

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:
Проректор по образовательной и
международной деятельности
_____/А.А. Кирсанкин/
« ____ » _____ 2026 г.

Рабочая программа учебной дисциплины

**УПРАВЛЕНИЕ КАЧЕСТВОМ И ТЕСТИРОВАНИЕ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

образовательной программы высшего образования –
программы бакалавриата

09.03.03 Прикладная информатика

Направленность:
Интеллектуальные информационные системы и технологии

Форма обучения: очная

Курган 2026

Рабочая программа дисциплины «Управление качеством и тестирование программного обеспечения» составлена в соответствии с учебными планами по программе бакалавриата «Прикладная информатика» (Интеллектуальные информационные системы и технологии), утвержденными:
для очной формы обучения «26» июня 2026 года

Рабочая программа дисциплины одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» «30» июня 2025 года, протокол № 12.

Рабочую программу составил:

Доцент кафедры
«Программное обеспечение
автоматизированных систем»
к.т.н., доцент

_____ А.М. Семахин

Согласовано:

Заведующий кафедрой
«Программное обеспечение
автоматизированных систем»
к.ф-м.н., доцент

_____ С. В. Косовских

Специалист по учебно-методической
работе учебно-методического отдела

_____ Г.В. Казанкова

Начальник управления
образовательной деятельности

_____ И. В. Григоренко

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 3 зачетных единицы трудоемкости (108 академических часов) (очная).

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		7
Аудиторные занятия (контактная работа с преподавателем), всего часов в том числе:	48	48
Лекции	16	16
Лабораторные работы	32	32
Практические занятия	-	-
Аудиторные занятия в интерактивной форме, часов	-	-
Самостоятельная работа, всего часов в том числе:	60	60
Контрольная работа	18	18
Подготовка к экзамену	27	27
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	15	15
Вид промежуточной аттестации	экзамен	экзамен
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	108	108

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Управление качеством и тестирование программного обеспечения» относится к дисциплинам обязательной части блока 1 «Дисциплины (модули)», часть, формируемая участниками образовательных отношений, технологии разработки и сопровождения информационно-коммуникационных систем.

Изучение дисциплины базируется на знаниях, умениях и навыках, приобретенных студентами, при изучении следующих дисциплин:

- Информатика.
- Основы программирования.
- Алгоритмы и структуры данных.
- Объектно-ориентированное программирование.
- Машинно-ориентированное программирование.

Результаты обучения по дисциплине необходимы для изучения дисциплин: «Технологии проектирования информационных систем», «Методы распознавания и идентификации объектов», «Вычислительные системы, сети и телекоммуникации».

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Цел освоения дисциплины – формирование знаний и практических навыков студентов в выявлении дефектов программного обеспечения.

Задачи дисциплины: формирование понятия качества программного обеспечения, изучение методов тестирования программного обеспечения, формирование навыков тестирования программного обеспечения.

Индикаторы и дескрипторы части соответствующей компетенции, формируемой в процессе изучения дисциплины «Управление качеством и тестирование программного обеспечения», оцениваются при помощи оценочных средств.

Компетенции, формируемые в результате освоения дисциплины:

- владение концепциями, атрибутами и методами обеспечения качества программного обеспечения, способность планировать и проводить верификацию выпусков программного продукта (ПК-9);

- способность проводить тестирование компонентов программного обеспечения информационных систем, осуществлять разработку, отладку, проверку работоспособности и рефакторинг программного кода (ПК-10).

Индикаторы и дескрипторы части соответствующей компетенции, формируемой в процессе изучения дисциплины «Управление качеством и тестирование программного обеспечения», оцениваются при помощи оценочных средств.

Планируемые результаты обучения по дисциплине «Управление качеством и тестирование программного обеспечения», индикаторы достижения компетенций ПК-9, ПК-10 перечень оценочных средств

№ п/п	Код индикатора достижения компетенции	Наименование индикатора достижения компетенции	Код планируемого результата обучения	Планируемые результаты обучения	Наименование оценочных средств
1	ИД-1ПК-9	Знать: методы обеспечения качества ПО, планирование и проведение верификации выпусков программного продукта	З (ИД-1ПК-9)	Знает стандарты и модели качества ПО	Вопросы для сдачи экзамена
2	ИД-2ПК-9	Уметь: владеть концепциями, атрибутами и методами обеспечения качества ПО, планировать и проводить верификацию выпусков программного продукта	У (ИД-2ПК-9)	Умеет: владеть концепциями, атрибутами и методами обеспечения качества ПО, планировать и проводить верификацию выпусков программного продукта	Вопросы для сдачи экзамена
3	ИД-3ПК-9	Владеть: концепциями, атрибутами и методами обеспечения качества ПО, способностью планировать и проводить верификацию выпусков программного продукта	В (ИД-3ПК-9)	Владеет: методами и инструментами контроля версий программного продукта.	Вопросы для сдачи экзамена

