

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«Саратовский государственный технический университет
имени Гагарина Ю.А.»**

Энгельсский технологический институт (филиал)

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по прохождению производственной практики (преддипломной) ПДП

для обучающихся специальности

09.02.07 Информационные системы и программирование

квалификация «Программист»

Энгельс 2025

Рассмотрено:

Протокол № 1 от «28» августа 2025 г.

Председатель ПЦМК _____ / А.А. Сдобнова /

Методические указания разработаны в соответствии с требованиями ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование» (приказ Минпросвещения России № 379 от 09.12.2016 с изменениями), рабочей программы производственной практики (преддипломной) и КОД 09.02.07-2-2026 демонстрационного экзамена по квалификации «Программист».

Составитель: преподаватель ОСПДО Сдобнова А.А.

Рецензент: _____

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ	5
2 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРАКТИКИ.....	6
3 СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИКИ	8
4 ОРГАНИЗАЦИЯ И РУКОВОДСТВО ПРАКТИКОЙ	8
5 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЁТНОЙ ДОКУМЕНТАЦИИ	9
6 ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	10
7 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ	10
8 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ	11
9 МЕТОДИКА ВЫПОЛНЕНИЯ ЗАДАНИЙ ПРАКТИКИ.....	13
Задание 1 (ПК 1.1) — Алгоритмы разработки модулей	13
Задание 2 (ПК 1.2) — Разработка кода по техническому заданию	16
Задание 3 (ПК 1.3) — Отладка модулей.....	22
Задание 4 (ПК 1.4) — Тестирование модулей	24
Задание 5 (ПК 1.5) — Рефакторинг и оптимизация	25
Задание 6 (ПК 1.6) — Мобильный модуль.....	27
Задание 7 (ПК 2.1) — Требования к модулям.....	28
Задание 8 (ПК 2.2) — Интеграция модулей и СКВ.....	30
Задание 9 (ПК 2.3) — Отладка интегрированной системы	31
Задание 10 (ПК 2.4) — Тестовые наборы и сценарии.....	31
Задание 11 (ПК 2.5) — Инспектирование кода.....	32
Задание 12 (ПК 4.1) — Инсталляция и настройка ПО	33

Задание 13 (ПК 4.2) — Эксплуатационные характеристики.....	33
Задание 14 (ПК 4.3) — Модификация ПО.....	34
Задание 15 (ПК 4.4) — Защита ПО	34
Задание 16 (ПК 11.1) — Сбор информации для БД	35
Задание 17 (ПК 11.2) — Проектирование БД	35
Задание 18 (ПК 11.3) — Объекты БД	36
Задание 19 (ПК 11.4) — Реализация БД в СУБД.....	36
Задание 20 (ПК 11.5) — Администрирование БД	37
Задание 21 (ПК 11.6) — Защита информации в БД	37

ВВЕДЕНИЕ

Производственная практика (преддипломная) ПДП является завершающим этапом практической подготовки обучающихся по специальности 09.02.07 «Информационные системы и программирование» и проводится в соответствии с ФГОС СПО, утверждённым приказом Минпросвещения России от 09.12.2016 № 379, Положением о практической подготовке обучающихся (приказ Минобрнауки России и Минпросвещения России от 05.08.2020 № 885/390), а также рабочей программой практики.

Практика направлена на углубление первоначального практического опыта обучающегося, развитие общих и профессиональных компетенций, проверку его готовности к самостоятельной трудовой деятельности, подготовку к выполнению дипломного проекта и прохождению государственной итоговой аттестации в форме демонстрационного экзамена по КОД 09.02.07-2-2026.

Объём практики составляет 144 академических часа (4 недели). Форма промежуточной аттестации — дифференцированный зачёт.

1 ЦЕЛИ И ЗАДАЧИ ПРАКТИКИ

Цель практики — формирование у обучающихся общих (ОК 01–ОК 11) и профессиональных компетенций, соответствующих видам деятельности «Разработка модулей программного обеспечения для компьютерных систем» (ВД.1), «Осуществление интеграции программных модулей» (ВД.2), «Сопровождение и обслуживание программного обеспечения компьютерных систем» (ВД.4), «Разработка, администрирование и защита баз данных» (ВД.11).

Задачи практики:

- закрепление и углубление теоретических знаний, полученных при изучении профессиональных модулей ПМ.01, ПМ.02, ПМ.03, ПМ.04;
- приобретение практического опыта разработки, отладки, тестирования и сопровождения программного обеспечения;

- сбор, систематизация и анализ материалов для выполнения дипломного проекта;
- подготовка к демонстрационному экзамену по КОД 09.02.07-2-2026;
- формирование готовности к самостоятельной профессиональной деятельности.

2 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРАКТИКИ

В результате прохождения практики обучающийся должен освоить следующие профессиональные компетенции:

Код	Наименование компетенции
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием.
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.4	Выполнять тестирование программных модулей.
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода.
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ.
ПК 2.1	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации.
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение.
ПК 2.3	Выполнять отладку программного модуля с

	использованием специализированных программных средств.
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.
ПК 4.1	Осуществлять инсталляцию, настройку и обслуживание ПО компьютерных систем.
ПК 4.2	Осуществлять измерения эксплуатационных характеристик ПО компьютерных систем.
ПК 4.3	Выполнять работы по модификации отдельных компонент ПО.
ПК 4.4	Обеспечивать защиту ПО компьютерных систем программными средствами.
ПК 11.1	Осуществлять сбор, обработку и анализ информации для проектирования баз данных.
ПК 11.2	Проектировать базу данных на основе анализа предметной области.
ПК 11.3	Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.
ПК 11.4	Реализовывать базу данных в конкретной СУБД.
ПК 11.5	Администрировать базы данных.
ПК 11.6	Защищать информацию в базе данных с использованием технологий защиты информации.

