

Документ подписан простой электронной подписью
Информация о документе:
ФИО: Соловьев Дмитрий Александрович
Должность: ректор ФГБОУ ВО Вавиловский университет
Дата подписания: 13.05.2024 14:11:26
Уникальный программный ключ:
528682d78e671e566ab07f01fe1ba2172f735a12

Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный университет генетики, биотехнологии и инженерии имени Н.И. Вавилова»
Финансово-технологический колледж

УТВЕРЖДАЮ
Директор Финансово-технологического
колледжа
ФГБОУ ВО Вавиловский университет
Г. Н. Митрофанова



Методические указания
по учебной практике УП.03.01
профессионального модуля

ПМ.03 Ревьюирование программных продуктов

программы подготовки специалистов среднего звена

по специальности

09.02.07 Информационные системы и программирование

Саратов, 2024 год

Методические указания по учебной практике УП 03.01. по специальности среднего профессионального образования 09.02.07 «Информационные системы и программирование» и в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование», утвержденного приказом Министерства образования и науки от 9 декабря 2016 года № 1547 (зарегистрирован Министерством юстиции Российской Федерации 26 декабря 2016г., регистрационный №44936)

Организация-разработчик: Финансово-технологический колледж

«Саратовский государственный университет генетики, биотехнологии и инженерии имени Н.И. Вавилова»

Разработчик: Зыбина Ольга Викторовна, преподаватель

Рассмотрено на заседании предметной комиссии агротехнологических дисциплин и модулей, протокол № 6 от «11» января 2024 года.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Обучающиеся проходят учебную практику по профессиональному модулю ПМ.03 Ревьюирование программных продуктов соответствии с учебным планом по специальности, графиком учебного процесса и рабочей программой профессионального модуля.

2. Учебная практика по профессиональному модулю направлена на формирование у обучающихся умений, приобретение первоначального практического опыта и реализуется в рамках профессионального модуля по основному виду профессиональной деятельности ВД.03 Ревьюирование программных продуктов для последующего освоения ими общих и профессиональных компетенций по избранной специальности.

3. Учебная практика проводится в учебных, учебно-производственных мастерских, лабораториях, учебно-опытных хозяйствах, учебных полигонах, учебных базах практики и иных структурных подразделениях образовательной организации либо в организациях в специально оборудованных помещениях на основе договоров между организацией, осуществляющей деятельность по образовательной программе соответствующего профиля и в колледже, в объеме 36 часов.

4. Учебная практика проводится преподавателями дисциплин профессионального цикла.

5. Результаты практики определяются программами практики, разрабатываемыми колледжем.

6. Организацию и руководство учебной практикой осуществляют руководители практики от колледжа.

7. Обучающийся в период прохождения практики в организациях, обязан:

-выполнять задания, предусмотренные программой учебной практики;

-соблюдать действующие в организации правила внутреннего распорядка;

-соблюдать требования охраны труда, безопасности жизнедеятельности и пожарной безопасности.

8.Обучающийся имеет право:

- получать консультации по организации практики;

-обращаться к заместителю директора по учебно-производственной работе, представителям администрации, руководителям практики от колледжа по вопросам организации практики;

-самостоятельно выбирать место прохождения практики (по согласованию с руководителем практики от колледжа), если программа практики может быть реализована в данной организации.

9.К прохождению учебной практики допускаются обучающиеся, освоившие теоретический курс обучения в рамках профессионального модуля.

10.В период прохождения практики обучающийся ведёт дневник практики и составляет отчёт по результатам практики.

11.Руководитель практики от колледжа:

-проводит с обучающимися организационные собрания, знакомит с целями и задачами практики, особенностями её организации, планируемыми результатами;

- инструктирует обучающихся о соблюдении требований охраны труда, безопасности жизнедеятельности и пожарной безопасности «под роспись»;

-знакомит обучающихся с процедурами оценки общих и профессиональных компетенций, освоенных в ходе прохождения практики, формами отчётности и оценочными материалами прохождения практики;

-контролирует реализацию программы практики обучающимся и посещаемость практики;

-оказывает методическую помощь обучающемуся при выполнении индивидуальных заданий и сборе материалов, необходимых для отчета о практике;

-проверяет соблюдение обучающимися требований охраны труда, безопасности жизнедеятельности и пожарной безопасности в период прохождения практики;

-проверяет дневник прохождения практики, отчет о практике, составляет рецензию на выполнение отчета с раскрытием положительных моментов отчета, указанием ошибок, недочетов.