4	ИД-1ПК-10	Знать: тестирование компонентов программного обеспечения информационных систем, разработку, отладку, проверку работоспособности и рефакторинг программного кода	З (ИД-1ПК-10)	Знает: языки и системы программирования	Вопросы для сдачи экзамена
5	ИД-2ПК-10	Уметь: проводить тестирование компонентов программного обеспечения информационных систем, осуществлять разработку, отладку, проверку работоспособности и рефакторинг программного кода	У (ИД-2ПК-10)	Умеет: проводить тестирование компонентов программного обеспечения информационных систем, осуществлять разработку, отладку, проверку работоспособности и рефакторинг программного кода	Вопросы для сдачи экзамена
6	ИД-3ПК-10	Владеть: методами тестирования компонентов программного обеспечения информационных систем, разработкой, отладкой, проверкой работоспособности и рефакторингом программного кода	В (ИД-3ПК-10)	Владеет: методами и технологиями тестирования компонентов ПО, документирования результатов тестирования, навыками системной интеграции и адаптации программного обеспечения.	Вопросы для сдачи экзамена

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план.

Рубеж	Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем	
			Лекции	Лабораторные работы
Рубеж 1	1	Основные понятия и определения тестирования. Критерии выбора тестов	2	2
	2	Структурное тестирование программ	2	4
	3	Оценка оттестированности программы. Метод тестирования базового пути	2	2
		Рубежный контроль №1	-	2
Рубеж 2	4	Функциональное тестирование программ. Эквивалентное разбиение. Анализ граничных значений	2	4
	5	Метод функциональных диаграмм. Предположение об ошибке. Системное и регрессионное тестирование	2	4
	6	Интеграционное тестирование. Метод нисходящего тестирования	2	4
	7	Метод восходящего тестирования	2	4
	8	Автоматизированное и ручное тестиро-	2	4

		вание		
		Рубежный контроль №2	-	2
		Всего:	16	32

4.2. Содержание лекционных занятий

Тема 1. Основные понятия и определения тестирования. Критерии выбора тестов

Концепция тестирования. Подходы к обоснованию истинности формул, программ и их связь с тестированием. Понятие тестирования и отладки. Организация тестирования. Методы поиска ошибок и процедура тестирования. Фазы тестирования. Проблемы тестирования и задача выбора конечного набора тестов. Классификация видов тестирования.

Требования к идеальному критерию тестирования. Классы критериев. Структурные критерии. Функциональные критерии. Стохастические критерии. Мутационный критерий.

Тема 2. Структурное тестирование программ

Особенности тестирования «белого ящика». Методы тестирования по принципу «белого ящика». Метод покрытия операторов. Метод покрытия решений (покрытия переходов). Метод покрытия условий. Метод покрытия решений/условий. Метод комбинаторного покрытия условий. Достоинства и недостатки тестирования «белого ящика».

Тема 3. Оценка оттестированности программы. Метод тестирования базового пути

Графовые модели проекта. Плоская модель. Иерархическая модель. Сложность тестирования. Степень оттестированности программы. Управляющий граф программы. Особенности управляющего графа программы. Метрики оттестированности.

Назначение метода тестирования базового пути. Цикломатическая сложность. Способы расчёта цикломатической сложности. Независимый путь. Базовое множество. Свойства базового множества. Алгоритм метода базового пути.

Тема 4. Функциональное тестирование программы. Эквивалентное разбиение. Анализ граничных значений

Функциональное назначение, цель и задачи тестирования «чёрного ящика». Эквивалентное разбиение. Этапы метода эквивалентного разбиения. Типы классов эквивалентности. Правила определения классов эквивалентности. Шаги построения тестов.

Анализ граничных значений. Правила метода анализа граничных значений. Отличия метода анализа граничных значений от метода эквивалентного разбиения.

Достоинства и недостатки методов. Пример тестирования программы с применением дерева разбиения области исходных данных.

Тема 5. Метод функциональных диаграмм. Предположение об ошибке. Системное и регрессионное тестирование

Особенности диаграммы причинно-следственных связей. Этапы построения тестов методом функциональных диаграмм. Базовые символы записи функциональных диаграмм. Функции. Логические отношения. Достоинства и недостатки. Пример.