3 СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИКИ

Общий объём практики — 144 часа. Распределение часов по разделам приведено в таблице.

Раздел / тема	Вид работ	Часы
Подготовительный этап	Инструктаж по охране труда и технике безопасности, согласование заданий	6
Темы 1–6 (ПМ.01)	Разработка, отладка, тестирование, рефакторинг, мобильная разработка	36
Темы 7–11 (ПМ.02)	Требования, интеграция, отладка, тестовые сценарии, инспектирование	30
Темы 12–15 (ПМ.04)	Инсталляция, эксплуатация, модификация, защита ПО	24
Темы 16–21 (ПМ.11)	Проектирование, разработка, реализация, администрирование и защита БД	36
Оформление отчёта	Систематизация материалов, оформление	6
Промежуточная аттестация	Дифференцированный зачёт	6
Итого		144

4 ОРГАНИЗАЦИЯ И РУКОВОДСТВО ПРАКТИКОЙ

Практика организуется в соответствии с Положением о практической подготовке обучающихся на базе профильных организаций, заключивших с институтом договор о практической подготовке. Руководство практикой

осуществляют руководитель от института (преподаватель ОСПДО) и руководитель от профильной организации.

Обязанности обучающегося: своевременно прибыть на базу практики; пройти инструктажи; выполнять задания индивидуального задания; вести дневник практики; соблюдать правила внутреннего распорядка организации; по окончании практики представить отчёт, дневник и отзыв-характеристику.

5 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЁТНОЙ ДОКУМЕНТАЦИИ

Отчёт по практике оформляется в соответствии с ГОСТ 7.32-2017 «Отчёт о научно-исследовательской работе. Структура и правила оформления», ГОСТ Р 7.0.5-2008 «Библиографическая ссылка» и ГОСТ 2.105-2019 «ЕСКД. Общие требования к текстовым документам».

Основные требования:

- формат бумаги А4, ориентация книжная;
- поля: левое — 30 мм, правое — 15 мм, верхнее и нижнее — 20 мм;
- шрифт Times New Roman, 14 пт, межстрочный интервал 1,5; абзацный отступ 1,25 см; выравнивание по ширине;
- нумерация страниц сквозная, арабскими цифрами, в центре нижнего поля; титульный лист не нумеруется, но включается в общую нумерацию;
- заголовки разделов — с новой страницы, прописными буквами, без точки в конце;
- иллюстрации и таблицы нумеруются арабскими цифрами в пределах раздела, подписываются (Рисунок 1.1 — Наименование; Таблица 1.1 — Наименование);
- библиографический список оформляется по ГОСТ Р 7.0.100-2018.

Структура отчёта: титульный лист; индивидуальное задание; дневник практики; содержание; введение; основная часть (по темам индивидуального

задания); заключение; список использованных источников; приложения; отзыв-характеристика с базы практики.

6 ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Промежуточная аттестация по практике проводится в форме дифференцированного зачёта. Оценка выставляется с учётом: полноты и качества выполнения индивидуального задания; качества оформления отчёта и дневника; отзыва-характеристики руководителя от профильной организации; результатов защиты отчёта.

Шкала оценивания:

Оценка	Критерии
«отлично»	задание выполнено в полном объёме, отчёт оформлен без нарушений ГОСТ, на защите даны исчерпывающие ответы;
«хорошо»	задание выполнено в полном объёме, имеются незначительные замечания по оформлению;
«удовлетворительно»	задание выполнено не менее чем на 70 %, имеются замечания, в том числе по оформлению;
«неудовлетворительно»	задание выполнено менее чем на 70 %, отчёт не соответствует требованиям

7 ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств (ФОС) сформирован в соответствии с Порядком организации и осуществления образовательной деятельности по программам СПО (приказ Минпросвещения России от 24.08.2022 № 762) и включает: перечень компетенций с показателями освоения; типовые контрольные задания; критерии и шкалы оценивания; методические материалы.

Показатели оценки результатов практики:

Код ПК	Показатель оценки	Средство оценивания
ПК 1.1–1.6	Наличие в отчёте кода модулей, схем алгоритмов, результатов тестирования и рефакторинга	Проверка отчёта, собеседование
ПК 2.1–2.5	Документированные требования, журнал коммитов системы контроля версий, отчёт о code review	Анализ артефактов, защита
ПК 4.1–4.4	Скриншоты инсталляции, протоколы замеров, описание модификаций, отчёт о защите ПО	Проверка отчёта
ПК 11.1–11.6	ER-модель, физическая схема БД, SQL-скрипты, протоколы резервного копирования	Защита отчёта

8 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
2. ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование» (приказ Минпросвещения России от 09.12.2016 № 379).
3. Положение о практической подготовке обучающихся (приказ Минобрнауки России и Минпросвещения России от 05.08.2020 № 885/390).
4. КОД 09.02.07-2-2026 «Программист» (единые оценочные материалы демонстрационного экзамена).

