

ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

«ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СПЕЦИАЛЬНЫХ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ»

Специальность

27.05.01 Специальные организационно-технические системы

Специализация

Информационные технологии и программное обеспечение в специальных
организационно-технических системах

Квалификация (степень) выпускника — инженер-системотехник

Форма обучения — очная, очно-заочная

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (контрольных заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Промежуточный контроль по дисциплине осуществляется путем проведения зачета. Форма проведения зачета – тестирование и выполнение практических заданий. При необходимости, проводится теоретическая беседа с обучаемым для уточнения оценки. Выполнение заданий на практических занятиях в течение семестра и заданий на самостоятельную работу является обязательным условием для допуска к зачету.

2. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного средства
Тема 1. Технология программирования. Технологичность программных продуктов	ПК-1.1, ПК-1.2, ПК-1.3	Зачет
Тема 2. Требования к программному обеспечению. Анализ требований	ПК-1.1, ПК-1.2, ПК-1.3	Зачет
Тема 3. Проектирование программного обеспечения при объектном подходе. Тестирование программных продуктов	ПК-1.1, ПК-1.2, ПК-1.3	Зачет
Тема 4. Составление программной документации	ПК-1.1, ПК-1.2, ПК-1.3	Зачет

3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Описание критериев и шкалы оценивания промежуточной аттестации

а) описание критериев и шкалы оценивания тестирования:

За каждый тестовый вопрос назначается максимально 1 балл в соответствии со следующим правилом:

- 1 балл – ответ на тестовый вопрос полностью правильный;
- 0,5 балла – отчет на тестовый вопрос частично правильный (выбраны не все правильные варианты, указаны частично верные варианты);
- 0 баллов – ответ на тестовый вопрос полностью не верный.

б) описание критериев и шкалы оценивания решения практического задания:

Шкала оценивания	Критерий
5 баллов (эталонный уровень)	Задание выполнено верно, полностью самостоятельно, без дополнительных наводящих вопросов преподавателя
3 балла (продвинутый уровень)	Задание выполнено верно, но имеются технические неточности

Шкала оценивания	Критерий
1 балл (пороговый уровень)	Задание выполнено верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задание не выполнено

На промежуточную аттестацию (зачет) выносятся 10 тестовых вопросов, два практических задания. Максимально студент может набрать 20 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено» и «не зачтено».

Оценка «зачтено» выставляется студенту, который набрал в сумме более 7 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 7 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация

Коды компетенций	Результаты освоения ОПОП Содержание компетенций
ПК-1	Способен руководить процессом разработки, проверки работоспособности и интеграцией программного обеспечения
ПК-1.1	Осуществляет руководство разработкой программного обеспечения
ПК-1.2	Осуществляет руководство проверкой работоспособности программного обеспечения
ПК-1.3	Осуществляет руководство интеграцией программного обеспечения

а) типовые тестовые вопросы:

- Для каких целей служит вариант использования на диаграмме вариантов использования:
 - представляет класс предметной области;
 - + описывает действия, совершаемые системой под воздействием актера;
 - представляет объект заданного класса;
 - описывает события в системе;
 - передает сообщение между объектами системы.
- Диаграмма классов используется для:
 - описания функций системы;
 - + представления классов системы и статических связей между ними;
 - задания взаимодействия системы с внешними объектами;
 - задания сервисов системы для актеров;
 - описания последовательности событий в системе.
- Как на диаграмме последовательности отображается время существования объекта в системе:
 - функцией отсчета времени;
 - фокусом активности;
 - + линией жизни;
 - временным интервалом между сообщениями;
 - типом объекта.
- Диаграмма кооперации показывает:
 - совокупность объектов предметной области;
 - + потоки данных между объектами;
 - операции объектов;

- атрибуты объектов;
- наследование объектов.

5. Триггерный переход между состояниями срабатывает:

- при завершении до-деятельности;
- при завершении указанного интервала времени;
- + при наступлении события, внешнего по отношению к исходному состоянию;
- при выполнении заданного условия;
- при совпадении имени внешнего события и внутреннего действия.

6. Нетриггерный переход между состояниями срабатывает:

- + при завершении до-деятельности;
- при завершении указанного интервала времени;
- при наступлении события, внешнего по отношению к исходному состоянию;
- при выполнении заданного условия;
- при совпадении имени внешнего события и внутреннего действия.

7. Какая модель жизненного цикла подразумевает выполнение проекта без возможности возврата на предыдущие этапы:

- +каскадная;
- с промежуточным контролем;
- спиральная;
- инкрементальная.

8. Какая модель жизненного цикла является итерационной разновидностью каскадной модели:

- каскадная;
- +с промежуточным контролем;
- спиральная;
- инкрементальная.

9. Какая модель жизненного цикла основана на постепенном наращивании функционала с повторными уточнениями задач:

- каскадная;
- с промежуточным контролем;
- +спиральная;
- инкрементальная.

10. Какую модель жизненного цикла предпочтительнее использовать при большом количестве итераций:

- каскадная;
- с промежуточным контролем;
- спиральная;
- +инкрементальная.

