

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«КУЗБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Т. Ф. ГОРБАЧЕВА»
Филиал КузГТУ в г. Белово

Кафедра инженерно-экономическая

МДК. 03.01 Моделирование и анализ программного обеспечения
Методические рекомендации
по выполнению самостоятельных работ
для специальности
09.02.07 «Информационные системы и программирование»

Составитель: Витвицкий М.Н.
Рассмотрены и утверждены на
заседании кафедры
Рекомендованы учебно-
методической комиссией
специальности 09.02.07
«Информационные системы и
программирование» в качестве
электронного издания для
использования в учебном
процессе
Протокол №4 от 11.12.2025

Белово 2025

СОДЕРЖАНИЕ

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	1
ПЛАНИРОВАНИЕ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	4
КОНТРОЛЬ РЕЗУЛЬТАТОВ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	6
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	7
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	15

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа обучающихся может рассматриваться как организационная форма обучения, обеспечивающих управление учебной деятельностью или деятельность обучающихся по освоению общих и профессиональных компетенций, знаний и умений учебной и научной деятельности без посторонней помощи.

В учебном процессе выделяют два вида самостоятельной работы: аудиторная, внеаудиторная.

Аудиторная самостоятельная работа по учебной дисциплине и профессиональному модулю выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется учащимся по заданию преподавателя, но без его непосредственного участия.

Самостоятельная работа обучающихся проводится с целью: систематизации и закрепления полученных теоретических знаний и

практических умений студентов;

углубления и расширения теоретических знаний;

формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;

развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности и организованности;

формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

развития исследовательских умений;

формирования общих и профессиональных компетенций.

ПЛАНИРОВАНИЕ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Преподавателем учебной дисциплины эмпирически определяются затраты времени на самостоятельное выполнение конкретного содержания учебного задания: на основании наблюдений за выполнением учащимися аудиторной самостоятельной работы, опроса студентов о затратах времени на то или иное задание, хронометража собственных затрат на решение той или иной задачи с внесением поправочного коэффициента из расчета уровня знаний и умений обучающихся.

При разработке рабочей программы по учебной дисциплине или профессиональному модулю при планировании содержания внеаудиторной самостоятельной работы преподавателей устанавливается содержание и объем теоретической учебной информации или практических заданий, которые выносятся на внеаудиторную самостоятельную работу, определяются формы и методы контроля результатов.

Содержание внеаудиторной самостоятельной работы определяется в соответствии с рекомендуемыми видами заданий согласно программе учебной дисциплины профессионального модуля.

Видами заданий для внеаудиторной самостоятельной работы могут быть:

- для овладения знаниями:* компетентностно-ориентированные задание, чтение текста (учебника, первоисточника, дополнительной литературы): составление плана текста; графическое изображение структуры текста; конспектирование текста; реферирование текста; выписки из текста; работа со словарями и справочниками, ознакомление с нормативными документами; учебно-исследовательская работа; использование аудио- и видеозаписей, компьютерной техники и Интернета и др.;

- для закрепления и систематизации знаний:* компетентностно-ориентированное задание, работа с конспектом лекции (обработка текста); повторная работа над учебным материалом (учебника, первоисточника, дополнительной литературы, аудио- и видеозаписей); составление плана и тезисов ответа; составление таблиц для систематизации учебного материала;

изучение нормативных материалов; ответы на контрольные вопросы; аналитическая обработка текста (аннотирование, рецензирование, реферирование, контент-анализ и др.); подготовка сообщений к выступлению на семинаре, конференции; подготовка рефератов, докладов; составление библиографии, тематических кроссвордов; тестирование и др.;

•*для формирования компетенций:* компетентностно-ориентированное задание, решение задач и упражнений по образцу; решение вариативных задач и упражнений; выполнение чертежей, схем; выполнение расчетно- графических работ; решение ситуационных педагогических задач; подготовка к деловым играм; проектирование и моделирование разных видов и компонентов профессиональной деятельности; подготовка курсовых работ; опытно-экспериментальная работа; упражнения на тренажере; упражнения спортивно-оздоровительного характера; рефлексивный анализ профессиональных умений с использованием аудио- и видеотехники и др.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать специфику специальности, изучаемой дисциплины, индивидуальные особенности студента.

