

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА
по профессиональному модулю
ПМ.04 РАЗРАБОТКА, ОТЛАДКА, ПРОВЕРКА РАБОТОСПОСОБНОСТИ,
МОДИФИКАЦИЯ КОМПЬЮТЕРНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
основной профессиональной образовательной программы (ОПОП-П)
по специальности СПО
10.02.05 Обеспечение информационной безопасности автоматизированных систем
(в рамках дополнительного профессионального блока ООО «ЭН+ДИДЖИТАЛ»)
Государственное бюджетное профессиональное образовательное учреждение
Краснодарского края
«Усть-Лабинский социально-педагогический колледж»

Усть-Лабинск, 2025 г

СОДЕРЖАНИЕ

1. Формы контроля и оценивания элементов профессионального модуля
2. Результаты освоения модуля, подлежащие проверке на экзамене (квалификационном)
 - 2.1. Общие/профессиональные компетенции, проверяемые дополнительно
 - 2.2. Требования к портфолио
3. Оценка освоения профессионального модуля
 - 3.1. Типовые задания для текущего контроля по МДК.04.01
 - 3.2. Типовые задания для рубежного контроля по МДК.04.01
 - 3.3. Типовые задания для промежуточного контроля по МДК.04.01
4. Требования к дифференцированному зачету по учебной практике
 - 4.1. Оценочные материалы (собеседование)
 - 4.2. Форма аттестационного листа (из дневника по практике)
5. Структура контрольно-оценочных материалов для экзамена (квалификационного)
6. Ведомость к экзамену квалификационному

1. Формы контроля и оценивания элементов профессионального модуля

Элемент модуля	Форма контроля и оценивания
МДК.04.01. Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения	Дифференцированный зачет (7 семестр – 34 часа), Экзамен (7 семестр – 66 часов, согласно учебному плану)
УП.04. Учебная практика	Дифференцированный зачет (7 семестр – 144 часа)
Производственная практика (преддипломная) (ПДП)	Дифференцированный зачет (8 семестр – 144 часа)
Профессиональный модуль ПМ.04	Экзамен (квалификационный) (8 семестр – 6 часов)

2. Результаты освоения модуля, подлежащие проверке на экзамене (квалификационном)

В результате аттестации по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Профессиональные и общие компетенции	Показатели оценки результата
ПК 4.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Составляет формализованные описания решений поставленных задач; разрабатывает алгоритмы (блок-схемы, псевдокод); проверяет корректность алгоритмов на контрольных примерах
ПК 4.2. Разрабатывать программные модули в соответствии с техническим заданием	Создает программный код на выбранном языке программирования (Python/C#) в соответствии с ТЗ; оптимизирует код; соблюдает сроки

	выполнения
ПК 4.3. Выполнять отладку программных модулей с использованием специализированных программных средств	Использует отладчик (breakpoints, трассировка, watch); анализирует сообщения об ошибках; исправляет логические и синтаксические ошибки
ПК 4.4. Выполнять тестирование программных модулей	Разрабатывает тестовые наборы данных (позитивные и негативные сценарии); проводит модульное тестирование (unit tests); документирует результаты
ПК 4.5. Осуществлять рефакторинг и оптимизацию программного кода	Выявляет признаки «плохого» кода (code smells); применяет техники рефакторинга; использует систему контроля версий (Git); оптимизирует код по скорости и памяти
ПК 4.6. Разрабатывать модули программного обеспечения для мобильных платформ (вариативная часть по заказу работодателя)	Разрабатывает простые мобильные приложения (Android/Kotlin); отлаживает на эмуляторе и реальном устройстве; собирает APK-файл
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	Распознает задачу, определяет этапы решения; составляет план; оценивает результат
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Определяет задачи и источники поиска информации; структурирует и интерпретирует информацию (документация, Stack Overflow, GitHub)
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие	Участвует в хакатонах, олимпиадах по программированию; ведет портфолио проектов на GitHub
ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	Участвует в парном программировании, code review; презентует результаты работы
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	Грамотно оформляет техническую документацию; комментирует код; корректно отвечает на вопросы
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей	Соблюдает нормы профессиональной этики; понимает значимость импортозамещения в IT-сфере
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях	Эргономично организует рабочее место; соблюдает режим труда и отдыха (профилактика «синдрома программиста»)
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности	Выполняет гимнастику для глаз и спины; соблюдает СанПиН при работе за ПК
ОК 09. Использовать информационные технологии в профессиональной деятельности	Применяет современные IDE, системы контроля версий, средства автоматизации тестирования
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке	Читает техническую документацию (API, мануалы) на русском и английском языках; оформляет код на английском