5. ГОСТ 7.32-2017. Отчёт о научно-исследовательской работе. Структура и правила оформления.
6. ГОСТ Р 7.0.5-2008. Библиографическая ссылка. Общие требования и правила составления.
7. ГОСТ Р 7.0.100-2018. Библиографическая запись. Библиографическое описание.
8. ГОСТ 19.701-90 (ИСО 5807-85). ЕСПД. Схемы алгоритмов, программ, данных и систем.
9. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания.
10. ГОСТ Р ИСО/МЭК 12207-2010. Процессы жизненного цикла программных средств.

9 МЕТОДИКА ВЫПОЛНЕНИЯ ЗАДАНИЙ ПРАКТИКИ

В разделе приведены пошаговые рекомендации по выполнению каждого из 21 задания индивидуального задания. По каждому заданию указаны порядок действий, рекомендуемые программные средства и состав материалов, включаемых в отчёт.

Подготовительный этап

- получить у руководителя от института индивидуальное задание;
- прибыть на базу практики, пройти инструктаж по охране труда, ТБ, пожарной безопасности и правилам внутреннего распорядка под подпись в журнале;
- согласовать с руководителем от профильной организации график работы и тему дипломного проекта;
- в отчёт включить: запись об инструктаже, согласованный график и структуру предприятия.

Задание 1 (ПК 1.1) — Алгоритмы разработки модулей

Цель выполнения данного задания состоит в том, чтобы на основе технического задания научиться выделять функции будущего программного продукта, разбивать систему на отдельные модули и представлять логику работы этих модулей в виде алгоритмов и блок-схем. В самом файле практики по заданию 1 указано, что в отчете необходимо представить описание предметной области программного продукта и блок-схему разрабатываемого модуля.

После изучения задания нужно кратко описать предметную область. То есть необходимо объяснить, в какой сфере будет использоваться программа, какие процессы она автоматизирует и какую проблему решает. Например, если разрабатывается система учета заявок, в описании предметной области нужно

указать, что система предназначена для регистрации, обработки и контроля выполнения заявок пользователей.

Далее необходимо определить, какие именно функции должна выполнять программа. Для этого из технического задания выделяются функциональные требования. Это конкретные действия системы, например:

- ввод данных пользователем;
- проверка корректности введенных данных;
- сохранение информации в базе данных;
- поиск записей;
- редактирование данных;
- формирование отчета;
- удаление записи;
- вывод результата на экран.

Каждое требование лучше формулировать отдельно и понятно. Например: «Система должна позволять пользователю вводить данные о клиенте», «Система должна проверять заполнение обязательных полей», «Система должна сохранять данные в базе данных». Чем точнее будут сформулированы требования, тем легче будет строить алгоритм.

После этого нужно разделить программу на отдельные модули. Модуль — это самостоятельная часть программы, которая отвечает за конкретную функцию. Делать это нужно так, чтобы у каждого модуля была своя понятная задача. В отчете желательно показать не только список модулей, но и кратко написать, за что отвечает каждый из них.

Когда требования и модули определены, нужно разработать алгоритм работы одного из модулей. Алгоритм — это последовательность действий, которую должна выполнить программа для получения результата.

Алгоритм обычно строится по следующей логике:

1. Начало работы модуля.

2. Получение входных данных.
3. Проверка корректности данных.
4. Выполнение основного действия.
5. При необходимости принятие решения по условию.
6. Вывод результата.
7. Завершение работы модуля.

После составления алгоритма его необходимо представить графически в виде блок-схемы. В задании прямо указано, что нужно построить блок-схему разрабатываемого модуля.

При построении блок-схемы нужно использовать стандартные элементы:

1. **овал** — начало и конец алгоритма;
2. **параллелограмм** — ввод или вывод данных;
3. **прямоугольник** — выполнение действия или процесса;
4. **ромб** — проверка условия, ветвление;
5. **стрелки** — направление выполнения алгоритма.

При этом схема должна быть логичной и читаемой:

- движения лучше располагать сверху вниз;
- условия в ромбе должны иметь выходы «Да» и «Нет»;
- не нужно перегружать схему слишком мелкими деталями;
- названия действий должны быть краткими и понятными.

Для построения схем можно использовать:

1. **MS Visio** — удобно для официальных блок-схем;
2. **draw.io (diagrams.net)** — бесплатный и очень удобный вариант;
3. **Dia** — простой редактор схем;
4. при необходимости можно использовать Word с фигурами, но это менее удобно.

Если схема делается в draw.io, лучше сохранить ее в PNG и вставить в отчет как рисунок.

В отчете по этому заданию лучше сделать так:

Описание предметной области

Кратко описать, для какой организации, процесса или задачи создается программа.

Цель разрабатываемого модуля

Указать, какую функцию выполняет выбранный модуль.

Функциональные требования

Перечислить, что именно должен делать модуль.

Декомпозицию системы на модули

Описать, какие модули входят в программу и как они связаны между собой.

Алгоритм работы выбранного модуля

Текстово расписать последовательность действий.

Блок-схему алгоритма

Вставить рисунок со схемой.