При предъявлении видов заданий на внеаудиторную самостоятельную работу рекомендуется использовать дифференцированный подход к студентам. Перед выполнением студентами внеаудиторной самостоятельной работы преподаватель проводит инструктаж по выполнению задания, который включает цель задания, его содержание, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. В процессе инструктажа преподаватель предупреждает обучающихся о возможных типичных ошибках, встречающихся при выполнении задания. Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Самостоятельная работа может осуществляться индивидуально или группами обучающихся в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности уровня умений обучающихся.

Отчет по самостоятельной работе обучающихся предоставляется в электронном виде.

КОНТРОЛЬ РЕЗУЛЬТАТОВ ВНЕАУДИТОРНОЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия по дисциплине и внеаудиторную самостоятельную работу обучающихся по дисциплине, может проходить в письменной, устной или смешанной форме, с представлением продукта деятельности учащегося.

В качестве форм и методов контроля внеаудиторной самостоятельной работы обучающихся могут быть использованы, *зачеты, тестирование, самоотчеты, контрольные работы, защита творческих работ и др., которые могут осуществляться на учебном занятии или вне его (например, оценки за реферат).*

Критериями оценки результатов внеаудиторной самостоятельной работы обучающегося являются:

- уровень освоения учащимся учебного материала;
- умение обучающегося использовать теоретические знания при выполнении практических задач;
- сформированность общих и профессиональных компетенций;
- обоснованность и четкость изложения ответа;
- оформление материала в соответствии с требованиями.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

РЕФЕРАТ

Реферат (от латинского – сообщаю) – краткое изложение в письменном виде содержания научного труда (трудов), литературы по теме. Это самостоятельная научно-исследовательская работа, где раскрывается суть исследуемой проблемы, изложение материала носит проблемно-тематический характер, показываются различные точки зрения, а также собственные взгляды на проблему. Содержание реферата должно быть логичным.

Критерии оценки реферата:

- соответствие теме;
- глубина проработки материала;
- правильность и полнота использования источников;
- оформление реферата.

ДОКЛАД

Доклад – вид самостоятельной работы обучающихся, используется в учебных и внеклассных занятиях, способствует формированию навыков исследовательской работы, расширяет познавательные интересы, приучает практически мыслить. При написании доклада по заданной теме следует составить план, подобрать основные источники. Работая с источниками, попытаться систематизировать полученные сведения, сделать выводы и обобщения. В настоящее время в учебных заведениях доклады содержательно практически ничем не отличаются от рефератов. Структура и оформление доклада такое же, как в реферате.

Критерии оценки доклада:

- соответствие теме;
- глубина проработки материала;
- правильность и полнота использования источников;
- оформление доклада.

Самостоятельная работа состоит из 2 заданий:

1. Теоретическое задание (реферат, доклад на 10 стр. А4);
2. Практическое задание (выдается преподавателем индивидуально согласно перечню).

Оформление работы

На титульном листе посередине его записывается вид работы, ниже на 10 мм – её название строчными буквами, справа в нижнем углу – фамилия автора разработки, группа. В нижней части титульного листа посередине указывается год написания разработки.

При наборе рекомендуется использовать основные системные гарнитуры шрифта TimesNewRoman. Текст набирается с соблюдением следующих правил: не допускаются ручной набор нумерации в главах и абзацах (только автонумарация); два и более пробела между символами. При наборе должны различаться тире и дефисы; маркеры и другие знаки должны быть сохранены аналогичными на протяжении всего материала. Между инициалами и после них (перед фамилией) ставится неразрывный пробел.