2.1. Общие/профессиональные компетенции, проверяемые дополнительно

ОК	Основные показатели результата	Портфолио	Курсовое проектирование	Промежуточная аттестация по практике
ОК 01	Выбранный способ решения задачи аргументирован; алгоритм оптимален	—	+	—
ОК 03	Наличие дипломов/сертификатов (олимпиады, хакатоны); проекты на GitHub; положительный отзыв с практики	+	—	+
ОК 05	Пояснительная записка (отчет) оформлена грамотно; код содержит осмысленные комментарии и названия переменных	—	+	—
ОК 09	Использование Git, IDE, средств отладки и профилирования	—	+	—
ОК 10	Список литературы содержит документацию на английском языке; код и комментарии на английском	—	+	—

2.2. Требования к портфолио

Тип портфолио: смешанный

Состав портфолио:

- дипломы, грамоты и сертификаты участия в олимпиадах по программированию, хакатонах, CTF (Capture The Flag);
- сертификаты о прохождении онлайн-курсов по программированию (Stepik, Coursera, Skillbox и др.);
- отзывы, характеристики с производственных практик;
- разработанные проекты (исходные коды программ на GitHub/GitLab, включая проекты по учебной практике);
- отчеты по учебной и производственной практике;
- ссылки на опубликованные мобильные приложения (при наличии).

3. Оценка освоения профессионального модуля

3.1. Типовые задания для текущего контроля по МДК.04.01 «Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения»

Вопросы для устного опроса по темам

Критерии оценки:

- «Отлично» – полный, развернутый ответ, четкая структура, грамотная речь, приведение примеров кода, демонстрация понимания.
- «Хорошо» – полный ответ, но допущены неточности в определениях или неполные примеры.
- «Удовлетворительно» – неполный ответ, нарушения логики, слабое знание синтаксиса.
- «Неудовлетворительно» – неполный ответ, грубые ошибки, отсутствие знаний.

Тема 1. Основы алгоритмизации

1. Понятие алгоритма. Свойства алгоритмов (детерминированность, конечность, массовость, результативность).
2. Способы описания алгоритмов (словесный, графический (блок-схемы), псевдокод, программный).
3. Основные алгоритмические конструкции: следование (линейные алгоритмы).
4. Ветвление (условный оператор if-else). Типы ветвлений.
5. Циклы: с предусловием (while), с постусловием (do-while), с параметром (for).

6. Вспомогательные алгоритмы: процедуры и функции. Параметры и возвращаемые значения.
7. Блок-схемы: назначение, основные элементы (начало/конец, ввод/вывод, действие, условие).
8. Проверка корректности алгоритмов на контрольных примерах (трассировка).

Тема 2. Введение в языки программирования

1. Классификация языков программирования (компилируемые vs интерпретируемые, высокого/низкого уровня).
2. Обзор языков Python, C++, C#, Java. Рекомендованный язык для освоения (Python и/или C#).
3. Структура программы на Python/C# (импорт библиотек, объявление переменных, функции, точка входа).
4. Переменные и типы данных (int, float, bool, str, char).
5. Операторы: арифметические (+, -, *, /, %, **), сравнения (==, !=, <, >, <=, >=), логические (and, or, not).
6. Ввод и вывод данных (input(), print() в Python; Console.ReadLine() в C#).
7. Форматированный вывод (f-строки в Python, String.Format в C#).