Краткий вывод

Написать, что в ходе выполнения задания был разработан алгоритм программного модуля в соответствии с техническим заданием.

Делить задание на подразделы не надо.

Задание 2 (ПК 1.2) — Разработка кода по техническому заданию

Цель данного задания состоит в том, чтобы на основе ранее разработанного алгоритма реализовать программный модуль на выбранном языке программирования, соблюдая требования технического задания, правила кодирования и требования к документированию кода.

Сначала необходимо взять алгоритм, разработанный в задании 1, и определить, каким именно будет программный модуль. То есть нужно понять:

1. какую функцию выполняет модуль;

2. какие данные он получает на вход;
3. какие данные выдает на выход;
4. с какими другими модулями взаимодействует;
5. какие условия и ограничения заданы в техническом задании.

После этого выбирается язык программирования и среда разработки.

Обычно используют:

- Python в PyCharm или VS Code;
- C# в Visual Studio;
- Java в IntelliJ IDEA;
- JavaScript в VS Code;
- при необходимости можно использовать и другие языки, если это соответствует теме дипломного проекта или требованиям базы практики.

Важно, чтобы выбранный язык действительно подходил под тему работы. Например, если проект связан с настольным приложением под Windows, логично использовать C#. Если разрабатывается веб-модуль — JavaScript, Python.

После выбора языка нужно перевести алгоритм в программный код. Делать это лучше поэтапно.

1. Сначала создается сам проект в среде разработки. Нужно оформить папки, файлы и модули так, чтобы структура была понятной. Например:

- папка с основным кодом;
- папка с вспомогательными модулями;
- папка с тестами;
- файл конфигурации;
- файл запуска программы.

Если проект небольшой, достаточно выделить отдельный файл модуля. Если проект более сложный, лучше сразу использовать архитектурное разделение на слои:

- пользовательский интерфейс;
- бизнес-логика;
- работа с данными.

2. На втором этапе пишется код, реализующий алгоритм. Все действия программы должны соответствовать логике, разработанной ранее:

При написании кода обязательно нужно соблюдать стандарт кодирования.

Если используется C#, нужно соблюдать Microsoft C# Coding Conventions:

- имена классов, методов, свойств — в PascalCase;
- локальные переменные и параметры — в camelCase;
- осмысленные названия методов;
- фигурные скобки в едином стиле;
- разделение ответственности между классами;
- понятная структура пространства имен.

В любом языке программирования нужно соблюдать общие правила:

- код должен быть читаемым;
- названия переменных должны быть понятными;
- не следует использовать непонятные сокращения;
- большие участки кода нужно делить на функции;
- одна функция должна выполнять одну понятную задачу;
- повторяющийся код желательно выносить в отдельные функции.

В задании указано, что каждая функция должна быть снабжена комментариями. Это означает, что нужно документировать код не случайными фразами внутри программы, а оформлять комментарии правильно.

Если используется C#, применяются XML-комментарии:

- краткое описание метода;
- описание параметров;
- описание возвращаемого значения.

Смысл комментариев в том, чтобы другой разработчик по коду сразу понял назначение функции. Не нужно писать очевидные комментарии вроде «прибавляем 1». Нужно комментировать именно смысл и назначение.

При реализации модуля рекомендуется придерживаться следующей структуры:

1. Подключение библиотек.
2. Объявление констант и настроек.
3. Описание основных функций.
4. Вспомогательные функции.
5. Основной блок запуска программы или вызовов.

Так код будет проще читать и объяснять на защите.

В отчете по заданию 2 обязательно нужно представить:

- листинг ключевых модулей. Можно в приложении, если код объемный;
- скриншот среды разработки;
- описание архитектуры модуля.

Это означает, что в отчете должен быть не просто кусок кода, а полноценное описание выполненной работы.

Также как и в предыдущем задании, подразделы внутри задания делать не следует. Внутри желательно расположить материал в следующем порядке.

1. Назначение модуля

Нужно кратко написать, какой модуль был реализован и для чего он предназначен.

2. Выбранные средства разработки

Здесь указываются:

- язык программирования;
- среда разработки;
- используемые библиотеки;
- при необходимости — система контроля версий.

3. Описание архитектуры

Нужно объяснить, как организован код:

- какие есть файлы;
- какие функции реализованы;
- как модули взаимодействуют между собой;
- какие данные передаются между функциями.

4. Листинг кода

Вставляется ключевой фрагмент кода. Лучше включать не весь проект полностью, а самые важные части:

- основную функцию;
- обработку условий;
- основные методы класса;
- реализацию бизнес-логики.

При оформлении отчета, листинг можно вставить:

- моноширинным шрифтом;
- с нумерацией строк;
- с подписью вида:
«Листинг 1 — Код программного модуля регистрации заявки».

5. Скриншот IDE

Нужно вставить снимок экрана среды разработки, где видно:

- название проекта;
- код модуля;
- структуру файлов;
- иногда — результат запуска.

Под рисунком делается подпись, например: «Рисунок 2 — Разработка программного модуля в среде Visual Studio Code». Темные скриншоты заменяются на светлые. Необходимо перейти в светлую тему.

6. Краткий вывод

В конце раздела нужно написать, что программный модуль был реализован в соответствии с техническим заданием, а код соответствует требованиям выбранного стандарта кодирования.