Особенности метода тестирования предположение об ошибке. Этапы метода. Достоинства и недостатки. Пример.

Системное тестирование. Назначение, цели и задачи. Категории тестов системного тестирования. Регрессионное тестирование. Особенности. Достоинства и недостатки.

Тема 6. Интеграционное тестирование. Метод нисходящего тестирования

Функциональное назначение, цели и задачи. Методы сборки модулей: монолитный и инкрементальный. Особенности интеграционного тестирования для процедурного программирования. Особенности интеграционного тестирования для объектно-ориентированного программирования.

Особенности нисходящего тестирования. Правила нисходящего тестирования. Главный управляющий модуль. Подчинённые модули. Этапы нисходящего тестирования. Заглушки. Категории заглушек: А-отображает проходящий параметр, В-отображает трассируемое сообщение, С-возвращает величину из таблицы, D-выполняет табличный поиск по ключу (входному параметру) и возвращает связанный с ним выходной параметр. Достоинства и недостатки нисходящего тестирования. Пример.

Тема 7. Метод восходящего тестирования

Особенности восходящего тестирования. Терминальные модули. Кластер. Драйвер. Типы драйверов: А-вызывает подчинённый модуль, В-посылает элемент данных из внутренней таблицы, С-отображает параметр из подчинённого модуля, С-является комбинацией драйверов В и С. Этапы метода восходящего тестирования. Достоинства и недостатки. Сравнение с методом нисходящего тестирования. Пример.

Тема 8. Автоматизированное и ручное тестирование программ

Автоматический прогон тестов. Структура тестового прогона. Инструментальная система автоматизации тестирования. Эффективность методов тестирования.

Модульное тестирование. Основные принципы модульного тестирования. Требования к системам тестирования. Тестовые фреймворки CppUnit, Boost, t2c. Задачи и особенности. Достоинства и недостатки. Пример.

Тестирование программных приложений с использованием Visual Studio 2019 Community. Создание модульных тестов. Запуск модульных тестов. Просмотр динамических результатов модульных тестов в Visual Studio. Создание модульных тестов с помощью IntelliTest. Платформа тестирования для модульного теста Google Test, Boost Test, CTest. Модульное тестирование приложения универсальной платформы Windows.

Ручное тестирование программ. Инспекции исходного текста. Сквозные просмотры. Проверка за столом. Оценка посредством просмотра. Сравнение ручного и автоматизированного тестирования.

4.3. Лабораторные занятия

Номер раздела, темы	Наименование раздела, темы	Наименование лабораторной работы	Норматив времени, час.
			Очная форма обучения
1	Основные понятия и определе- ния тестирования. Критерии выбора тестов	Тестирование компиляторов языка программирования Си на соответствие стандарту ANSI ISO/IEC 9899:1999	2
2	Структурное тестирование про- грамм	Тестирование программ мето- дами «белого ящика»	4
3	Оценка оттестированности программы. Метод тестирова- ния базового пути	Разработка управляющего по- токового графа	2
		Рубежный контроль №1	2
4	Функциональное тестирование программ. Эквивалентное раз- биение. Анализ граничных зна- чений	Тестирование программ мето- дами «чёрного ящика». Дерево решений	4
5	Метод функциональных диа- грамм. Предположение об ошибке. Системное и регресси- онное тестирование	Тестирование программ мето- дами «чёрного ящика». Диа- грамма причинно- следственных связей	4
6	Интеграционное тестирование. Метод нисходящего тестирова- ния	Нисходящее тестирование	4
7	Метод восходящего тестирова- ния	Восходящее тестирование	4
8	Автоматизированное и ручное тестирование	Автоматизированное тестиро- вание программ	4
		Рубежный контроль №2	2
Всего:			32

4.4 Практические занятия

Практические занятия не предусмотрены учебным планом.

4.5. Контрольная работа

Контрольная работа посвящена разработке визуального приложения, формализующего алгоритм решения задачи, представленной в варианте задания согласно методическим рекомендациям, указанным в разделе 8.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Лекционный курс основывается на методе обучения, использующем технологию, при которой обучающиеся конспектируют теоретический материал, участвуют в опросах и дискуссиях. В этом случае задействованы зрительная, слуховая, моторная и ассоциативная виды памяти.

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующей лабораторной работы.

Преподавателем запланировано использование при чтении лекций технологии учебной дискуссии. Поэтому рекомендуется фиксировать для себя интересные моменты с целью их активного обсуждения на дискуссии в конце лекции.