Тема 3. Объектно-ориентированное программирование (ООП)

1. Парадигма ООП: основные понятия (класс, объект, поле, метод).
2. Конструктор и деструктор. Назначение, примеры.
3. Инкапсуляция: модификаторы доступа (public, private, protected). Свойства (get/set).
4. Наследование: базовый и производный классы. Переопределение методов (override).
5. Полиморфизм: виртуальные функции, абстрактные классы и интерфейсы.
6. Отличие класса от объекта. Примеры.

Тема 4. Отладка программного кода

1. Типы ошибок: синтаксические, логические, времени выполнения.
2. Понятие отладки. Инструменты отладчика: точки останова (breakpoints).
3. Трассировка: шаг с заходом (Step Into), шаг с обходом (Step Over), шаг с выходом (Step Out).
4. Просмотр значений переменных (Watch), стек вызовов (Call Stack).
5. Print-отладка (print() в Python, Console.WriteLine() в C#).
6. Обработка исключений: конструкция try-catch-finally. Примеры.

Тема 5. Тестирование программных модулей

1. Цели и задачи тестирования программного обеспечения.
2. Уровни тестирования: модульное (unit testing), интеграционное, системное.
3. Виды тестирования: функциональное, нагрузочное, регрессионное.
4. Тестовые наборы данных: позитивные и негативные сценарии.
5. Понятие тестового покрытия (code coverage).
6. Фреймворки для модульного тестирования (unittest в Python, NUnit в C#).

Тема 6. Работа с базами данных в приложениях

1. Подключение к базам данных из приложений. Библиотеки (SQLite, [ADO.NET](#), SQLAlchemy).
2. Выполнение SQL-запросов (SELECT, INSERT, UPDATE, DELETE) из программы.
3. Параметризованные запросы: защита от SQL-инъекций.
4. ORM (Object-Relational Mapping): основные принципы, преимущества и недостатки.
5. Понятие CRUD (Create, Read, Update, Delete). Примеры.

Тема 7. Рефакторинг и оптимизация кода

1. Понятие рефакторинга. Отличие рефакторинга от оптимизации.
2. Признаки «плохого» кода (code smells): дублирование кода, длинные методы, большие классы, длинный список параметров.

3. Техники рефакторинга: переименование (rename), выделение метода (extract method), удаление дублирования (remove duplication), замена магического числа константой.
4. Оптимизация по скорости выполнения и потреблению памяти.
5. Профилирование кода: назначение, инструменты.
6. Системы контроля версий: Git (init, add, commit, push, pull, branch, merge).

Тема 8. Разработка для мобильных платформ (вариативная часть)

1. Особенности мобильной разработки (ограниченные ресурсы, сенсорный интерфейс, работа с датчиками).
2. Обзор мобильных ОС: Android и iOS.
3. Среда разработки Android Studio. Структура проекта.
4. Основы языка Kotlin (или Java) для Android.
5. Компоненты Android-приложения: Activity, Intent, Layout (XML).
6. Жизненный цикл Activity (onCreate, onStart, onResume, onPause, onStop, onDestroy).
7. Отладка на эмуляторе и реальном устройстве. Сборка APK-файла.

3.2. Типовые задания для рубежного контроля по МДК.04.01

Типовая контрольная работа

Тема: «Основы алгоритмизации и программирования»

Задание. Ответить письменно на поставленные вопросы.

Вариант 1

1. Дайте определение алгоритма. Перечислите свойства алгоритмов.
2. Опишите словесно и изобразите в виде блок-схемы алгоритм нахождения наибольшего числа из трех.
3. Напишите фрагмент кода на Python (или C#) для вычисления факториала числа n с использованием цикла for.