Размеры полей «обычное»: верхнее 1 см, левое 2 см, нижнее 1 см, правое 1 см. Нумерация страниц – внизу «по центру» шрифтом 12 пт. гарнитуры шрифта TimesNewRoman, нумерация страниц записки сквозная, причем начинается простановка номеров со страницы «Содержание», с учетом всех впереди стоящих страниц, на которых номера не проставляются.

В рабочей программе:

Самостоятельная работа №1 «Сравнительный анализ офисных пакетов»	4
Самостоятельная работа №2 Сравнительный анализ браузеров»	4

Самостоятельная работа №3 «Анализ структуры заданного программного модуля»	6
--	---

Темы самостоятельной работы

№ раздела (темы)	Вопросы, выносимые на самостоятельное изучение	Количество часов
Тема 3.1.1. Задачи и методы моделирования и анализа программных продуктов.	Самостоятельная работа 1. Моделирование процесса.	2
	Самостоятельная работа 2. Сбор и приоритизация требований.	2
	Самостоятельная работа 3. Анализ данных для улучшения.	2

	Самостоятельная работа 4. Прототипирование экрана "Личный кабинет".	2
Тема 3.1.2 Организация ревьюирования. Инструментальные средства ревьюирования.	Самостоятельная работа 5. Анализ структуры заданного программного модуля	6

Описание выполнения задания

Задание к самостоятельной 1. Моделирование процесса.

Разрабатывается функционал.

Визуализировать основной процесс.

1. Определите ключевые шаги.

2. Постройте бизнес - модель.

3. Проанализируйте модель.

Укажите 1 "узкое место" (например, обязательная регистрация перед оплатой).

Предложите 1 улучшение (например, "Гостевая оплата").

Критерии оценки:

Четкость и полнота отображения процесса.

Корректное использование символов блок-схемы.

Практичность предложенного улучшения.

Задание к самостоятельной 2. Сбор и приоритизация требований.

Выявить и систематизировать потребности пользователей системы.

1. Определите минимум 2 группы пользователей.

2. Составьте список требований (по 3-4 для каждой группы).

3. Приоритизируйте требования по методу MoSCoW.

Must have (Обязательно), без этого система не работает.

Should have (Желательно), важно, но можно без первого релиза.

Could have (Возможно), улучшение, если останется время.

Won't have (Не сейчас), откладывается.

4. Обоснуйте 1 выбор из категории "Must have" для каждой группы.

Критерии оценки:

Релевантность требований целевым группам.

Четкое понимание и применение MoSCoW.

Логичность обоснования приоритетов.

Задание к самостоятельной 3. Анализ данных для улучшения.

В приложении пользователи редко используют функцию.

Задача, проанализировать возможные причины и предложить решения.

1. Сформулируйте 3 гипотезы о причинах низкой вовлеченности:

Функционал ограничен.

Нет персонализации под цели.

Интерфейс создания неудобный.

2. Предложите 2 метода сбора данных для проверки гипотез:

Качественный: Короткие интервью с 5 пользователями.

Количественный: Аналитика кликов на кнопке Создать и времени, проведенного в разделе.

3. Предложите 1 конкретное улучшение функции на основе одной из гипотез.

Критерии оценки:

Реалистичность гипотез.

Адекватность предложенных методов проверки.

Практичность и конкретность улучшения.

Задание к самостоятельной 4. Прототипирование экрана "Личный кабинет".

Создать простой прототип главного экран.

1. Перечислите 5 ключевых элементов**, которые *обязательно* должны быть на экране.

2. Создайте низкодетализированный прототип (wireframe).

Используйте бумагу и ручку или простой инструмент (Balsamiq, Figma - простые фигуры).

Нарисуйте схему расположения элементов (без дизайна, цвета).

Подпишите основные блоки.

3. Объясните расположение 1 важного элемента.

Критерии оценки:

Полнота охвата ключевых функций.

Удобство и логичность расположения элементов на схеме.

Четкость обоснования выбора позиции элемента.

Задание к самостоятельной 5. Анализ структуры заданного программного модуля.