Залогом качественного выполнения лабораторных работ является самостоятельная подготовка к ним накануне путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале занятия.

Преподавателем запланировано применение на лабораторных занятиях технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к лабораторным занятиям, рубежным контролям и экзамену, выполнение контрольной работы.

Рекомендуемая для очной формы обучения трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
	очная форма обучения
Самостоятельное изучение тем дисциплины:	7
Индустриальное тестирование	0,5
Альфа-тестирование	0,5

Бета-тестирование	1
Юзабилити-тестирование	1
Нагрузочное тестирование	1
Тестирование локализации	1
Позитивное тестирование	1
Негативное тестирование	1
Подготовка к лабораторным занятиям (по 0,5 часа на каждое занятие очной формы обучения)	7
Подготовка к рубежным контролям (по 0,5 часа на каждый рубеж)	1
Выполнение контрольной работы	18
Подготовка к экзамену	27
Всего:	60

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности обучающихся в КГУ (для очной формы обучения).
2. Отчеты обучающихся по лабораторным работам.
3. Банк заданий к рубежным контролям № 1, № 2 (для очной формы обучения).
4. Вопросы к экзамену.
5. Контрольная работа

6.2. Система балльно-рейтинговой оценки работы обучающихся по дисциплине

Очная форма обучения

№	Наименование	Содержание						
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения обучающихся на первом учебном занятии)	Распределение баллов, 7 семестр						
		Вид учебной работы:	Посещение лекций	Выполнение и защита отчетов по лабораторным работам	Контрольная работа	Рубежный контроль №1	Рубежный контроль №2	Экзамен
		Балльная оценка:	16*8=86	26*2+76*6=466	6	5	5	30

2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и экзамена	60 и менее баллов – неудовлетворительно; 61...73 – удовлетворительно; 74... 90 – хорошо; 91...100 – отлично
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов	Для допуска к промежуточной аттестации по дисциплине за семестр обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла. В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается. Для получения экзамена без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность. Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине не снижается. За академическую активность в ходе освоения дисциплины, участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность составляет 30. Основанием для получения дополнительных баллов являются: - выполнение дополнительных заданий по дисциплине; дополнительные баллы начисляются преподавателем; - участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	В случае если к промежуточной аттестации (экзамену) набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в форме письменного тестирования, экзамен в устной форме виде ответов на вопросы в билетах к экзамену.

Перед проведением рубежного контроля преподаватель прорабатывает с обучающимися основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии.

Варианты заданий для рубежных контролей № 1, № 2 состоят из 20 вопросов. Для определения баллов при проверке рубежных контролей используются интервальные оценки, представленные в таблице

Количество правильных ответов	1-5	6-8	9-11	12-14	15-17	18-20
Количество баллов	0	1	2	3	4	5

На каждую подготовку к рубежному контролю обучающемуся отводится 1 академический час.

Преподаватель оценивает в баллах результаты рубежных контролей каждого обучающегося по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

Билет к экзамену состоит из 2 вопросов. Вопросы к экзамену доводятся до обучающихся на последней лекции в семестре. На подготовку ответа по вопросам билета к экзамену обучающемуся отводится 1 астрономический час. Каждый вопрос оценивается в 15 баллов.

Результаты текущего контроля успеваемости и экзамена заносятся преподавателем в экзаменационную ведомость, которая сдается в организационный отдел института в день экзамена, а также выставляются в зачетную книжку обучающегося.

6.4 Примеры оценочных средств для рубежных контролей, экзамена

6.4.1 Примеры заданий для рубежного контроля №1

Вариант 1_1

1. Что называется тестированием ПО?

1. Процесс исследования ПО с целью получения информации о сложности.
2. Процесс исследования ПО с целью получения информации о надежности.
3. Процесс исследования ПО с целью получения информации о производительности.
4. Процесс исследования ПО с целью получения информации о качестве.

2. Что называется гейзенбагом?

1. Семантическая ошибка.
2. Плавающая ошибка.
3. Синтаксическая ошибка.
4. Логическая ошибка.

3. Какие причины возникновения гейзенбагов?

1. Низкоуровневые ошибки.

2. Ошибки проектирования и реализации программы.
3. Ошибки синтаксиса алгоритмического языка.
4. Ошибка деления на ноль.

4. В чём отличие между аттестацией и верификацией программного продукта?

1. Аттестация подтверждает, что созданный продукт соответствует предъявляемым требованиям заказчику, верификация, что создан продукт.

2. Отличий нет.

3. Аттестация подтверждает, что создан продукт, необходимый заказчику, верификация, что созданный продукт соответствует предъявляемым требованиям.

