нижнюю часть контейнера |
| 2) poScreenCenter | b) Есть обе полосы прокрутки |
| 3) bvNone | c) Рамка вдавлена |
| 4) alBottom | d) Форма выводится в центре экрана, её высота и ширина не изменяются |
| 5) ssBoth | e) Рамка отсутствует |
| 1 – c, 2- d, 3 – e, 4 – a, 5 -b | |
39. VCL – это...
- 1) Приложение Delphi
 - 2) Алгоритмизация решения задачи
 - 3) **Библиотека визуальных компонентов**
 - 4) Характеристики объектов
40. Дополнить предложение. Компоненты, которые видны на форме, как во время создания приложения, так и во время работы приложения называются ...
- 1) **Визуальными**
 - 2) Реальными
 - 3) Невизуальными
 - 4) Основными
41. К невидимым компонентам относят:
- 1) Кнопки, метки, списки блоков

2) **Таймеры, компоненты для работы с базами данных, списки изображений**

3) Библиотеки, звуки, коды

42. Установить соответствие:

1) Standart	a) Обеспечивает доступ к 32-битным элементам Windows
2) Additional	b) Компоненты, реализующие интерфейс с пользователем и процесс управления данными для БД
3) Win32	c) Включает стандартные компоненты, обеспечивающие некоторые функции интерфейса пользователя
4) System	d) Специализированные компоненты, организующие доступ к БД
5) DataAccess	e) Набор компонентов для доступа к системным ресурсам (OLE, DDE)
6) DataControl	f) Дополнительные интерфейсные компоненты для красочного оформления приложения
1) – c, 2) – f, 3) – a, 4) – e, 5) – d, 6) – b;	

43. Дополнить предложение. Имя компонента состоит из... (несколько вариантов ответа)

- 1) **Названия компонента**
- 2) Заготовки компонента
- 3) **Порядкового номера компонента**
- 4) Произвольного значения ItemIndex

44. Дополнить предложение. Свойства Height и Width показывают...

- 1) Какое действие должен выполнить программист
- 2) **Размеры компонента**
- 3) Исполняемый блок программы

45. Дополнить предложение. Проект Delphi представляет собой...

- 1) **Набор программных единиц – модулей**
- 2) Основной интерфейсный элемент в Delphi
- 3) Атрибуты (основные характеристики), которые описывают особенности объекта

46. Модуль – это...

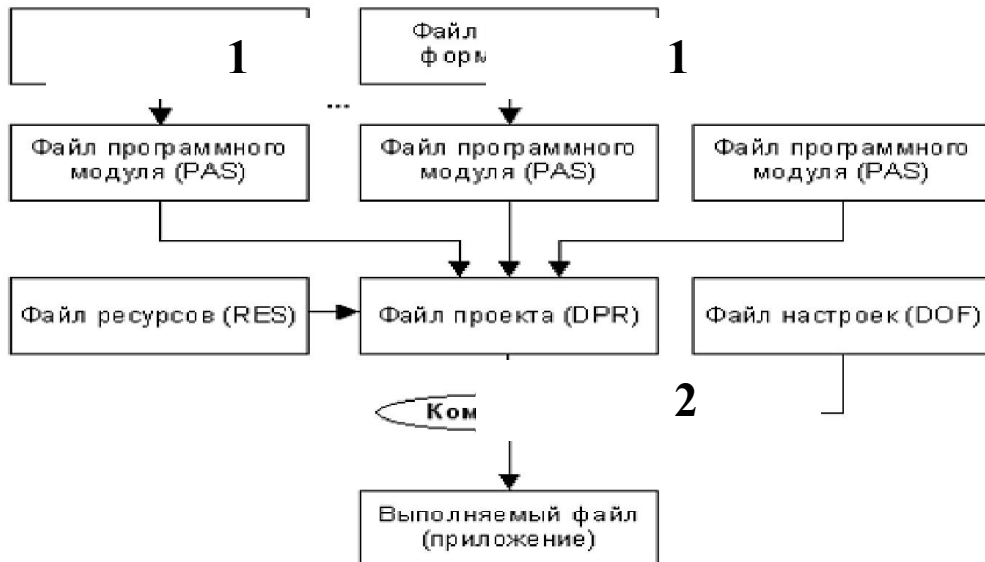
- 1) Средство для создания, изменения исходных файлов, которые содержат написанную на языке программирования программу
- 2) **Автономно компилируемая программная единица, включающая в себя различные компоненты раздела описаний (типы, константы, переменные, процедуры и функции) и, возможно некоторые исполняемые операторы иницилирующей части**
- 3) Набор инструментов, которые используются для преобразования символов