Вариант 2

1. Назовите и охарактеризуйте три основные алгоритмические конструкции.
2. Опишите словесно и изобразите в виде блок-схемы алгоритм проверки, является ли число простым.
3. Напишите фрагмент кода на Python (или C#) для поиска суммы всех четных чисел от 1 до N.

Вариант 3

1. Что такое вспомогательный алгоритм? Приведите пример процедуры/функции.
2. Опишите словесно и изобразите в виде блок-схемы алгоритм сортировки массива методом пузырька.
3. Напишите фрагмент кода на Python (или C#) для реверсирования строки.

Критерии оценки (аналогичны образцу ПМ.03):

- «Отлично» – полное, развернутое решение, корректная блок-схема, код без синтаксических ошибок.
- «Хорошо» – полное решение, но есть 1-2 неточности в блок-схеме или в коде.
- «Удовлетворительно» – решение неполное, блок-схема или код имеют существенные ошибки.
- «Неудовлетворительно» – решение отсутствует или полностью неверное.

Типовой тест для рубежного контроля

I. Что из перечисленного НЕ является свойством алгоритма?

1. дискретность
2. массовость
3. оптимальность ✓
4. результативность

II. Какой символ в блок-схеме отображает условие (ветвление)?

1. прямоугольник

2. параллелограмм
3. ромб ✓
4. овал

III. Какой цикл выполнится хотя бы один раз независимо от условия?

1. for
2. while
3. do-while ✓
4. foreach

IV. Расшифруйте аббревиатуру ООП:

Ответ: Объектно-ориентированное программирование

V. Выберите верные утверждения об инкапсуляции:

1. это механизм создания нового класса на основе существующего
2. это сокрытие внутренних данных и предоставление доступа через методы ✓
3. это способность объекта выбирать подходящий метод в зависимости от контекста
4. для реализации используются модификаторы доступа (private, public, protected) ✓

VI. Установите соответствие между типом ошибки и ее примером:

1) синтаксическая	A) деление на ноль
2) логическая	B) пропущена точка с запятой
3) времени выполнения	C) неверное условие в цикле (бесконечный цикл)

Правильный ответ: 1-B, 2-C, 3-A

VII. Какой фреймворк используется для модульного тестирования в Python?

1. NUnit
2. JUnit
3. unittest ✓
4. Mocha

VIII. Что такое SQL-инъекция? (Запишите ответ)

Ответ: это вид атаки на веб-приложения, при которой злоумышленник внедряет произвольный SQL-код в запрос для получения несанкционированного доступа к базе данных или выполнения вредоносных операций.

IX. Какой оператор Git используется для сохранения изменений в локальном репозитории?

1. git push
2. git commit ✓
3. git add
4. git merge

X. Назовите не менее 3 техник рефакторинга:

Ответ: переименование (rename), выделение метода (extract method), удаление дублирования (remove duplication), инлайн (inline), замена магического числа константой.

3.3. Типовые задания для промежуточного контроля по МДК.04.01

Вопросы для подготовки к экзамену (7 семестр, 66 часов)

1. Понятие алгоритма. Свойства алгоритмов.
2. Способы описания алгоритмов. Блок-схемы: элементы, правила построения.
3. Алгоритмическая конструкция «следование» (линейные алгоритмы).
4. Алгоритмическая конструкция «ветвление» (полное и неполное). Примеры.
5. Алгоритмическая конструкция «цикл»: цикл с предусловием (while), с постусловием (do-while), с параметром (for).
6. Вспомогательные алгоритмы: процедуры и функции. Передача параметров по значению и по ссылке.
7. Рекурсивные алгоритмы (факториал, числа Фибоначчи).
8. Классификация языков программирования (компилируемые/интерпретируемые, уровни).