4. Отличия незначительные.

5. Тестирование – динамический метод верификации и аттестации ПО?

1. Да.

2. Нет

Вариант 1_2

1. Что называется модульным тестированием?

1. Пошаговое (по модульное) наращиванием комплекса программ с пошаговым тестированием собираемого комплекса.

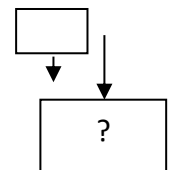
2. Тестирование, характеризующееся одновременным объединением всех модулей в тестируемый комплекс.

3. Тестирование программы на уровне отдельно взятых модулей, функций, классов.

4. Сборка и тестирование системы начинаются с модулей нижнего уровня иерархии. Модули подключаются движением снизу вверх. Отсутствие заглушек.

2. Какая категория заглушки изображена на рисунке?

1. Заглушка А – отображает проходящий параметр.
2. Заглушка В – отображает трассируемое сообщение.
3. Заглушка С – возвращает величину из таблицы.
4. Заглушка D – выполняет табличный поиск по ключу и возвращает связанный с ним выходной параметр.



3. Что называется тестовым драйвером (test driver)?

1. Драйвер, управляющий тестированием кластеров.

2. Драйвер, управляющий заглушками.

3. Драйвер, классами эквивалентности.

4. Драйвер, управляющий диаграммами причинно-следственных связей.

4. Что называется инкрементальным тестированием?

1. Объединение модулей движением сверху вниз по управляющей иерархии, начиная с главного управляющего модуля. Подчиненные модули добавляются в структуру в результате поисков глубину или в ширину.

2. Тестирование программы на уровне отдельно взятых модулей, функций, классов.

3. Пошаговое (помодульное) наращиванием комплекса программ с пошаговым тестированием собираемого комплекса.

4. Тестирование, характеризующееся одновременным объединением всех модулей в тестируемый комплекс.

5. Какой недостаток метода восходящее тестирование интеграции?

1. Количество независимых маршрутов может быть велико.

2. Нельзя обнаружить ошибки, появление которых зависит от обрабатываемых данных.

3. Необходимость заглушек и связанные с ними трудности тестирования.

4. Система не существует как объект, пока не будет добавлен последний модуль.

6.4.2 Примеры заданий для рубежного контроля №2

Вариант 2_1

1. Что называется системным тестированием?

1. Тестирование логики программы.

2. Тестирование функций программы.

3. Тестирование системы в целом, оперирующим на уровне пользовательских интерфейсов

4. Тестирование программ при сборке проекта

2. Какая основная задача системного тестирования?

1. Выявление дефектов, связанных с работой системы в целом.

2. Выявление дефектов, связанных с логикой программы.

3. Выявление дефектов, связанных с функций программы.

4. Выявление дефектов, связанных с изменениями в программе.

3. Что относится к категории тестов системного тестирования?

1. Корректность операторов.

2. Корректность переходов.

3. Корректность условий.

4. Корректность документации.

4. Что называется регрессионным тестированием?

1. Тестирование логики программы.

2. Тестирование, проводимое при внесении изменений на фазе системного тестирования или сопровождения продукта.

3. Тестирование функций программы.

4. Тестирование системы в целом, оперирующим на уровне пользовательских интерфейсов.

5. Что называется автоматизированным тестированием?

1. Тестирование, использующее программные средства для выполнения тестов и проверки результатов выполнения, приводящее к сокращению времени тестирования и упрощению процесса.

2. Тестирование системы в целом, оперирующим на уровне пользовательских интерфейсов.
3. Тестирование логики программы.
4. Тестирование, проводимое при внесении изменений на фазе системного тестирования или сопровождения продукта.

Вариант 2_2

1. Что называется регрессионным тестированием?

1. Тестирование логики программы.
2. Тестирование, проводимое при внесении изменений на фазе системного тестирования или сопровождения продукта.
3. Тестирование функций программы.
4. Тестирование системы в целом, оперирующим на уровне пользовательских интерфейсов.

2. Что называется автоматизированным тестированием?

1. Тестирование, использующее программные средства для выполнения тестов и проверки результатов выполнения, приводящее к сокращению времени тестирования и упрощению процесса.
2. Тестирование системы в целом, оперирующим на уровне пользовательских интерфейсов.
3. Тестирование логики программы.
4. Тестирование, проводимое при внесении изменений на фазе системного тестирования или сопровождения продукта.

3. Какая цель интеграционного тестирования?