47. Установить соответствие

1) Файл проекта	a) (.pas) соответствующий файл модуль для хранения кода
2) Файл модуля	b) (.res) содержит пиктограмму и прочие ресурсы
3) Файл формы	c) (.dpr) текстовый файл используется для хранения информации о формах и модулях, содержит операторы инициализации и запуска программы на выполнение

4) Файл опций проекта	d) (.dfm) двоичный файл, который создается для хранения информации о ваших формах и фреймах
5) Файл ресурсов	e) (.cfg) хранит установки проекта
6) Файл конфигурации проекта	f) (.dof) хранит установки опций проекта
1) – с, 2) – а, 3) – d, 4) – f, 5) – b, 6) – e	

48. Заполнить схему:



- 1) **1 – файл формы (DFM); 2 – файл проекта (DPR)**
- 2) 1 – файл проекта (DPR); 2 – файл формы (DFM)
- 3) 1 – файл формы (DPR); 2 – файл проекта (DFM)
- 4) 1 – файл проекта (DFM); 2 – файл формы (DPR)

49. Дополнить предложение. Набор всех файлов, необходимых для создания приложения, называется...

- 1) Модулем
- 2) Проектом**
- 3) Задачей
- 4) Формой

50. Файлы описания форм – это...

- 1) Отклики на внешнее воздействие
- 2) Файлы подключающие все используемые программные модули
- 3) Текстовые файлы с расширением DFM, в котором сохраняются значения свойств формы и ее компонентов**

51. Содержимое dfm-файла вызывается:

- 1) Контекстное меню формы -> View as Text**
- 2) Контекстное редактора кода -> View as Form
- 3) File -> Save as...

52. Дополнить предложение. При сборке приложения описание из DFM-файла помещается в...

- 1) Файл модуля
- 2) Область ресурсов**
- 3) Область установок проекта

53. Программный модуль (Unit) содержит:

- 1) **Все относящиеся к форме объявления и методы обработки событий, написанные на языке Object Pascal**
 - 2) Заготовку, макет одного из окон разрабатываемого приложения
 - 3) Пиктограммы, которые представляют компоненты VCL , которые можно включить в приложение
54. Дополнить предложение. Имя модуля должно совпадать с...
- 1) Приложением Windows
 - 2) Спецификацией задачи
 - 3) Объектом в Delphi
 - 4) **Именем файла при сохранении File->Save as...**
55. Интерфейсная секция модуля (interface) содержит: (несколько вариантов ответа)
- 1) **Список стандартных модулей библиотеки VCL**
 - 2) **Раздел описания типов**
 - 3) Компилятор
 - 4) **Раздел описания переменных (объявление собственно объекта формы)**
 - 5) Инспектор объектов и проектировщик форм
56. Дополнить предложение. Класс формы объявляется в разделе...
- 1) **Type**
 - 2) Uses
 - 3) Project
 - 4) Var
57. Дополнить предложение. В описании класса форм помещенные на форму компоненты представлены...
- 1) Границами формы
 - 2) **Полями формы**
 - 3) Индексами формы
58. Published – это...
- 1) **Атрибут видимости, позволяющий работать с компонентами и методами обработки событий на визуальном уровне**
 - 2) Директива препроцессора
 - 3) Текст программы на языке Object Pascal
59. Пустые секции, в которых можно размещать любые вспомогательные поля, методы и свойства: (несколько вариантов ответа)
- 1) **Private**
 - 2) Uses
 - 3) Var
 - 4) **Public**
 - 5) Program
60. Объявление объекта формы:
- 1) unit Unit1;
 - 2) type TForm1 = class(TForm)
 - 3) **var Form1: TForm1;**
61. implementation – это...
- 1) Тип данных Delphi
 - 2) Племя в Южной Африке
 - 3) **Раздел реализации объявлений интерфейса**
62. Подключение файла описания формы:
- 1) **{SR *.dfm}**
 - 2) {SR *.pas}