Задание 1: Анализ модуля аутентификации (Python/Псевдокод.

Дан модуль `auth.py`, отвечающий за вход пользователя в систему. Оценить связность (cohesion) и зацепление (coupling), выявить структурные проблемы.

```
python
# Фрагмент модуля auth.py
def login(username, password):
    user = db_query("SELECT * FROM users WHERE username = ?",
(username,)) # Прямой запрос к БД
    if not user:
        return "Пользователь не найден"
    if hash_password(password) != user['password_hash']: # Хеширование
внутри функции
        return "Неверный пароль"
    session['user_id'] = user['id'] # Прямая работа с сессией
    log_activity(f"User {username} logged in") # Логирование внутри
    return "Успешный вход"

def hash_password(password): # Вспомогательная функция внутри модуля
    return hashlib.sha256(password.encode()).hexdigest()
...

```

1. Разбейте модуль на компоненты.

Выделите основные ответственности (например: проверка данных, работа с БД, хеширование, управление сессией, логирование).

2. Оцените связность (Cohesion).

Является ли модуль высокосвязным? Все ли функции относятся к одной задаче?

Ответьте: Какие функции нарушают принцип единственной ответственности (SRP)? Почему?

3. Оцените зацепление (Coupling).

С какими внешними системами/модулями взаимодействует `login()` (БД, сессия, логгер)?

Является ли зацепление сильным (tight) или слабым (loose)? Обоснуйте.

4. Предложите рефакторинг.

Как можно выделить ответственности в отдельные модули/классы?

Предложите новую структуру (например: `UserRepository` для БД, `PasswordHasher`, `SessionManager`, `ActivityLogger`).

5. Напишите псевдокод улучшенной функции `login()`, использующей зависимости через параметры или внедрение.

Критерии оценки.

Точность выделения ответственностей.

Глубина анализа нарушения SRP и сильного зацепления.

Практичность предложений по рефакторингу.

Корректность псевдокода после изменений.

Задание 2: Анализ цикломатической сложности модуля обработки заказов (Java/Псевдокод).

Дан метод `processOrder(Order order)` в модуле `OrderService.java`.

Рассчитать цикломатическую сложность, выявить сложные для тестирования участки.

```
```java
// Фрагмент OrderService.java
public String processOrder(Order order) {
 if (order == null || order.getItems().isEmpty()) { // Условие 1
 throw new InvalidOrderException("Пустой заказ");
 }

 for (Item item : order.getItems()) { // Цикл 1
 if (item.getStockCount() <= 0) { // Условие 2 (внутри цикла)
 if (!inventory.restockItem(item.getId(), 10)) { // Условие 3
 (вложенное) throw new InventoryException("Не удалось пополнить " +
 item.getName());
 }
 }
 inventory.reserveItem(item.getId(), 1); // Действие
 }

 if (paymentGateway.charge(order.getTotal(), order.getPaymentToken())) { //
Условие 4
 order.setStatus(Status.PAID);
 notificationService.sendEmail(order.getUserEmail(), "Заказ оплачен"); //
Зависимость
 return "Заказ успешно обработан";
 } else {
 order.setStatus(Status.PAYMENT_FAILED);
 return "Ошибка оплаты";
 }
}
```
```

1. Рассчитайте цикломатическую сложность (СС).

Используйте формулу: $СС = E - N + 2P$, где `E` - рёбра графа потока управления, `N` - узлы, `P` - компоненты связности (обычно 1 для метода).

Или подсчитайте: * `1` (начало) + количество условий (`if`, `for`, `while`, `case`) + 1.

Укажите итоговое значение СС.

2. Оцените результат.

Какое значение СС считается высоким (обычно >10)? Превышает ли метод

этот порог?

Какие последствия у высокой сложности (тестируемость, читаемость, поддержка)?

3. Выделите проблемные участки.

Перечислите блоки кода (по условиям/циклам), которые сложнее всего протестировать изолированно. Почему?

4. Предложите упрощение.