1. Проверка функциональных требований к программе.
2. Выявление локализованных в модуле ошибок в реализации алгоритмов и определение степени готовности системы к переходу на следующий уровень разработки.
3. Выявление дефектов, связанных с работой системы в целом
4. Поиск дефектов, связанных с ошибками в реализации и интерпретации интерфейсного взаимодействия между модулями.

4. Какая цель модульного тестирования?

1. Выявление локализованных в модуле ошибок в реализации алгоритмов и определение степени готовности системы к переходу на следующий уровень разработки.
2. Выявление дефектов, связанных с работой системы в целом.
3. Проверка функциональных требований к программе.
4. Поиск дефектов, связанных с ошибками в реализации и интерпретации интерфейсного взаимодействия между модулями.

5. Что называется безопасным методом?

1. Метод, обеспечивающий выбор всех тестов, по определенным правилам.
2. Метод, обеспечивающий выбор всех тестов, обнаруживающих изменения.
3. Метод, обеспечивающий выбор всех тестов, случайным образом.
4. Метод, обеспечивающий выбор всех тестов, оптимальным образом.

6.4.3 Таблица ответов

№ вопроса	Правильные ответы			
	Рубежный контроль №1		Рубежный контроль №2	
	Вариант 1_1	Вариант 1_2	Вариант 2_1	Вариант 2_2
1	4	3	3	2
2	2	2	1	1
3	1	1	4	4
4	3	3	2	1
5	1	4	1	2

6.4.4 Примерный перечень вопросов для экзамена

1. Цель, задачи тестирования. Основные понятия и определения. Этапы тестирования программ.
2. Классификация методов тестирования программ.
3. Основные принципы тестирования программ.
4. Соотношение качества программы и времени разработки.
5. Процессы тестирования и их связь с процессами проектирования.
6. Концепция тестирования программ. Методы обоснования истинности формул.
7. Идеальный критерий тестирования. Требования к идеальному критерию тестирования. Классы критериев тестирования.
8. Структурные критерии тестирования.
9. Функциональные критерии тестирования.
10. Стохастические критерии тестирования.
11. Мутационный критерий тестирования.
12. Оценка тестируемости программы. Плоская модель управляющего графа программы.
13. Оценка тестируемости программы. Иерархическая модель управляющего графа программы.
14. Интегральная оценка тестируемости программ. Методика интегральной оценки тестируемости.
15. Методы ручное тестирования. Инспекции исходного текста.

16. Методы ручного тестирования. Сквозные просмотры.
17. Методы ручного тестирования. Проверка за столом.
18. Методы ручного тестирования. Оценка посредством просмотра.
19. Тестирование путем покрытия логики программы. Покрытие операторов. Пример.
20. Структурное тестирование. Покрытие решений. Пример.
21. Структурное тестирование. Покрытие решений/условий. Пример.
22. Структурное тестирование. Комбинаторное покрытие условий. Пример.
23. Функциональное тестирование. Эквивалентное разбиение. Предположение об ошибке.
24. Функциональное тестирование. Анализ граничных значений.
25. Функциональное тестирование. Метод диаграмм причинно-следственных связей.
26. Интеграционное тестирование программы. Метод нисходящего тестирования.
27. Интеграционное тестирование программы. Метод восходящего тестирования.
28. Интеграционное тестирование. Монолитная сборка модулей. Сравнение монолитного и интегрального подходов.
29. Особенности интеграционного тестирования для процедурного программирования.
30. Особенности интеграционного тестирования для объектно-ориентированного программирования.
31. Модульное тестирование программы. Принципы тестирования.
32. Системное тестирование программ. Категории тестов системного тестирования программ.
33. Автоматизация тестирования программ. Структура инструментальной системы автоматизации тестирования.
34. Тестовые фреймворки CppUnit, Boost, t2c. Задачи и особенности. Достоинства и недостатки. Пример.
35. Интегрированное тестирование программных продуктов с использованием Visual Studio 2019 Community. Платформы тестирования для модульного теста Google Test, Boost Test, CTest.
36. Тестирование готового программного продукта. Тестирование документации

6.4.5 Примерный перечень тем контрольной работы

6.4.5.1 Назначение, цели и задачи контрольной работы

Разрабатываемое приложение контрольной работы предназначено для формализации решения задачи тестирования программы методом базового пути.

Цель выполнения контрольной работы – закрепление теоретических знаний и повышение практических навыков в разработке программы тестирования методом базового пути на языке C++.

Задачи, решаемые студентом в процессе выполнения контрольной работы:

- изучение теоретического обоснования предметной области;
- разработка алгоритма решения задачи;
- разработка программного приложения, формализующего решения задачи;
- документирование контрольной работы.