- 3) {R *.dpr}
63. Дополнить предложение. После подключения файла описания формы следует...
- 1) Раздел описания типов
 - 2) Атрибут видимости
 - 3) Описание размеров используемых компонентов
 - 4) **Реализация методов обработки событий**
64. Установить метод обработки события Button1Click: (несколько вариантов ответа)
- 1) **Двойной щелчок по компоненту Button1 на форме**
 - 2) **Выделить Button1 ->Events -> OnClick**
 - 3) Выделить BitBtn1 ->Events -> OnClick
65. Программный файл проекта – это...
- 1) Программист
 - 2) **Главная часть приложения, с которой начинается выполнение программы и которая обеспечивает инициализацию других модулей**
 - 3) Пустая секция
66. К комментариям не относятся: (несколько вариантов ответа)
- 1) **Program**
 - 2) {Form1}
 - 3) **{SR *.res}**
 - 4) //выполнение команды
67. Дополнить предложение. Сочетание end со следующей за ней точкой называется...
- 1) Терминалом программной единицы
 - 2) **Терминатором программной единицы**
 - 3) Андройдом программной единицы
68. Содержимое DPR-файла вызывается: (несколько вариантов ответа)
- 1) Ctrl+alt
 - 2) File -> New->Application
 - 3) **Project -> View Source**
 - 4) **Ctrl+F12**
69. Дополнить предложение. Зарезервированное слово **uses** сообщает компилятору о...
- 1) **Модулях, которые должны рассматриваться как необъемлемые части программы и которые содержатся в других файлах**
 - 2) Начале работы приложения
 - 3) Ошибках, которые могут возникнуть при компилировании
70. Модуль Forms – это...
- 1) Дополнительный модуль, содержащий неиспользуемые компоненты
 - 2) Модуль для построения более красочного приложения
 - 3) **Обязательный для всех программ модуль, который содержит определение объекта Application**
71. Запись {SR *.res} означает:
- 1) Комментарий, указывающий программисту где искать ресурсы
 - 2) **Директиву, подключающую к результирующему выполняемому файлу ресурсы**
 - 3) Директиву, подключающую к результирующему выполняемому файлу форму
72. Установить соответствие:
- | | |
|---------------------------|-------------------------------------|
| 1) Application.Initialize | а) загружает и инициализирует форму |
|---------------------------|-------------------------------------|

2) Application.CreateForm	b) активизирует форму и начинает выполнение приложения
3) Application.Run	с) подготавливает приложение к работе
1 – с; 2 – а; 3 – b	

