



Частное профессиональное образовательное учреждение
«Северо-Кавказский межотраслевой колледж»

ИНН/КПП 0600001944/060001001 ОГРН 1220600000455
386101, Республика Ингушетия, г. Назрань, пр-т. Базоркина, д. 3

ПРИНЯТО
на заседании учебно-методического
совета Протокол
от «13» апреля 2026 г. № 3

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ
СРЕДСТВ
МДК 01.02 Поддержка и тестирование
программных модулей**

Специальность: 09.02.07 Информационные системы и программирование

*Тип образовательной программы: Программа подготовки специалистов
среднего звена*

Квалификация: Программист

Форма обучения: очная

Назрань, 2026

Содержание

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	3
2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК	4
3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ	5
4. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО МДК.01.02....	6
5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ ПО МДК.....	7
6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК	12
7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	14

1. ОБЩИЕ ПОЛОЖЕНИЯ

Комплект контрольно-оценочных средств (КОС) междисциплинарного курса МДК 01.02 Поддержка и тестирование программных модулей является частью ПМ.01 программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование (квалификация «Программист»).

КОС предназначен для оценки достижения запланированных результатов обучения по междисциплинарному курсу и включает оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации.

Текущий контроль успеваемости осуществляется в пределах учебного времени, отведённого на изучение МДК. Результаты текущего контроля фиксируются в журнале учебных занятий по пятибалльной системе («5», «4», «3», «2»).

Промежуточная аттестация по МДК проводится в форме дифференцированного зачета. Дифференцированный зачет проводится после завершения освоения МДК в сроки, установленные календарным учебным графиком. Вопросы и задания доводятся до сведения обучающихся в течение первых двух месяцев от начала обучения.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК

Общие компетенции (ОК)

Код	Формулировка компетенции
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 04	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке
ОК 09	Использовать информационные технологии в профессиональной деятельности
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках

Профессиональные компетенции

Код	Формулировка компетенции
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей

В результате освоения МДК обучающийся должен:

Иметь практический опыт:

- использования инструментальных средств на этапе отладки программного продукта;
- проведения тестирования программного модуля по определённом сценарию.

Уметь:

- выполнять отладку и тестирование программы на уровне модуля;
- использовать инструментальные средства отладки (точки останова, окна Watch, Immediate);
- применять методы тестирования «белого ящика» и «чёрного ящика»;
- оформлять документацию на программные средства.

Знать:

- основные принципы отладки и тестирования программных продуктов;
- виды ошибок, методы отладки;
- методы тестирования (модульное, интеграционное, системное);
- классификацию тестирования по уровням;
- средства разработки технической документации;
- стандарты Единой системы программной документации (ЕСПД).

3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ

Формы текущего контроля по МДК:

- устный опрос (фронтальный, индивидуальный, комбинированный);
- тестирование (письменное или компьютерное);
- письменная проверка (ответы на вопросы, решение задач, составление схем, выполнение заданий для самостоятельной работы);
- практическая проверка (при проведении практических и лабораторных занятий);
- самоконтроль и взаимопроверка.

Критерии оценки профессиональных компетенций (фрагмент)

Компетенция	Критерии оценки	Методы оценки
ПК 1.3 Выполнять отладку	«Отлично» – выполнена отладка модуля с использованием инструментария среды, пояснены особенности отладочных классов, сохранены и представлены результаты отладки.	Дифференцированный зачёт (практическое задание), защита отчётов по лабораторным работам, наблюдение.
ПК 1.4 Выполнять тестирование	«Отлично» – выполнено тестирование модуля (в т.ч. с помощью инструментальных средств), оформлены результаты тестирования в соответствии со стандартами.	Дифференцированный зачёт, защита отчётов, наблюдение.

Оценки по ОК выставляются на основе экспертного наблюдения за выполнением работ.

4. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО МДК.01.02

№	Наименование темы	Результаты обучения (умения, знания)	ПК, ОК	Текущий контроль успеваемости
Тема 1.2.1. Отладка и тестирование программного обеспечения				
1	Тестирование как часть процесса верификации ПО	31, 32	ПК 1.4, ОК 01,02	Устный опрос, тестирование
2	Виды ошибок (синтаксические, семантические, логические)	33	ПК 1.3	Устный опрос, тестирование
3	Методы отладки (ручная трассировка, точки останова, отладочный вывод)	У1, 34	ПК 1.3	Лабораторная работа
4	Методы тестирования «белого ящика» (покрытие операторов, решений, условий)	У2, 35	ПК 1.4	Лабораторная работа
5	Методы тестирования «чёрного ящика» (эквивалентное разбиение, анализ граничных значений)	У3, 35	ПК 1.4	Лабораторная работа
6	Классификация тестирования по уровням (модульное, интеграционное, системное, приёмочное)	У4, 36	ПК 1.4	Устный опрос, тестирование
7	Тестирование производительности (нагрузочное, стресс-тестирование)	У5, 37	ПК 1.4	Лабораторная работа
8	Регрессионное тестирование	У6, 38	ПК 1.4	Устный опрос
Тема 1.2.2. Документирование программного обеспечения				
9	Средства разработки технической документации. Технологии разработки документов	У7, 39	ПК 1.4, ОК 10	Устный опрос, тестирование
10	Документирование ПО в соответствии с Единой системой программной документации (ЕСПД)	У8, 310	ПК 1.4	Практическая работа
11	Автоматизация разработки технической документации (Sandcastle, Doxygen, GhostDoc)	У9, 311	ПК 1.4	Лабораторная работа

5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ ПО МДК

5.1. Типовые задания для оценки знаний и умений

Тема 1.2.1. Отладка и тестирование программного обеспечения

Вопросы для устного опроса:

1. Что такое верификация программного обеспечения? Чем отличается от валидации?
2. Какие виды ошибок в программах вы знаете? Приведите примеры.
3. Что такое отладка? Какие методы отладки существуют?
4. Как установить точку останова (breakpoint) в среде Visual Studio?
5. Для чего используются окна Watch, Locals, Immediate в отладчике?
6. Что такое метод тестирования «белого ящика»? Назовите основные техники.
7. Что такое метод тестирования «чёрного ящика»? Приведите примеры.
8. Что такое модульное тестирование? Какие фреймворки используются?
9. Что такое интеграционное тестирование? В чём его отличие от модульного?
10. Что такое регрессионное тестирование? Когда оно применяется?

Тестовые задания:

1. **Какой вид ошибки связан с нарушением правил языка программирования?**
 - a) Синтаксическая ошибка
 - b) Логическая ошибка
 - c) Семантическая ошибка
 - d) Ошибка времени выполнения

Ответ: a
2. **Какой метод отладки предполагает вывод промежуточных значений в консоль?**
 - a) Использование точек останова
 - b) Отладочный вывод (Debug.WriteLine)
 - c) Профилирование
 - d) Трассировка стека

Ответ: b
3. **Какая техника тестирования «чёрного ящика» делит входные данные на классы, для которых поведение программы одинаково?**
 - a) Анализ граничных значений
 - b) Эквивалентное разбиение
 - c) Попарное тестирование
 - d) Таблицы решений

Ответ: b
4. **Какое тестирование проверяет взаимодействие между отдельными модулями?**
 - a) Модульное тестирование
 - b) Интеграционное тестирование
 - c) Системное тестирование
 - d) Приёмочное тестирование

Ответ: b
5. **Что такое «регрессионное тестирование»?**
 - a) Тестирование нового функционала
 - b) Повторное тестирование ранее работавших функций после изменений
 - c) Тестирование производительности
 - d) Тестирование удобства интерфейса