6.4.5.2 Требования к контрольной работе

Контрольная работа включает визуальное приложение на языке C++ и пояснительную записку, содержащую разделы:

- введение;
- постановка задачи;
- алгоритм решения задачи;
- результаты работы приложения (скриншоты);
- заключение;
- список использованных источников;
- содержание;

К пояснительной записке прилагается комплект документации:

- спецификация (ГОСТ 19.202-78);
- описание программы (ГОСТ 19.402-78);
- руководство пользователя (ГОСТ 19.505-79);
- руководство программиста (ГОСТ 19.504-79).

6.4.5.3 Варианты контрольной работы

Разработать визуальное приложение на языке C++, формализующее алгоритм метода структурного тестирования базового пути.

Вариант 1

$$y = \begin{cases} a * (x + c)^2 - b, \text{ при } x = 0, b \neq 0 \\ \frac{x - a}{-c}, \text{ при } x = 0 \text{ и } b = 0 \\ a + \frac{x}{c}, \text{ в остальных случаях} \end{cases}$$

Вариант 2

$$y = \begin{cases} a * x^2 - cx + b, \text{при } x + 10 < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{при } x + 10 > 0 \text{ и } b = 0 \\ \frac{-x}{a - c}, \text{в остальных случаях} \end{cases}$$

Вариант 3

$$y = \begin{cases} a * x^3 + b * x^2, \text{при } x < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{при } x > 0 \text{ и } b = 0 \\ \frac{x + 5}{c(x - 10)}, \text{в остальных случаях} \end{cases}$$

Вариант 4

$$y = \begin{cases} a * (x + 7)^2 - b, \text{при } x < 5, b \neq 0 \\ \frac{x - c * d}{a * x}, \text{при } x > 5 \text{ и } b = 0 \\ \frac{x}{c}, \text{в остальных случаях} \end{cases}$$

Вариант 5

$$y = \begin{cases} -\frac{2 * x - c}{c * x - a}, \text{при } x < 0, b \neq 0 \\ \frac{x - a}{x - c}, \text{при } x > 0 \text{ и } b = 0 \\ -\frac{x}{c} + \frac{-b}{2 * c}, \text{в остальных случаях} \end{cases}$$

Вариант 6

$$y = \begin{cases} a * x^3 + b^2 + c, \text{при } x < 0, b \neq 0 \\ \frac{x - a}{c}, \text{при } x > 0 \text{ и } c \neq 0 \\ \frac{15 * x}{c}, \text{в остальных случаях} \end{cases}$$

Вариант 7

$$y = \begin{cases} a * x^2 + 2 * b * x + 5, \text{при } x < 5, b \neq 0 \\ \frac{3 * x - a}{c}, \text{при } x \geq 5 \text{ и } c \neq 0 \\ \frac{10 * x}{c + 6}, \text{в остальных случаях} \end{cases}$$

Вариант 8

$$y = \begin{cases} (a + 2) * x^2 + 5 * b, \text{при } x < 10, b \neq 0 \\ \frac{x + a}{7 * c}, \text{при } x \geq 10 \text{ и } c \neq 0 \\ \frac{x}{c + 14}, \text{в остальных случаях} \end{cases}$$

Вариант 9

$$y = \begin{cases} a * x^4 + b * x^2, \text{при } x < 0, b \neq 0 \\ \frac{x + a}{c}, \text{при } x \geq 0 \text{ и } c \neq 0 \\ \frac{15 * x}{c + 9}, \text{в остальных случаях} \end{cases}$$

Вариант 10

$$y = \begin{cases} a * x^2 + 25 * b * x, \text{при } x < 0, b \neq 0 \\ \frac{x - a}{c}, \text{при } x \geq 0 \text{ и } c \neq 0 \\ \frac{16 * x}{c + 8}, \text{в остальных случаях} \end{cases}$$

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная учебная литература

1. Джеф Решка, Джон Пол, Элфрид Дастин. Тестирование программного обеспечения. Внедрение, управление и автоматизация. – М.: Лори, 2012. – 600 с.
2. Кори Сандлер, Том Баджет, Гленфорд Майерс. Искусство тестирования программ. – М.: Вильямс, 2012. – 272 с.
3. Левинсон Дж. Тестирование ПО с помощью Visual Studio 2010. – М.: ЭКОМ Паблишерз, 2012. – 336 с.