Чтобы задание выглядело лучше и объемнее, можно добавить:

- схему структуры проекта;
- таблицу функций модуля;
- таблицу соответствия функций требованиям ТЗ;
- пример входных и выходных данных;
- фрагмент результата выполнения программы;
- краткое описание обработки ошибок.

Типичные ошибки

Часто в этом задании допускают такие ошибки:

- код пишется без связи с алгоритмом из задания 1;
- используются непонятные имена переменных;
- функции слишком большие и делают сразу всё;
- отсутствуют комментарии к функциям;
- код вставляется в отчет без пояснений;

- нет описания архитектуры;
- скриншот IDE неинформативный или в темной теме;
- листинг содержит слишком много второстепенного кода и мало основной логики.

Задание 3 (ПК 1.3) — Отладка модулей

Цель задания состоит в выявлении и устранении ошибок в разработанном программном модуле с использованием встроенных средств отладки среды разработки. В рамках выполнения задания необходимо проверить правильность работы программы, найти логические и технические ошибки, определить причины их возникновения и документально зафиксировать результаты отладки. В индивидуальном задании по практике указано, что по заданию 3 в отчете нужно представить тестовые покрытия с указанием процедур и функций для тестирования.

Сначала необходимо открыть реализованный на предыдущем этапе программный модуль в выбранной среде разработки. После этого следует определить проблемные участки программы, то есть функции, в которых возможны ошибки обработки данных, неверные условия, некорректная работа циклов, исключения или сбои при вводе данных. Далее нужно запустить программу в режиме отладки и установить точки останова в ключевых местах кода: перед вводом данных, перед выполнением условных операторов, перед записью данных и перед возвратом результата.

Во время отладки рекомендуется использовать следующие инструменты:

- точки останова;
- пошаговое выполнение программы;
- просмотр значений переменных;
- маркерные переменные;
- стек вызовов;

- окно локальных переменных;
- журнал ошибок и исключений.

Если в процессе выполнения задания были выявлены ошибки. Для каждой ошибки нужно описать:

- внешний признак ошибки;
- причину возникновения;
- участок кода, где обнаружена ошибка;
- способ исправления;
- результат после исправления.

Например, можно выявить такие ошибки:

- программа принимает пустой ввод и не выдает предупреждение;
- условие проверки диапазона значений работает неверно;
- при обращении к несуществующему объекту возникает исключение;
- данные сохраняются в неправильном формате;
- цикл не завершается при выполнении условия.

В отчете по этому заданию необходимо включить краткое описание используемой среды разработки, указание примененных инструментов отладки, таблицу вида «симптом, причина, исправление», а также скриншоты сессии отладки, на которых видны точки останова, значения переменных и процесс пошагового выполнения. Кроме того, поскольку в индивидуальном задании предусмотрено представление тестовых покрытий с указанием процедур и функций для тестирования, в отчет можно включить перечень функций, которые были проверены в процессе отладки.

Пример вывода для отчета можно сформулировать так: в ходе выполнения задания были использованы встроенные средства отладки среды разработки, выявлены и устранены ошибки логики и обработки данных, после чего работоспособность программного модуля была восстановлена.

Задание 4 (ПК 1.4) — Тестирование модулей

Цель задания состоит в проверке корректности работы программного модуля с помощью заранее подготовленных тестовых сценариев. В рамках этого задания необходимо убедиться, что модуль правильно обрабатывает входные данные, выдает ожидаемый результат, корректно реагирует на ошибочные значения и стабильно работает в различных условиях.

Для выполнения задания сначала необходимо определить, какие функции модуля подлежат тестированию. После этого разрабатываются тестовые случаи. На каждый модуль должно быть подготовлено не менее пяти тест-кейсов. Желательно, чтобы они включали:

- позитивные сценарии, когда пользователь вводит корректные данные;
- негативные сценарии, когда вводятся ошибочные данные;
- граничные значения;
- пустые значения;
- повторные вызовы функций.

Если используется C#, можно использовать NUnit. Если Java — JUnit. Даже если полноценный автоматизированный тест не делается, тестовый план и ручная проверка по кейсам обязательно должны быть отражены в отчете.

Тестовый план оформляется в виде таблицы, где указываются:

- идентификатор теста;
- наименование проверяемой функции;
- входные данные;
- ожидаемый результат;
- фактический результат;
- статус выполнения;
- тип тестирования.

- В отчете следует представить:
- тест-план в табличной форме;
- листинг тестов;
- краткое описание процесса тестирования;
- отчет о покрытии, если он формировался автоматически;
- вывод о корректности работы модуля.

Если программа успешно проходит все тесты, это нужно указать отдельно. Если некоторые тесты сначала завершались с ошибкой, а затем были исправлены, это также будет полезно показать, потому что демонстрирует реальную работу по качеству программного продукта.

Задание 5 (ПК 1.5) — Рефакторинг и оптимизация

Цель данного задания заключается в улучшении структуры программного кода без изменения его функциональности, а также в повышении эффективности работы программного модуля. В индивидуальном задании по практике указано, что в отчете по заданию 5 также необходимо представить план усовершенствования будущей системы.

Сначала необходимо проанализировать написанный ранее код и определить его слабые места. Обычно это:

- длинные и перегруженные функции;
- дублирование кода;
- непонятные названия переменных;
- лишние промежуточные переменные;
- сложные вложенные условия;
- неудобная или неправильная структура классов и методов, которая не соответствует выбранному архитектурному паттерну.

После анализа выполняется рефакторинг. Можно использовать такие приемы:

- **Выделение метода** — вынесение части кода в отдельный самостоятельный метод;
- **Переименование** — замена имён переменных, функций и классов на более понятные и осмысленные;
- **Замена числовых констант** — замена «магических чисел» именованными константами;
- **Встраивание переменной** — удаление лишних промежуточных переменных, которые используются только один раз;
- **Упрощение условий** — сокращение сложных логических выражений и вложенных ветвлений;
- **Устранение дублирования** — удаление повторяющихся фрагментов кода путём их объединения;
- **Перераспределение ответственности** — перенос методов и полей между классами так, чтобы каждый класс отвечал за одну задачу.

Желательно выполнять рефакторинг с помощью встроенных средств IDE, например Refactor - Rename или Extract Method. Это уменьшает риск ошибок и показывает владение современными инструментами разработки.

После рефакторинга необходимо провести оптимизацию. Для этого выполняется замер работы программы до и после изменений. Можно сравнить:

- время выполнения;
- объем кода;
- количество функций;
- количество строк в основном методе;
- цикломатическую сложность;
- скорость обработки данных.

В отчете рекомендуется показать код «до» и «после», дать таблицу сравнения метрик и отдельно написать, какие изменения были внесены и

почему. Также полезно включить краткий план усовершенствования системы, как и требуется в индивидуальном задании: например, выделение слоев архитектуры, подключение логирования, перенос конфигурации в отдельный файл, внедрение тестирования, добавление обработки исключений.

Задание 6 (ПК 1.6) — Мобильный модуль

Цель задания состоит в разработке программного модуля для мобильной платформы. В индивидуальном задании указано, что по результатам выполнения задания 6 в отчете нужно представить руководство пользователя.

Для выполнения задания необходимо выбрать платформу разработки. Это может быть:

1. Android Studio на Kotlin или Java;
2. Flutter;
3. React Native.

Далее нужно определить, какую именно функцию будет выполнять мобильный модуль. Это может быть:

- форма авторизации;
- экран списка объектов;
- модуль регистрации заявки;
- просмотр информации из базы данных;
- расчетный модуль;
- мобильный интерфейс для уже разработанной системы.

После этого создается пользовательский интерфейс.

Кроме интерфейса обязательно должна быть реализована хотя бы одна бизнес-функция, то есть не просто визуальный экран, а работающая логика: расчет, отправка данных, валидация, сохранение, авторизация или выборка данных.

Программу нужно запустить как минимум на эмуляторе. Если есть возможность, желательно показать запуск и на физическом устройстве. В отчете следует представить:

- скриншоты интерфейса;
- описание назначения экрана;
- фрагменты Activity, Fragment, Widget или соответствующих файлов;
- manifest-файл;
- краткое руководство пользователя с описанием порядка запуска и использования.

Задание 7 (ПК 2.1) — Требования к модулям

Цель задания состоит в формализации требований к программным модулям на основе анализа технической и проектной документации. В отчете должно быть представлено руководство оператора.

Сначала нужно изучить проектную и техническую документацию. Затем следует выделить в каждом модуле:

- функциональные требования;
- нефункциональные требования;
- ограничения;
- требования к интерфейсу;
- требования к взаимодействию модулей.

Функциональные требования описывают, что должна делать система. Нефункциональные определяют качество работы: производительность, надежность, безопасность, совместимость, удобство использования.

Далее требования нужно оформить в структурированном виде. Обычно указывают:

- идентификатор требования;

- источник требования;
- краткое содержание;
- приоритет;
- модуль, к которому оно относится.

После этого строятся UML-диаграммы, например:

Структурные диаграммы (описывают устройство системы):

- **Диаграмма классов** — показывает классы программы, их атрибуты, методы и связи между классами (наследование, ассоциация, зависимость). Основная диаграмма при объектно-ориентированном проектировании.
- **Диаграмма объектов** — похожа на диаграмму классов, но показывает конкретные экземпляры объектов в определённый момент работы программы.
- **Диаграмма компонентов** — отображает из каких модулей, библиотек и файлов состоит система и как они связаны между собой.
- **Диаграмма развёртывания** — показывает, на каких серверах и устройствах размещаются компоненты системы.

Поведенческие диаграммы (описывают работу системы):

- **Диаграмма последовательности** — показывает, в каком порядке объекты обмениваются сообщениями при выполнении конкретного сценария. Удобна для описания взаимодействия между модулями.
- **Диаграмма коммуникации** — аналогична диаграмме последовательности, но акцент делается не на порядке, а на связях между объектами.
- **Диаграмма состояний** — описывает, в каких состояниях может находиться объект и при каких условиях он переходит из одного

состояния в другое. Используется для моделирования жизненного цикла объекта.

- **Диаграмма взаимодействия** — обзорная схема, объединяющая несколько сценариев взаимодействия объектов в одном представлении.

Вы строите те диаграммы, которые необходимы вам для пояснений работы вашей системы.