Ответ: b
6. **Какой инструмент в Visual Studio позволяет пошагово выполнять код?**
 - a) IntelliSense
 - b) Debugger (F10, F11)

c) NuGet

d) Git

Ответ: b

7. **Какая техника тестирования «белого ящика» требует проверки всех возможных путей выполнения программы?**

a) Покрытие операторов

b) Покрытие решений

c) Покрытие условий

d) Покрытие путей

Ответ: d

8. **Какое тестирование проводится конечными пользователями перед выпуском продукта?**

a) Модульное

b) Интеграционное

c) Приёмочное (UAT)

d) Нагрузочное

Ответ: c

9. **Какой метод тестирования проверяет работу системы при экстремальных нагрузках?**

a) Нагрузочное тестирование

b) Стресс-тестирование

c) Конфигурационное тестирование

d) Регрессионное тестирование

Ответ: b

10. **Что такое «точка останова» (breakpoint)?**

a) Место в коде, где выполнение программы приостанавливается

b) Ошибка компиляции

c) Комментарий

d) Переменная

Ответ: a

Лабораторные работы:

1. Тестирование «белым ящиком» (покрытие операторов, решений) на примере функции расчёта скидки.
2. Тестирование «чёрным ящиком» (эквивалентное разбиение, граничные значения) для функции ввода возраста.
3. Модульное тестирование с использованием фреймворка NUnit/xUnit.
4. Интеграционное тестирование на примере взаимодействия двух классов.
5. Отладка программы с ошибками: использование точек останова, окон Watch и Immediate.

Тема 1.2.2. Документирование программного обеспечения

Вопросы для устного опроса:

1. Какие виды документации на программное обеспечение существуют (пользовательская, техническая)?
2. Что такое Единая система программной документации (ЕСПД)?
3. Какие стандарты ЕСПД вы знаете (ГОСТ [19.XXX](#))?
4. Какие разделы включает техническое задание (ТЗ) на разработку ПО?
5. Что такое «руководство программиста»? Какие сведения оно содержит?
6. Что такое «руководство оператора»?
7. Какие инструменты автоматической генерации документации по коду вы знаете (Doxxygen, Sandcastle)?
8. Как оформлять комментарии в коде для последующей генерации документации

- (XML-комментарии в C#)?
9. Что такое «спецификация требований» (SRS)?
 10. Какие требования предъявляются к оформлению текстовых документов (ГОСТ 2.105-95)?

Тестовые задания:

1. **Какой стандарт относится к Единой системе программной документации (ЕСПД)?**
 - a) ГОСТ [19.XXX](#)
 - b) ГОСТ [2.XXX](#)
 - c) ISO 9001
 - d) IEEE 829

Ответ: a
2. **Какой документ содержит описание порядка установки и запуска программы?**
 - a) Техническое задание
 - b) Руководство пользователя
 - c) Руководство программиста
 - d) Спецификация

Ответ: b
3. **Какой инструмент используется для генерации документации из XML-комментариев в C#?**
 - a) Sandcastle
 - b) MSBuild
 - c) NuGet
 - d) Git

Ответ: a
4. **Какой раздел ТЗ описывает требования к функциональности системы?**
 - a) Общие положения
 - b) Требования к программному обеспечению
 - c) Состав и содержание работ
 - d) Порядок контроля и приёмки

Ответ: b
5. **Какой формат комментариев в C# поддерживает генерацию документации?**
 - a) //
 - b) /* */
 - c) /// (XML-комментарии)
 - d) #

Ответ: c
6. **Что такое «руководство программиста»?**
 - a) Документ для конечного пользователя
 - b) Документ, описывающий архитектуру и интерфейсы для разработчиков
 - c) Тест-план
 - d) Инструкция по установке

Ответ: b
7. **Какой документ определяет цели, задачи и требования к разрабатываемому ПО?**
 - a) Техническое задание (ТЗ)
 - b) Руководство оператора
 - c) Программа и методика испытаний
 - d) Спецификация

Ответ: a
8. **Какой стандарт определяет требования к текстовым документам (общие требования)?**

- a) ГОСТ 19.101-77
- b) ГОСТ 2.105-95
- c) ГОСТ 34.602-89
- d) ISO 9001

Ответ: b

9. **Что такое «аннотация» в программной документации?**

- a) Краткое описание назначения и области применения программы
- b) Полное описание алгоритмов
- c) Список ошибок
- d) Инструкция по установке

Ответ: a

10. **Какой инструмент позволяет создавать документацию в формате HTML/PDF из XML-комментариев?**

- a) Doxygen
- b) Visual Studio Code
- c) Notepad++
- d) Git Bash

Ответ: a

Практические и лабораторные работы:

1. Оформление документации на программный модуль с использованием инструментальных средств (например, создание технического задания по ГОСТ 19.201-78).
2. Генерация документации из XML-комментариев с помощью Sandcastle или Doxygen.
3. Создание руководства пользователя для разработанного приложения.

5.2. Критерии оценивания

5.2.1. Критерии оценивания устного ответа

Оценка	Характеристика ответа
«5»	Ответ правильный, полный, логически выстроен, литературным языком.
«4»	Ответ правильный, полный, но есть отдельные затруднения в формулировке выводов.
«3»	Ответ в основном правильный, но схематичный или с нарушениями последовательности, неполный.
«2»	Непонимание основного содержания, грубые ошибки, отсутствие логики и обобщений.

5.2.2. Критерии оценивания тестовых заданий

Оценка	Процент выполнения
«5»	90–100 %
«4»	70–89,9 %
«3»	50–69,9 %
«2»	0–49,9 %

5.2.3. Критерии оценивания выполнения заданий на лабораторных и практических занятиях

Оценка	Характеристика выполнения
«5»	Работа выполнена полностью и правильно, сделаны верные выводы.
«4»	Работа выполнена правильно с 1–2 несущественными ошибками, исправленными по требованию преподавателя.

«3»	Работа выполнена не менее чем наполовину или допущены 3–4 существенные ошибки.
«2»	Допущено 5 и более существенных ошибок, которые обучающийся не может исправить.

5.2.4. Общая классификация ошибок

Грубые ошибки:

- незнание основных понятий, законов;
- неумение выделить главное, обобщить;
- неумение применить знания для решения задач;
- неумение пользоваться справочной литературой;
- нарушение техники безопасности.

Негрубые ошибки:

- неточность формулировок, определений;
- недостаточно продуманный план ответа;
- нерациональные методы работы с литературой.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК

6.1. Вопросы к дифференцированному зачёту (теоретическая часть)

1. Роль тестирования в процессе верификации программного обеспечения.
2. Виды ошибок в программах (синтаксические, семантические, логические, ошибки времени выполнения).
3. Методы отладки программного кода (ручная трассировка, точки останова, отладочный вывод).
4. Инструментарий отладчика Visual Studio: точки останова, окна Watch, Locals, Immediate, Call Stack.
5. Методы тестирования «белого ящика»: покрытие операторов, решений, условий, путей.
6. Методы тестирования «чёрного ящика»: эквивалентное разбиение, анализ граничных значений, таблицы решений, попарное тестирование.
7. Классификация тестирования по уровням: модульное, интеграционное, системное, приёмочное.
8. Модульное тестирование: цели, фреймворки (JUnit, xUnit), написание тестов.
9. Интеграционное тестирование: подходы «снизу вверх», «сверху вниз», «большой взрыв».
10. Системное тестирование: функциональное и нефункциональное (производительность, безопасность, удобство использования).
11. Тестирование производительности: нагрузочное тестирование, стресс-тестирование, тестирование стабильности.
12. Регрессионное тестирование: цели, автоматизация, выбор тестов.
13. Понятие верификации и валидации программного обеспечения.
14. Автоматизация тестирования: инструменты, преимущества и недостатки.
15. Единая система программной документации (ЕСПД): стандарты, состав документов.
16. Структура и содержание технического задания (ТЗ) на разработку ПО (по ГОСТ 19.201-78 или 34.602-89).
17. Руководство пользователя: назначение, типовое содержание.
18. Руководство программиста: назначение, типовое содержание.
19. Автоматическая генерация документации из кода: XML-комментарии, инструменты (Sandcastle, Doxygen).
20. Требования к оформлению текстовых документов (ГОСТ 2.105-95).

6.2. Практические задания к дифференцированному зачёту

1. Отладка: Предоставлен код с логической ошибкой (неправильный расчёт среднего арифметического). Используя отладчик Visual Studio (точки останова, окна Watch), найти ошибку и исправить её.
2. Модульное тестирование: Написать модульные тесты (JUnit/xUnit) для функции вычисления факториала (проверить граничные значения: 0, 1, 5, отрицательное число – ожидаемое исключение).
3. Тестирование «чёрного ящика»: Разработать тестовые сценарии (набор входных данных и ожидаемых результатов) для функции проверки возраста (допустимый возраст от 18 до 60 лет). Использовать эквивалентное разбиение и анализ граничных значений.
4. Тестирование «белого ящика»: Для фрагмента кода с условными операторами (if-else) построить граф потока управления и определить минимальное количество тестов для покрытия всех решений.
5. Интеграционное тестирование: Предоставлены два класса (Калькулятор и Логгер). Написать интеграционный тест, проверяющий, что при вызове

- метода Divide (деление на ноль) в логгер записывается сообщение об ошибке.
6. Документирование: Оформить фрагмент кода (класс «Сотрудник») XML-комментариями для последующей генерации документации. Сгенерировать HTML-документацию с помощью Sandcastle или Doxygen.
 7. Регрессионное тестирование: После внесения изменений в функцию сортировки (ускорение) запустить существующий набор тестов и проанализировать результаты. Составить отчёт о регрессионном тестировании.
 8. Нагрузочное тестирование: Используя простой инструмент (например, Apache JMeter или ручную имитацию), провести нагрузочное тестирование метода, который делает задержку (Thread.Sleep). Измерить время отклика при 10, 50, 100 одновременных вызовах.
 9. Отладка с использованием отладочного вывода: В код добавить вывод отладочной информации через Debug.WriteLine (только в Debug-конфигурации). Показать, как это помогает найти ошибку в алгоритме.
 10. Оформление документации: Составить фрагмент технического задания (раздел «Требования к программному обеспечению») для простого приложения «Калькулятор» (функции: сложение, вычитание, умножение, деление, память).

Критерии:

Оценка	Характеристика
«5» (отлично)	Теоретический ответ полный, без ошибок. Практическое задание выполнено полностью и верно, студент уверенно использует инструменты отладки/тестирования/документирования, правильно интерпретирует результаты.
«4» (хорошо)	Теоретический ответ с незначительными неточностями. Практическое задание выполнено, но есть небольшие недочёты (неполный набор тестов, не все граничные значения учтены), которые студент может исправить с помощью наводящих вопросов.
«3» (удовлетворительно)	Теоретический ответ неполный, есть ошибки в основных понятиях. Практическое задание выполнено частично или содержит существенные ошибки (например, тесты не проходят, отладка не привела к исправлению).
«2» (неудовлетворительно)	Теоретический ответ отсутствует или полностью неверен. Практическое задание не выполнено или выполнено с грубыми нарушениями (студент не владеет инструментарием).

7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Основные печатные издания

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник. – М.: Академия, 2020. – 384 с.

Основные электронные издания

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: электронный учебно-методический комплекс. – М.: Академия, 2021.

Дополнительные источники

1. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для СПО. – М.: Юрайт, 2021. – 235 с.
2. Белугина С.В. Разработка программных модулей программного обеспечения для компьютерных систем. – СПб: Лань, 2021. – 312 с.
3. Куликов С.С. Тестирование программного обеспечения. – М.: Интернет-Университет Информационных Технологий, 2019.

Интернет-ресурсы

- <https://metanit.com> – учебник по C# и .NET
- <https://learn.microsoft.com/ru-ru/visualstudio/debugger/> – документация по отладчику Visual Studio
- <https://nunit.org/> – официальный сайт NUnit
- <https://www.doxygen.nl/> – инструмент генерации документации