7.2. Дополнительная учебная литература

1. Канер Сэм, Фолк Джек. Тестирование программного обеспечения. – М.: ДиаСофт, 2001. – 538 с.
2. Тампе Луиза. Введение в тестирование программного обеспечения. – М.: Вильямс, 2003. – 320 с.
3. Элфрид Дастин, Джеф Решка, Джон Пол. Автоматизированное тестирование программного обеспечения. – М.: Лори, 2003. – 592 с.
4. Степанченко И.В. Методы тестирования программного обеспечения: Учебное пособие. - Волгоград: ВолГТУ, 2006. - 74 с.
<http://window.edu.ru/resource/765/45765> (дата обращения: 06.03.2017).

8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

1. Семахин А.М. Управление качеством и тестирование программного обеспечения. Методические указания к выполнению лабораторных работ для студентов направления подготовки 09.03.04 «Программная инженерия». Курган, КГУ, 2017. – 43 с..
2. Семахин А.М. Управление качеством и тестирование ПО. Методические указания к выполнению практических и контрольных работ для студентов направления подготовки 09.03.04 «Программная инженерия». Курган, КГУ, 2017. – 36 с. (электронный).
3. Семахин А.М. Методы верификации и оценки качества программного обеспечения : учебное пособие. – Курган : Изд-во КГУ, 2018. – 150 с.

9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Федеральный портал «Российское образование» URL:
<http://www.edu.ru/>
2. Сайт дистанционного обучения в НОУ «ИНТУИТ». URL:
<http://www.intuit.ru/>

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

- 1 ЭБС «Лань»
- 2 ЭБС «Консультант студента»
- 3 ЭБС «Znaniium.com»
- 4 «Гарант» – справочно-правовая система

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение по реализации дисциплины осуществляется в соответствии с требованиями ФГОС ВО по данной образовательной программе.

12. ДЛЯ ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме онлайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2 либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнения ведущего преподавателя и доводится до обучающихся.

Аннотация к рабочей программе дисциплины
**«Управление качеством и тестирование
программного обеспечения»**

образовательной программы высшего образования –
программы бакалавриата

09.03.03 – Прикладная информатика

Направленность:

Интеллектуальные информационные системы и технологии

Трудоемкость дисциплины: 3 ЗЕ (108 академических часа) (очная)

Семестр: 7 (очная)

Форма промежуточной аттестации: экзамен

Содержание дисциплины

Концепция тестирования. Подходы к обоснованию истинности формул, программ и их связь с тестированием. Понятие тестирования и отладки. Классификация видов тестирования. Требования к идеальному критерию тестирования. Классы критериев.

Особенности тестирования «белого ящика». Методы тестирования по принципу «белого ящика». Достоинства и недостатки тестирования «белого ящика». Графовые модели проекта. Плоская модель. Иерархическая модель. Назначение метода тестирования базового пути. Цикломатическая сложность.

Функциональное назначение, цель и задачи тестирования «чёрного ящика». Эквивалентное разбиение. Анализ граничных значений. Особенности диаграммы причинно-следственных связей и предположения об ошибке.

Системное тестирование. Назначение, цели и задачи. Категории тестов системного тестирования. Регрессионное тестирование. Особенности. Достоинства и недостатки.

Функциональное назначение, цели и задачи. Методы сборки модулей: монолитный и инкрементальный. Особенности нисходящего тестирования и восходящего тестирования.

Автоматический прогон тестов. Структура тестового прогона. Инструментальная система автоматизации тестирования. Модульное тестирование. Основные принципы модульного тестирования. Требования к системам тестирования. Тестовые фреймворки CppUnit, Boost, t2c. Задачи и особенности. Достоинства и недостатки. Пример.

Интегрированное тестирование программных продуктов с использованием Visual Studio 2019 Community. Платформа тестирования для модульного теста Google Test, Boost Test, CTest. Ручное тестирование программ.

ЛИСТ
регистрации изменений (дополнений) в рабочую программу
учебной дисциплины
«Управление качеством и тестирование программного обеспечения»

Изменения / дополнения в рабочую программу
на 20__ / 20__ учебный год:

Ответственный преподаватель _____ / Семяхин А.М. /

Изменения утверждены на заседании кафедры «__» _____ 20__ г.,
Протокол № ____

Заведующий кафедрой _____ «__» _____ 20__ г.

Изменения / дополнения в рабочую программу
на 20__ / 20__ учебный год:

Ответственный преподаватель _____ / Семяхин А.М. /

Изменения утверждены на заседании кафедры «__» _____ 20__ г.,
Протокол № ____

Заведующий кафедрой _____ «__» _____ 20__ г.