9. Структура программы на Python (импорт, переменные, функции, main).
10. Типы данных в Python/C#: целые (int), вещественные (float), логические (bool), строковые (str).
11. Операторы ветвления: if-elif-else.
12. Операторы цикла: for, while. Операторы break и continue.
13. Списки (list) в Python: создание, индексация, срезы, методы (append, insert, remove).
14. Словари (dict) в Python: создание, обращение по ключу, методы (keys, values, items).
15. Кортежи (tuple) и множества (set) в Python.
16. Функции: определение (def), аргументы по умолчанию, return. Область видимости переменных.
17. Объектно-ориентированное программирование: классы и объекты.
18. Конструктор (__init__ в Python, конструктор в C#).
19. Инкапсуляция: модификаторы доступа (public, private, protected). Свойства (@property, get/set).
20. Наследование: базовый и производный классы. Переопределение методов (override).
21. Полиморфизм: виртуальные функции, абстрактные классы и интерфейсы.
22. Обработка исключений: try-except-finally (Python), try-catch-finally (C#).
23. Отладка: точки останова, трассировка, watch, call stack.
24. Типы ошибок: синтаксические, логические, времени выполнения. Примеры.
25. Модульное тестирование (unit testing). Фреймворк unittest в Python.
26. Разработка тестовых наборов данных (позитивные/негативные сценарии).
27. Подключение к базам данных из приложений. Библиотека sqlite3 в Python.
28. Выполнение SQL-запросов (SELECT, INSERT, UPDATE, DELETE) из программы. Параметризованные запросы.
29. Понятие SQL-инъекции и способы защиты.
30. ORM (Object-Relational Mapping) на примере SQLAlchemy или Entity Framework.
31. Рефакторинг: понятие, цели. Признаки «плохого» кода (code smells).
32. Техники рефакторинга (rename, extract method, remove duplication, inline).
33. Оптимизация кода по скорости и памяти. Профилирование.
34. Системы контроля версий: Git. Основные команды (init, add, commit, push, pull, clone, branch, merge).
35. Работа с удаленными репозиториями (GitHub, GitLab).
36. Основы разработки для Android: среда Android Studio.
37. Язык Kotlin для Android: основные конструкции (переменные, функции, классы).
38. Компоненты Android-приложения: Activity, Intent, Layout (XML).
39. Жизненный цикл Activity (onCreate, onStart, onResume, onPause, onStop, onDestroy).
40. Отладка на эмуляторе и реальном устройстве. Сборка APK-файла.
41. Работа с датчиками мобильного устройства (GPS, акселерометр, камера).
42. Инструменты для автоматизации сборки (Gradle).
43. Особенности публикации приложений в Google Play.
44. Сравнительная характеристика мобильных ОС Android и iOS.
45. Принципы безопасного программирования (защита от переполнения буфера, валидация ввода).
46. Работа с файловой системой из программ (чтение/запись файлов).
47. Многопоточное программирование: потоки, синхронизация, гонки данных.
48. Асинхронное программирование: async/await.
49. Паттерны проектирования (одиночка, фабрика, наблюдатель) – общее знакомство.
50. Особенности командной разработки: code review, pull requests, Issues.

4. Требования к дифференцированному зачету по учебной практике

Дифференцированный зачет по учебной практике выставляется с учетом данных аттестационного листа (характеристики профессиональной деятельности обучающегося) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика (ООО «ЭН+ДИДЖИТАЛ»).

4.1. Оценочные материалы

Перечень вопросов к собеседованию по учебной практике (УП.04)

1. Краткая характеристика места практики (структура ИТ-отдела, используемые технологии).
2. Перечень языков программирования и сред разработки, используемых в организации.
3. С какими системами контроля версий вы работали на практике (Git/SVN)?
4. Какие алгоритмические задачи вы решали в ходе практики?
5. Приведите пример разработанного вами программного модуля (назначение, входные/выходные данные).
6. Какие инструменты отладки вы использовали при разработке?
7. Как вы проводили тестирование разработанных модулей (какие тестовые наборы данных использовали)?
8. Проводили ли вы рефакторинг существующего кода? Если да, какие техники применяли?
9. Работали ли вы с базами данных в своих проектах (подключение, выполнение запросов, ORM)?
10. Использовали ли вы профилировщик для оптимизации кода?
11. С какими системами автоматического тестирования вы познакомились?
12. Разрабатывали ли вы мобильные приложения (Android)? Если да, то какие.
13. Какие трудности возникли у вас при прохождении практики и как вы их преодолевали?
14. Какую документацию вы оформляли по результатам работы?
15. Какие новые технологии/языки программирования вы освоили за время практики?
16. Какие принципы ООП вы применяли при разработке?
17. Как вы обеспечивали безопасность разработанного кода (защита от SQL-инъекций, валидация ввода)?
18. Взаимодействовали ли вы с другими разработчиками (парное программирование, code review)?
19. Какие методы обработки исключений вы использовали?
20. Оцените свою работу на практике по 5-балльной шкале и обоснуйте оценку.

4.2. Форма аттестационного листа (из дневника по практике)

АТТЕСТАЦИОННЫЙ ЛИСТ ПО УЧЕБНОЙ ПРАКТИКЕ

ФИО обучающегося: _____

Курс: _____ Группа: _____ Специальность: _____

10.02.05 Обеспечение информационной безопасности автоматизированных систем

Обучающийся(ая) успешно прошел(ла) учебную практику по профессиональному модулю ПМ.04 «Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения»

в объеме 144 часов с «» _____ 20 г. по «» _____ 20 г.

в организации / на предприятии: ООО «ЭН+ДИДЖИТАЛ» (или иная база практики)

Виды работ, выполненных обучающимся(ейся) во время практики:

№	Вид работы	Объем (час)	Качество
			(оптимальный/средний/допустимый уровень)

выполнения

- 1 Разработка алгоритмов в соответствии с техническим заданием
- 2 Написание программного кода на выбранном языке программирования
- 3 Отладка и тестирование разработанных модулей
- 4 Оформление программного кода в соответствии с требованиями (комментирование, форматирование)
- 5 Работа в системе контроля версий (Git)
- 6 Документирование результатов разработки
- 7 Подготовка отчета по практике

Руководитель от предприятия

(должность, фамилия, имя, отчество)

Дата «» _____ 20 г.

_____ / _____

(подпись) / (расшифровка подписи)

Руководитель практики от ГБПОУ КК «Усть-Лабинский социально-педагогический колледж»

(должность, фамилия, имя, отчество)

Дата «» _____ 20 г.

_____ / _____

(подпись) / (расшифровка подписи)

5. Структура контрольно-оценочных материалов для экзамена (квалификационного)

Экзамен (квалификационный) по ПМ.04 проводится в **8 семестре** (6 часов) в форме **выполнения практико-ориентированного задания** (разработка программного модуля в соответствии с техническим заданием) и **защиты портфолио проектов**.

Типовое задание для экзаменуемого

Текст задания (вариант 1):

ООО «ЭН+ДИДЖИТАЛ» разрабатывает систему управления складом готовой продукции. Вам поручено создать программный модуль для учета товаров на складе.

Требования к модулю:

1. Реализовать класс Product со следующими полями:
 - id (целое число, уникальный идентификатор)
 - name (строка, наименование товара)
 - quantity (целое число, количество на складе)

- price (вещественное число, цена за единицу)
2. Реализовать класс Warehouse, который содержит:
 - список товаров (коллекция/список)
 - метод add_product(product: Product) – добавление товара (если товар с таким id уже существует, увеличивать количество)
 - метод remove_product(product_id: int) – удаление товара
 - метод update_quantity(product_id: int, new_quantity: int) – обновление количества
 - метод get_total_value() – возвращает общую стоимость всех товаров на складе (сумма quantity * price)
 - метод find_by_name(keyword: str) – возвращает список товаров, содержащих keyword в имени
 3. Разработать модульные тесты (не менее 3 тестов) для проверки работоспособности класса Warehouse (использовать unittest в Python или NUnit в C#).
 4. Оформить программный код с комментариями (на английском или русском языке).
 5. Подготовить краткую пояснительную записку (титульный лист, цель, описание классов, результаты тестирования).

Дополнительное задание (на оценку «отлично»):
 Реализовать сохранение данных склада в файл (JSON или SQLite) и загрузку из файла.

Форма сдачи:

- Исходный код проекта (архив ZIP или ссылка на репозиторий GitHub).
- Пояснительная записка (PDF или DOCX).
- Устная защита (объяснение архитектуры, алгоритмов, демонстрация работы программы, ответы на вопросы комиссии).

Критерии оценки:

Критерий	Максимальный балл
Соответствие ТЗ (реализованы все требуемые методы)	10
Качество кода (комментирование, именование, отсутствие дублирования)	5
Работоспособность модуля (корректность выполнения операций)	10
Наличие и качество модульных тестов	5
Оформление пояснительной записки	5
Ответы на вопросы при защите	5
Дополнительное задание (файловое хранение)	+5
ИТОГО	40 / 45

Шкала перевода баллов в оценку:

- 36–45 баллов – «отлично»

- 27–35 баллов – «хорошо»
- 8–26 баллов – «удовлетворительно»
- 0–17 баллов – «неудовлетворительно»

Вид профессиональной деятельности «Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения» считается освоенным при получении оценки не ниже «удовлетворительно».

6. Ведомость к экзамену квалификационному

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ КРАСНОДАРСКОГО КРАЯ
«УСТЬ-ЛАБИНСКИЙ СОЦИАЛЬНО-ПЕДАГОГИЧЕСКИЙ КОЛЛЕДЖ»

ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ

Специальность: 10.02.05 Обеспечение информационной безопасности
автоматизированных систем

Профессиональный модуль: ПМ.04 «Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения»

Дата проведения: «» _____ 20 г.

№ п/п	ФИО обучающегося	Экзменационный билет	Коды проверяемых компетенций	Оценка (зачтено/не зачтено)	Подпись экзаменатора
1			ПК 4.1 – 4.6, ОК 01 – 10		
2			ПК 4.1 – 4.6, ОК 01 – 10		
3			ПК 4.1 – 4.6, ОК 01 – 10		

Заключение по результатам экзамена квалификационного:

Вид профессиональной деятельности «Разработка, отладка, проверка работоспособности, модификация компьютерного программного обеспечения»:

- Освоен с оценкой: _____
- Не освоен: _____

Члены экзаменационной (квалификационной) комиссии:

1. _____ (должность, ФИО) / _____
(подпись)
2. _____ (должность, ФИО) / _____
(подпись)
3. _____ (должность, ФИО) / _____
(подпись)

Представитель работодателя (ООО «ЭН+ДИДЖИТАЛ»):

_____ (должность, ФИО) / _____ (подпись)

Приложение. Пример структуры портфолио студента по ПМ.04

Раздел 1. Документы об участии в конкурсах и олимпиадах

- Дипломы/грамоты/сертификаты участника олимпиад по программированию
- Сертификаты Stepik/Coursera/Skillbox

Раздел 2. Отзывы и характеристики

- Характеристика с учебной практики (ООО «ЭН+ДИДЖИТАЛ»)
- Характеристика с производственной практики (преддипломной)

Раздел 3. Выполненные проекты (ссылки и скриншоты)

- Ссылка на репозиторий GitHub с проектами по МДК.04.01
- Скриншоты работающих программ (складской учет, CRUD-приложение)
- APK-файлы мобильных приложений (при наличии)

Раздел 4. Документация

- Отчеты по учебной и производственной практике
- Технические задания на разработанные модули
- Результаты тестирования (скриншоты пройденных unit test)