11. Диаграмма деятельности предназначена:

- для определения условий перехода между состояниями;
- для описания внутренних функций состояний;
- + для детализации выполняемых системой действий;
- для задания времени нахождения в состоянии;
- для задания дорожек ответственности классов.

12. Для описания поведения моделируемой системы используется:

- диаграмма вариантов использования;
- диаграмма компонентов;
- диаграмма кооперации;
- +диаграмма состояний;

диаграмма деятельности.

13. Диаграмма компонентов языка UML показывает:

- иерархию классов предметной области;
- результаты объектной декомпозиции;
- + физическую структуру разрабатываемого программного обеспечения;
- функции, выполняемые системой;
- потоки данных в системе.

14. Диаграмма развертывания языка UML показывает:

- иерархию классов предметной области;
- результаты объектной декомпозиции;
- аппаратную конфигурацию узлов системы;
- + аппаратную конфигурацию узлов системы с программными компонентами;
- Функции, выполняемые системой.

15. Архитектура объектно-ориентированной программы представляет собой:

- набор процедур и функций;
- иерархию классов предметной области;
- + множество объектов, обменивающихся сообщениями;
- обработчики внутренних и внешних событий;
- множество экранных форм.

16. 5. Класс - это:

- +тип данных, включающий описание свойств и функций объектов;
- множество объектов с общим состоянием и поведением;
- тип данных, включающий описание функций с возможностью их повторного использования;
- множество объектов, имеющих общие свойства и функции.

17. Объект - это:

- абстрактный тип данных, включающий состояние и поведение;
- набор данных и функций работы с ними;
- +экземпляр класса;
- тип данных, реализующий функции класса и хранящий его состояние.

18. Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью называется:

- инкапсуляцией;
- виртуализацией;
- полиморфизмом;
- +наследованием.

19. Тестирование белого ящика предназначено:

- для проверки правильности выполняемых функций программы;
- + для проверки внутренней логики программы;
- для проверки допустимых комбинаций входных данных;
- для проверки допустимых комбинаций выходных данных.

20. Как представляется программа при тестировании базового пути:

- множеством объектов;
- списком операторов;
- + потоковым графом;
- списками входных и выходных параметров;
- командами языка ассемблера.

21. Количество регионов потокового графа показывает:

- + количество тестовых вариантов в способе тестирования базового пути;

количество циклов в программе;
количество модулей программы;
количество переходов в программе;
количество входных данных программы.

22. Тестирование черного ящика предназначено:
для проверки допустимых комбинаций входных данных;
для проверки допустимых комбинаций выходных данных;
+ исследования функции программы на всей области определения;
определения экстремумов функции;
проверки точности вычисления функций программы.

23. Технология экстремального программирования предназначена для:
работы в экстремальных условиях;
+ для работы в условиях неопределенных или быстро меняющихся требований;
для сокращения объема программного кода;
для уменьшения времени на разработку технического задания;
для сокращения количества тестовых вариантов.

24. Основными целями технология экстремального программирования являются:
повышение качества интерфейса;
+ максимальное сокращение сроков выпуска работоспособной программы;
минимизация ошибок;
повышение быстродействия программы;
оптимизация программного кода.

25. Итерации экстремального программирования выполняются на основе:
вариантов использования;
+пользовательских историй;
анализа исходного текста программы;
результатов тестирования программы;
анализа быстродействия программы.

26. Технология визуального программирования это:
способ создания экранных форм;
+способ создания программы путём манипулирования графическими объектами вместо написания её текста;
построение графических образов;
технология улучшения визуального качества изображений.

б) типовые практические задания:

Задание 1

В соответствии с вариантом разработать для заданной предметной области диаграмму вариантов использования и описательную спецификацию.

Критерии выполнения задания 1

Задание считается выполненным, если обучающийся определил варианты использования создаваемой системы и отношения между ними, а также назначил актеров, определяющих внешние подсистемы по отношению к создаваемой.

Задание 2

В соответствии с вариантом разработать для заданной предметной области диаграмму классов.

Критерии выполнения задания 2

Задание считается выполненным, если обучающийся выделил части предметной области

задачи и представил каждую из них соответствующим классом с необходимыми атрибутами и операциями, а также определил отношения в иерархии классов.

Задание 3

В соответствии с вариантом разработать для заданной предметной области диаграмму классов и последовательностей.

Критерии выполнения задания 3

Задание считается выполненным, если обучающийся определил сообщения необходимых типов между экземплярами классов системы и представил результаты в графическом виде на языке UML.

Задание 4

В соответствии с вариантом разработать для заданной предметной области диаграмму классов и кооперации.

Критерии выполнения задания 4

Задание считается выполненным, если обучающийся определил потоки данных и представил результаты в графическом виде на языке UML.

Задание 5

В соответствии с вариантом разработать для заданной предметной области диаграмму состояния для всей системы.

Критерии выполнения задания 5

Задание считается выполненным, если обучающийся выделил состояния создаваемой системы, события, вызывающие переходы между ними и представил результаты в графическом виде на языке UML.

Задание 6

В соответствии с вариантом разработать для заданной предметной области диаграмму деятельности для всей системы.

Критерии выполнения задания 6

Задание считается выполненным, если обучающийся выделил виды деятельности создаваемой системы и их взаимодействия, и представил результаты в графическом виде на языке UML.