Как можно разбить метод на более мелкие (например, `validateOrder()`, `reserveInventory()`, `processPayment()`)?

Предложите 1-2 конкретных изменения для снижения СС.

Критерии оценки.

Правильность расчета цикломатической сложности.

Понимание влияния высокой СС на качество кода.

Точность выявления сложных для тестирования фрагментов.

Эффективность предложений по декомпозиции.

Задание 3: Анализ зависимостей и поиск "Божественного объекта" в модуле отчетов (C#/Псевдокод).

Дан класс `ReportGenerator.cs`, формирующий различные отчеты.

Построить граф зависимостей, выявить нарушение принципов модульности.

```C#

// Фрагмент ReportGenerator.cs

public class ReportGenerator {

private DatabaseConnector \_dbConnector; // Зависимость 1: Работа с БД

private ExcelExporter \_excelExporter; // Зависимость 2: Экспорт в Excel

private PdfRenderer \_pdfRenderer; // Зависимость 3: Рендер PDF

private EmailService \_emailService; // Зависимость 4: Отправка email

public void GenerateSalesReport(DateTime from, DateTime to, string format) {

var data = \_dbConnector.GetSalesData(from, to); // Исп. Зав.1

if (format == "Excel") {

var file = \_excelExporter.Export(data, "Sales"); // Исп. Зав.2

\_emailService.Send("sales@company.com", "Отчет", file); // Исп.

Зав.4

} else if (format == "PDF") {

var file = \_pdfRenderer.Render(data); // Исп. Зав.3

\_emailService.Send("sales@company.com", "Отчет", file); // Исп.

Зав.4

}

}

public void GenerateInventoryReport(string format) {

var data = \_dbConnector.GetInventoryData(); // Исп. Зав.1

// ... Аналогичная логика выбора формата и отправки ...

}

// ... Еще 5 методов для разных отчетов ...

}

...

1. Постройте граф зависимостей.

Изобразите схему (можно текстом/списком), показывающую, какие внешние сервисы (`DatabaseConnector`, `ExcelExporter` и т.д.) использует класс `ReportGenerator`.

Укажите, сколько разных ответственностей реализует сам `ReportGenerator` (получение данных, преобразование в формат, отправка).

2. Выявите "Божественный объект" (God Object).

Обладает ли класс `ReportGenerator` слишком многими обязанностями?

Какие признаки "Божественного объекта" здесь присутствуют (размер, количество зависимостей, несвязанные функции)?

3. Проанализируйте устойчивость.

Как изменение в `EmailService` (например, новый параметр в `Send()`) повлияет на `ReportGenerator`?

Нарушает ли класс принцип открытости/закрытости (ОСР)? Почему?

4. Предложите новую структуру.

. Как разделить ответственности? (Например: `ReportDataProvider`, `ReportFormatter` (с подклассами `ExcelFormatter`, `PdfFormatter`), `ReportSender`).

. Как `ReportGenerator` должен взаимодействовать с этими новыми компонентами (использовать интерфейсы)?

Критерии оценки.

Точность построения графа зависимостей.

Глубина анализа признаков "Божественного объекта".

Понимание нарушения ОСР и влияния изменений.

Практичность предложенной декомпозиции и снижения связанности.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

## Основная литература

1. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 146 с. — (Профессиональное образование). — ISBN 978-5-534-18094-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563828>.

## Дополнительная литература

2. Казанский, А. А. Программирование на C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 181 с. — (Профессиональное образование). — ISBN 978-5-534-21380-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/569863>.

3. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : Учебное пособие / Г. Н. Федорова. — Москва : КУРС : ИНФРА-М, 2022. — 336 с. (Среднее профессиональное образование). — ISBN 9785906818416. — Текст : непосредственный.

4. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563151>.

Составитель  
Витвицкий Максим Николаевич

Методические указания по выполнению самостоятельной работы  
для студентов очной формы обучения  
по направлению специальности  
09.02.07 «Информационные системы и программирование»

Публикуется в авторской редакции