В отчете нужно привести таблицу требований и соответствующую диаграмму. Дополнительно можно описать, каким пользователям предназначен модуль, и включить элементы руководства оператора, как указано в индивидуальном задании.

Задание 8 (ПК 2.2) — Интеграция модулей и СКВ

Цель задания состоит в объединении программных модулей в единую систему и использовании системы контроля версий для управления изменениями.

Для выполнения задания нужно:

- создать репозиторий в Git;
- настроить ветки main, develop, feature;
- подключить исходные файлы;
- выполнить сборку проекта;
- объединить модули;
- проверить совместную работу модулей;
- сделать серию коммитов.

Коммиты желательно оформлять осмысленно, например:

- feat: add registration module;
- fix: correct validation logic;
- refactor: split service layer;
- docs: update module setup guide.

В отчете нужно показать:

- скриншот git log;
- краткое описание ветвления;
- перечень основных коммитов;

- при наличии — merge request;
- описание установки и настройки программного модуля

Задание 9 (ПК 2.3) — Отладка интегрированной системы

Цель задания состоит в проверке совместной работы нескольких модулей после их интеграции. В задании следует представить листинг кода, исправленный после проведения тестирования.

На этом этапе уже важно не только то, как работает каждый модуль отдельно, но и как они взаимодействуют между собой. Нужно включить логирование, настроить вывод сообщений об ошибках и проверить:

- передачу данных между модулями;
- корректность вызова функций;
- обработку исключений;
- корректность ответа интегрированной системы.

Желательно воспроизвести 2–3 интеграционные ошибки, например:

- неверная передача параметров;
- отсутствие ожидаемого ответа от другого модуля;
- ошибки обращения к данным;
- конфликт форматов.

В отчете приводятся:

- фрагменты лог-файлов;
- таблица дефектов;
- исправленный листинг кода;
- краткое описание того, как были устранены найденные ошибки.

Задание 10 (ПК 2.4) — Тестовые наборы и сценарии

Цель задания состоит в разработке набора тестовых сценариев для комплексной проверки программного обеспечения. В задании указано, что в отчете необходимо представить нормативные документы, регламентирующие разработку кода.

На этом этапе нужно разработать сценарии нескольких типов:

- позитивные;

- негативные;
- граничные;
- нагрузочные;
- пользовательские.

Если тестируется API, можно использовать Postman. Если пользовательский интерфейс — Selenium. Для нагрузочных испытаний — JMeter.

В отчете полезно показать:

- матрицу покрытия требований тестами;
- перечень сценариев;
- результаты выполнения;
- таблицу успешно и неуспешно пройденных тестов;
- ссылки на стандарты и нормативные документы разработки, как указано в индивидуальном задании.

Задание 11 (ПК 2.5) — Инспектирование кода

Цель задания заключается в проверке качества программного кода на соответствие стандартам кодирования и общим требованиям к разработке. В отчете должна быть представлена таблица анализа соответствия стандартам кодирования компонентов программного модуля.

Для выполнения задания необходимо использовать чек-лист, в который можно включить:

- читаемость имен;
- единый стиль оформления;
- отсутствие дублирования;
- обработку ошибок;
- корректность комментариев;
- структуру методов;
- соблюдение архитектуры.

Желательно дополнительно использовать статические анализаторы, такие как SonarQube, Pylint, ESLint или ReSharper. В отчете следует показать:

- краткий отчет анализатора;
- таблицу нарушений;

- способы исправления;
- итоговую оценку соответствия кода стандартам.

Задание 12 (ПК 4.1) — Инсталляция и настройка ПО

Цель задания состоит в развертывании программного продукта на рабочем месте или тестовом стенде. В индивидуальном задании указано, что в отчете нужно представить перечень программного обеспечения, а также конфигурации ПК для корректной работы программного модуля.

Сначала нужно определить, какие компоненты требуются для запуска программы:

- операционная система;
- среда выполнения;
- библиотеки;
- база данных;
- зависимости.

Затем выполняется установка и настройка. В отчете приводятся:

- перечень необходимого программного обеспечения;
- требования к ПК;
- этапы установки;
- файл конфигурации;
- скриншоты установки и настройки.

Задание 13 (ПК 4.2) — Эксплуатационные характеристики

Цель задания заключается в измерении качества работы программного продукта. В отчете нужно представить эксплуатационные характеристики качества программного модуля.

На этом этапе следует измерить:

- скорость запуска;
- время выполнения операций;
- устойчивость к ошибкам;
- нагрузку на процессор;
- использование оперативной памяти;
- стабильность работы и другие метрики.

Измерения можно оформить в виде таблиц и графиков. В выводах нужно указать, удовлетворяет ли программа требованиям качества и какие параметры можно улучшить.

Задание 14 (ПК 4.3) — Модификация ПО

Цель задания состоит в внесении изменений в программный продукт в соответствии с новыми требованиями или потребностями пользователя. В отчете необходимо представить возможные модификации программного модуля.

Сначала формулируется **заявка на изменение**. Затем описывается, какие компоненты затрагивает изменение, какие риски возникают и какие действия будут выполнены. После внесения изменений обновляется версия программы и документация.

В отчете нужно показать:

- перечень изменений;
- возможные направления дальнейшей модификации;
- фрагменты измененного кода;
- обновленную инструкцию пользователя;
- при наличии — changelog.