73. Включение множественного выбора элементов компонента ListBox1
- 1) ListBox2.MultiSelect:=false;
 - 2) ListBox1.MultiSelect:=true;**
 - 3) ListBox1.MultiSelect:=false;
 - 4) ListBox1.Checked:=true;
74. Запись ListBox1.Selected[3]:=true означает что:
- 1) Необходимо установить три дополнительных элемента
 - 2) Необходимо удалить три лишних элемента
 - 3) Выделен четвертый элемент в компоненте**
 - 4) Выделен третий элемент в компоненте
75. Простейшая функция вывода на экран текстовых сообщений:
- 1) ListBox1.Items.Add();
 - 2) ShowMessage();**
 - 3) Case...of;
 - 4) CheckBox1.Checked:=true;
76. Полная очистка списка:
- 1) ListBox1.Clear;**
 - 2) ListBox1.Items.Add();
 - 3) RadioGroup1.ItemIndex:=0;
77. Снятие «флажка» компонента CheckBox1
- 1) ListBox1.Checked:=true;
 - 2) CheckBox1.Checked:=false;**
 - 3) CheckBox1.Checked:=true;
 - 4) CheckBox1.Caption :='Снять флажок';
78. Свойство Align=AllBottom:
- 1) Компонент перемещается в центр формы
 - 2) Компонент перемещается в верхнюю часть контейнера
 - 3) Компонент перемещается в нижнюю часть контейнера и его ширина становится равной ширине контейнера**
79. Объект ComboBox - ...
- 1) Многострочный редактор
 - 2) Комбинированный список**
 - 3) Функциональная кнопка
80. Установка в ComboBox1 первой строки массива Item:
- 1) ItemIndex=0**
 - 2) ItemIndex=-1
 - 3) ItemIndex=1
 - 4) ItemIndex=356
81. Объект RadioGroup - ...
- 1) Команда «Авторadio»
 - 2) Панель группы радиокнопок**
 - 3) Метка
82. Дополнить предложение. Компонент-флажок, который используется для обозначения включения или отключения какой либо опции, называется...
- 1) RadioGroup

- 2) Функциональная кнопка
- 3) **CheckBox**
- 4) ListBox

3. МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Форма промежуточной аттестации – зачет.

Вопросы к зачету

1. Введение: тестирование - способ обеспечения качества программного продукта
2. Основные понятия тестирования
3. Подходы к обоснованию истинности формул и программ и их связь с тестированием. Вопросы организации тестирования. Фазы тестирования, основные проблемы тестирования и поставлена задача выбора конечного набора тестов.
4. Требования к идеальному критерию тестирования и классы частных критериев. Особенности применения структурных и функциональных критериев на базе конкретных примеров. Особенности применения методов стохастического тестирования и метод оценки скорости выявления ошибок. Мутационный критерий и на примере иллюстрируется техника работы с ним.
5. Оценка оттестированности проекта: метрики и методика интегральной оценки
6. Графовые модели проекта, метрики оценки оттестированности проекта, приводятся примеры плоской и иерархической моделей проекта.
7. Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных, динамические и статические методы при структурном подходе. Взаимосвязь сборки модулей и методов интеграционного тестирования. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования. Рассматриваются особенности интеграционного тестирования в процедурном программировании.
8. Интеграционное тестирование и его особенности для объектно-ориентированного программирования
9. Модель объектно-ориентированной программы, использующая понятие R-путей и MM-путей. Оценки сложности тестирования и методика тестирования объектно-ориентированной программы. Рассматривается пример интеграционного тестирования.
10. Разновидности тестирования: системное и регрессионное тестирование
11. Автоматизация тестирования
12. Особенности индустриального тестирования
13. Документирование и оценка индустриального тестирования
14. Описываются особенности документирования тестовых процедур для ручных и автоматизированных тестов, описаний тестовых наборов и тестовых отчетов. Рассматривается жизненный цикл дефекта. Обсуждаются метрики, используемые при тестировании.
15. Регрессионное тестирование: цели и задачи, условия применения, классификация тестов и методов отбора
16. Регрессионное тестирование: разновидности метода отбора тестов
17. Регрессионное тестирование: методики, не связанные с отбором тестов и методики порождения тестов
18. Регрессионное тестирование: алгоритм и программная система поддержки
19. Описание тестируемой системы и ее окружения. Планирование тестирования
20. Модульное тестирование на примере классов

21. Интеграционное тестирование
22. Системное тестирование
23. Ручное тестирование
24. Автоматизация тестирования с помощью скриптов
25. Автоматическая генерация тестов на основе формального описания
26. Описание ручного тестирования
27. Автоматизация тестирования с помощью скриптов
28. Описание автоматической генерации MSC тестов

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА

Критерии оценки (недифференцированной)

Оценка «зачтено» выставляется студенту, проявившему знания основного программного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, но допустившему неполные или слабо аргументированные ответы, испытывающему затруднения.

Оценка «не зачтено» выставляется студенту, обнаружившему значительные пробелы в знании программного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине.