



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 02.02 Инструментальные средства разработки программного обеспечения

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.02.02.....	6
5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ ПО МДК.....	7
6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК	12
7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	13

1. ОБЩИЕ ПОЛОЖЕНИЯ

Комплект контрольно-оценочных средств (КОС) междисциплинарного курса МДК 02.02 Инструментальные средства разработки программного обеспечения является частью ПМ.02 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 зачёта. Дифференцированный зачёт 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	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках

Профессиональные компетенции

Код	Формулировка компетенции
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

В результате освоения МДК обучающийся должен:

Уметь:

- использовать выбранную систему контроля версий (У10, У13, У16);
- выполнять отладку с использованием методов условной компиляции (классы Debug и Trace) (У6);
- организовывать интеграцию модулей в программное обеспечение (У3);
- выполнять тестирование интеграции (У11);
- организовывать постобработку данных (У12);
- выявлять ошибки в системных компонентах (У9);
- применять методы обработки исключительных ситуаций (У6, У9).

Знать:

- современные технологии и инструменты интеграции (35);
- основные протоколы доступа к данным (36);
- методы и способы идентификации сбоев и ошибок при интеграции приложений (37);
- методы отладочных классов (38);
- стандарты качества программной документации (39);
- основы организации инспектирования и верификации (310);
- методы организации работы в команде разработчиков (313);
- методы и схемы обработки исключительных ситуаций (315).

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 2.2 Выполнять интеграцию модулей	«Отлично» – выбрана верная версия проекта в СКВ, архитектура доработана, выбраны способы форматирования данных, интеграция протестирована, результат сохранён.	Дифференцированный зачёт (практическое задание), защита отчётов по лабораторным работам, наблюдение.
ПК 2.3 Выполнять отладку	«Отлично» – выполнена отладка с применением инструментальных средств, использованы точки останова, окна Watch, условная компиляция, определены качественные показатели.	Дифференцированный зачёт, защита отчётов, наблюдение.
ПК 2.5 Инспектирование кода	«Отлично» – знание стандартов кодирования более чем одного языка, выявлены все несоответствия в предложенном коде.	Дифференцированный зачёт (инспекция кода), защита отчётов, наблюдение.

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

4. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО МДК.02.02

№	Наименование темы	Результаты обучения (умения, знания)	ПК, ОК	Текущий контроль успеваемости
Тема 2.2.1 Современные технологии и инструменты интеграции				
1	Понятие репозитория проекта, структура проекта	35, 36, 313	ПК 2.2, ОК 01-05,09	Устный опрос, тестирование
2	Виды, цели и уровни интеграции программных модулей	33, 34, 35	ПК 2.2	Устный опрос, тестирование
3	Выбор источников и приемников данных, сопоставление объектов данных	У4, У12, 36	ПК 2.2	Устный опрос, тестирование
4	Транспортные протоколы. Стандарты форматирования сообщений	У3, 35, 36	ПК 2.2	Устный опрос, тестирование
5	Организация работы команды в системе контроля версий	У10, У13, У16, 313	ПК 2.2, ПК 2.5	Лабораторная работа
Тема 2.2.2 Инструментарий тестирования и анализа качества				
6	Отладка программных продуктов. Инструменты отладки. Отладочные классы	У6, 38	ПК 2.3	Устный опрос, тестирование
7	Ручное и автоматизированное тестирование. Методы и средства организации тестирования	У11, У14, 316	ПК 2.3	Лабораторная работа
8	Инструментарии анализа качества программных продуктов в среде разработки	У15, 311	ПК 2.5	Лабораторная работа
9	Обработка исключительных ситуаций. Методы и способы идентификации сбоев и ошибок	У9, 37, 315	ПК 2.3, ПК 2.5	Устный опрос, тестирование
10	Выявление ошибок системных компонентов	У9, 37	ПК 2.5	Лабораторная работа

Самостоятельная работа обучающихся (темы для конспектирования):

- Анализ требований и их формализация (32, 39)
- Примеры систем контроля версий (39, 313)
- Стандарт кодирования C# (39)
- Основные этапы развития технологии разработки ПО (31, 32)
- Эволюция моделей жизненного цикла ПО (31, 32)
- Стандарты, регламентирующие процесс разработки ПО (31, 32, 39)
- Анализ проблемы и постановка задачи (312)
- Техническое задание – ГОСТ 34.602-89 (32, 39)

5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ ПО МДК

5.1. Типовые задания для оценки знаний и умений

Тема 2.2.1. Современные технологии и инструменты интеграции

Вопросы для устного опроса:

1. Что такое репозиторий проекта? Какие элементы в него входят?
2. Назовите виды интеграции программных модулей (горизонтальная, вертикальная, «звёздочка», шина данных).
3. Какие протоколы используются для доступа к данным (HTTP, FTP, JDBC, ODBC)?
4. Что такое REST API? Какие форматы данных чаще всего используются (JSON, XML)?
5. Как выбрать источник и приёмник данных при интеграции?
6. Что такое система контроля версий? Приведите примеры.
7. Какие команды Git вы знаете? Опишите add, commit, push, pull.
8. Как разрешить конфликт при слиянии веток в Git?
9. Что такое CI/CD? Как это связано с интеграцией модулей?
10. Какие инструменты автоматизации сборки вы знаете (MSBuild, Maven, Gradle)?

Тестовые задания:

1. **Какая команда Git используется для отправки изменений в удалённый репозиторий?**
a) git commit
b) git push
c) git pull
d) git fetch
Ответ: b
2. **Что такое «репозиторий» в системе контроля версий?**
a) Место хранения файлов проекта с историей изменений
b) Инструмент для компиляции
c) Отчёт о дефектах
d) План тестирования
Ответ: a
3. **Какой протокол чаще всего используется для REST API?**
a) FTP
b) HTTP/HTTPS
c) SMTP
d) POP3
Ответ: b
4. **Какая команда Git создаёт новую ветку?**
a) git branch new_branch
b) git checkout new_branch
c) git merge new_branch
d) git clone
Ответ: a
5. **Что такое «заглушка» (stub) при интеграционном тестировании?**
a) Полноценный модуль
b) Временная имитация модуля
c) Отчёт об ошибках
d) Система контроля версий
Ответ: b
6. **Какой формат данных чаще всего используется для обмена между микросервисами?**

- a) JSON
- b) DOCX
- c) EXE
- d) DLL

Ответ: a

7. **Что такое CI/CD?**

- a) Continuous Integration / Continuous Delivery
- b) Code Inspection / Code Debugging
- c) Compiler Interface / Compiler Directive
- d) Component Integration / Component Deployment

Ответ: a

8. **Какая система контроля версий использует централизованную модель?**

- a) Git
- b) Mercurial
- c) Subversion (SVN)
- d) Bazaar

Ответ: c

9. **Что делает команда git merge?**

- a) Объединяет изменения из одной ветки в другую
- b) Удаляет ветку
- c) Создаёт новый репозиторий
- d) Откатывает изменения

Ответ: a

10. **Какой инструмент используется для управления зависимостями в проектах .NET?**

- a) NuGet
- b) Maven
- c) pip
- d) npm

Ответ: a

Лабораторные работы:

1. Настройка локального репозитория Git, выполнение commit, push.
2. Работа с ветками (создание, слияние, разрешение конфликтов).
3. Разработка структуры проекта в Visual Studio.
4. Интеграция модуля в существующий проект (командная работа).

Тема 2.2.2. Инструментарий тестирования и анализа качества

Вопросы для устного опроса:

1. Какие инструменты отладки предоставляет Visual Studio?
2. Что такое точка останова (breakpoint)? Как её установить?
3. В чём разница между отладочной (Debug) и релизной (Release) конфигурацией?
4. Для чего используются классы Debug и Trace в .NET?
5. Что такое «исключение»? Приведите примеры стандартных исключений.
6. Как обработать исключение с помощью try-catch-finally?
7. Что такое «инспекция кода»? Какие стандарты кодирования вы знаете?
8. Какие инструменты автоматического анализа кода существуют (StyleCop, SonarQube)?
9. Что такое модульное тестирование? Какие фреймворки используются?
10. Как провести тестирование интеграции вручную и автоматизированно?

Тестовые задания:

1. **Что делает отладчик (debugger)?**
 - a) Оптимизирует код

- b) Позволяет пошагово выполнять программу и смотреть значения переменных
- c) Компилирует программу
- d) Упаковывает программу в установщик
- Ответ: b*
- 2. **Какой блок используется в C# для перехвата исключений?**
 - a) catch
 - b) finally
 - c) throw
 - d) try*Ответ: a*
- 3. **Что такое «точка останова» (breakpoint)?**
 - a) Место в коде, где выполнение программы приостанавливается для отладки
 - b) Конечная точка программы
 - c) Ошибка компиляции
 - d) Метка для комментария*Ответ: a*
- 4. **Для чего используется класс Debug в .NET?**
 - a) Для вывода отладочной информации только в отладочной сборке
 - b) Для работы с базой данных
 - c) Для шифрования
 - d) Для создания потоков*Ответ: a*
- 5. **Что такое «обработка исключений»?**
 - a) Механизм реагирования на ошибки во время выполнения программы
 - b) Оптимизация кода
 - c) Компиляция программы
 - d) Документирование кода*Ответ: a*
- 6. **Какой инструмент в Visual Studio позволяет проверить код на соответствие стандартам именования?**
 - a) IntelliSense
 - b) StyleCop (или анализаторы кода)
 - c) Git
 - d) NuGet*Ответ: b*
- 7. **Что такое «условная компиляция»?**
 - a) Включение или исключение фрагментов кода в зависимости от директив препроцессора
 - b) Компиляция только изменённых файлов
 - c) Оптимизация циклов
 - d) Компиляция в режиме реального времени*Ответ: a*
- 8. **Какой фреймворк используется для модульного тестирования в .NET?**
 - a) NUnit / xUnit
 - b) JUnit
 - c) Selenium
 - d) PyTest*Ответ: a*
- 9. **Что такое «регрессионное тестирование»?**
 - a) Тестирование нового функционала
 - b) Повторное тестирование ранее работавших функций после изменений
 - c) Тестирование производительности

d) Тестирование безопасности

Ответ: b

10. Какой метод отладки позволяет выводить сообщения без остановки выполнения?

a) Debug.WriteLine

b) Console.ReadLine

c) MessageBox.Show

d) Thread.Sleep

Ответ: a

Лабораторные работы:

1. Отладка программы с использованием точек останова и окон Watch.
2. Использование классов Debug и Trace для вывода диагностики.
3. Обработка исключений (try-catch-finally, создание своего исключения).
4. Инспекция кода с помощью StyleCop или SonarQube.
5. Написание модульных тестов с использованием NUnit.
6. Тестирование интеграции (ручное и автоматизированное).

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

Вопросы к зачету по МДК.02.02

Вопросы к дифференцированному зачёту

Теоретические вопросы:

1. Назначение и основные функции систем контроля версий.
2. Чем отличается централизованная СКВ от распределённой? Примеры.
3. Опишите жизненный цикл файла в Git (от создания до отправки на сервер).
4. Что такое ветка в Git? Как создать, переключиться, удалить ветку?
5. Как разрешить конфликт при слиянии веток?
6. Какие инструменты отладки предоставляет Visual Studio?
7. В чём разница между отладочной (Debug) и релизной (Release) конфигурацией?
8. Для чего используются классы Debug и Trace в .NET?
9. Что такое исключительная ситуация? Назовите стандартные исключения C#.
10. Какие стандарты кодирования (именование, форматирование, комментирование) вы знаете?

Практические задания:

1. Создать локальный репозиторий Git, добавить файл, выполнить commit и push.
2. Создать ветку feature, сделать два коммита, затем слить с main.
3. Смоделировать конфликт при слиянии и разрешить его вручную.
4. В готовом проекте установить точку останова и отследить изменение переменной.
5. Написать код, генерирующий исключение DivideByZeroException, и перехватить его.
6. Использовать Debug.WriteLine для вывода информации в консоль (только в Debug).
7. Создать собственный класс исключения InvalidAgeException.
8. Провести инспекцию фрагмента кода (найти 3 нарушения стандартов).
9. Написать модульный тест для функции сложения чисел.
10. Настроить условную компиляцию: блок кода только в DEBUG-режиме.

Критерии:

- «5» – задание выполнено полностью, код рабочий, использованы все необходимые конструкции, отладка проведена, отчёт оформлен.
- «4» – задание выполнено, но есть мелкие недочёты.
- «3» – задание выполнено частично, есть ошибки.
- «2» – задание не выполнено.

7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Основные печатные издания

1. Рудаков А. В. Технология разработки программных продуктов: учебник для студентов среднего профессионального образования. – М.: Издательский центр «Академия», 2018.

Основные электронные издания

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1794453> – Режим доступа: по подписке.

Дополнительные источники

1. Зубкова Т.М. Технология разработки программного обеспечения. – Санкт-Петербург: Лань, 2021. – 252 с.

Интернет-ресурсы

- <http://www.intuit.ru> – Интернет-университет информационных технологий (ИНТУИТ.ру)
- <http://programm.ws/index.php> – Учебники по программированию
- <https://metanit.com> – Сайт по программированию