Задание 15 (ПК 4.4) — Защита ПО

Цель задания состоит в обеспечении безопасности программного продукта. В отчете по этому заданию необходимо представить таблицу анализа рисков.

Для выполнения задания нужно рассмотреть основные угрозы:

- несанкционированный доступ;
- изменение данных;
- повреждение файлов;
- подмена компонентов;
- утечка конфиденциальной информации.

После этого выбираются меры защиты:

- цифровая подпись;

- обфускация;
- контроль целостности;
- ограничение доступа;
- проверка лицензии;
- шифрование данных.

Также можно применить модель STRIDE для анализа угроз. В отчете обязательно включается таблица анализа рисков, описание реализованных мер защиты и краткий вывод о повышении уровня безопасности системы.

Задание 16 (ПК 11.1) — Сбор информации для БД

Цель задания состоит в подготовке исходных данных для проектирования базы данных. В отчете нужно представить нормативные документы, регламентирующие разработку баз данных данной предметной области.

Сначала необходимо изучить предметную область:

- документы организации;
- существующие формы учета;
- правила обработки информации;
- типы пользователей;
- основные бизнес-процессы.

После этого выделяются сущности, атрибуты и связи между объектами. В отчете нужно показать перечень сущностей и привести нормативные документы, имеющие отношение к разработке БД в конкретной компании.

Задание 17 (ПК 11.2) — Проектирование БД

Цель задания — построение модели базы данных на основе анализа предметной области. В отчете необходимо представить концептуальную и логическую модели проектируемой базы данных.

Сначала строится концептуальная модель: сущности и связи между ними. Затем формируется логическая модель: атрибуты, ключи, типы связей, ограничения. После этого выполняется нормализация до 3НФ.

В отчете приводятся:

- концептуальная модель;
- логическая модель;
- описание ключевых сущностей;
- обоснование нормализации.

Задание 18 (ПК 11.3) — Объекты БД

Цель задания состоит в создании объектов базы данных. В отчете по этому этапу должна быть представлена физическая схема базы данных.

Для выполнения задания разрабатываются:

- таблицы;
- первичные ключи;
- внешние ключи;
- ограничения;
- индексы;
- представления;
- триггеры;
- хранимые процедуры.

В отчете нужно представить физическую схему базы данных и листинг SQL-скриптов создания основных объектов.

Задание 19 (ПК 11.4) — Реализация БД в СУБД

Цель задания состоит в создании и наполнении базы данных в конкретной системе управления. В отчете следует представить экранные снимки разработанных таблиц.

На этом этапе необходимо:

- создать базу данных в СУБД;
- выполнить SQL-скрипты;
- заполнить таблицы тестовыми данными (не менее 30 записей);
- проверить связи и ограничения;
- выполнить запросы разной сложности.

В отчете показываются:

- скриншоты таблиц;
- примеры SQL-запросов;

- результаты выполнения запросов;
- краткий комментарий по корректности структуры и наполнения.

Задание 20 (ПК 11.5) — Администрирование БД

Цель задания заключается в обеспечении администрирования и резервирования базы данных. В отчете необходимо описать процедуру резервного копирования и мониторинга базы данных.

Для выполнения задания нужно:

- создать пользователей и роли;
- настроить права доступа;
- выполнить резервное копирование;
- выполнить восстановление базы;
- проверить корректность восстановления;
- организовать мониторинг состояния БД.

В отчете следует привести:

- SQL-команды управления доступом;
- описание резервного копирования;
- скриншоты или журнал восстановления;
- описание мониторинга БД.

Задание 21 (ПК 11.6) — Защита информации в БД

Цель задания состоит в обеспечении информационной безопасности на уровне базы данных. В отчете необходимо описать методы и способы защиты базы данных.

На этом этапе рекомендуется рассмотреть:

- разграничение прав доступа;
- шифрование данных;
- аудит действий пользователей;
- парольные политики;
- резервирование;
- защиту от несанкционированного изменения информации;
- защиту учетных записей.

В отчете нужно подробно описать примененные меры защиты, привести примеры настроек безопасности, скриншоты и сделать вывод о том, как обеспечивается безопасность базы данных.

Оформление отчета

После выполнения всех заданий необходимо собрать материалы в единый отчет.

Структура отчета должна быть такой:

1. титульный лист;
2. индивидуальное задание;
3. дневник практики;
4. содержание;
5. введение;
6. основная часть по заданиям практики;
7. заключение;
8. список использованных источников;
9. приложения;
10. характеристика и аттестационный лист, которые не прошиваются и не считаются.

При оформлении нужно проверить:

- поля по ГОСТ;
- заполнение по ширине;
- шрифт Times New Roman 14;
- межстрочный интервал 1,5;
- нумерацию страниц;
- правильное оформление таблиц (сквозное нумерование, подпись справа вверху, допускается 12 шрифт) и рисунков (сквозное нумерование, по центру под рисунком, допускается 12 шрифт);
- корректное оформление библиографии.

Защита отчета

Защита отчета проходит в форме дифференцированного зачета.

Для защиты нужно подготовить краткий план выступления:

- цель практики;
- задачи практики;

- что было сделано по каждому блоку ПК;
- какой программный продукт или база данных разрабатывались;
- какие результаты получены;
- какие трудности возникли;
- что было улучшено в ходе практики.