12.Руководителями учебной практики от колледжа по результатам практики формируется аттестационный лист, содержащий сведения об уровне освоения обучающимся профессиональных компетенций в период прохождения практики.

13. Обучающимся по результатам практики предоставляется дневник практики составляется отчет учебной практике.

14. Практика завершается дифференцированным зачетом при условии положительного аттестационного листа по практике руководителя практики от колледжа об уровне освоения профессиональных компетенций, полноты и своевременности представления отчета о практике в соответствии с заданием на практику.

15. Отчет учебной практике обучающиеся представляют в виде оформленных инструкционно-технологических карт по видам работ, предусмотренных профессиональным модулем.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПРИ ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ

В результате прохождения учебной практики обучающийся должен продемонстрировать следующие результаты обучения.

Общие компетенции:

- ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
- ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.
- ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Профессиональные компетенции:

- ПК 3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией
- ПК 3.2. Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям.
- ПК 3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.
- ПК 3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием

В результате прохождения практики обучающийся должен **приобрести первичный практический опыт:**

- измерения характеристик программного проекта;
- использования основных методологий процессов разработки программного обеспечения;
- оптимизации программного кода с использованием специализированных программных средств

.уметь:

- работать с проектной документацией, разработанной с использованием графических языков спецификаций;
- выполнять оптимизацию программного кода с использованием специализированных программных средств;
- использовать методы и технологии тестирования и ревьюирования кода и проектной документации;
- применять стандартные метрики по прогнозированию затрат, сроков и качества.

знать:

- задачи планирования и контроля развития проекта;
- принципы построения системы деятельности программного проекта;
- современные стандарты качества программного продукта и процессов его обеспечения.

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРАКТИКЕ ОБ УРОВНЕ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ

Профессиональный модуль ПМ.03 Ревьюирование программных продуктов
Обучающийся _____

_____ курса по специальности 09.02.07 Информационные системы и программирование успешно прошел учебную практику в объеме _____ часов с «__» _____ 20__ г. по «__» _____ 20__ г. на базе Финансово-технологического колледжа ФГБОУ ВО Вавиловского университета кабинет технического оснащения для специальности 09.02.07 Информационные системы и программирование.

В ходе учебной практики обучающимся освоены профессиональные компетенции:

Код ПК	Профессиональная компетенция	Уровни освоения	Результат освоения (от 1 до 4)
ПК 3.1.	Осуществлять ревьюирование программного кода в соответствии с технической документацией	<p>1 уровень. Демонстрирует элементарные умения и знания, допускает ошибки при ревьюировании программного кода в соответствии с технической документацией, работает под контролем руководителя практики добавила</p> <p>2 уровень. Демонстрирует умения и знания в ревьюировании программного кода в соответствии с технической документацией, работает под контролем руководителя практики</p> <p>3 уровень. Демонстрирует умения, знания и практические навыки, самостоятельно осуществляет ревьюирование программного кода в соответствии с технической документацией</p> <p>4 уровень. Демонстрирует высокий уровень умений, знаний и практических навыков, свободно осуществляет ревьюирование программного кода в соответствии с технической документацией.</p>	
ПК 3.2.	Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям.	<p>1 уровень. Демонстрирует элементарные умения и знания, допускает ошибки при выполнении процесса измерения характеристик компонент программного продукта для определения соответствия заданным критериям, работает под контролем руководителя практики</p> <p>2 уровень. Демонстрирует умения и знания в выполнении процесса измерения характеристик компонент программного продукта для определения соответствия заданным критериям, работает под контролем руководителя практики</p> <p>3 уровень. Демонстрирует умения, знания и практические навыки, самостоятельно выполняет процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям.</p> <p>4 уровень. Демонстрирует высокий уровень умений, знаний и практических навыков при выполнении процесса измерения характеристик компонент программного продукта для определения соответствия заданным критериям</p>	
ПК 3.3	Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.	<p>1 уровень. Демонстрирует элементарные умения и знания, допускает ошибки при исследовании созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма, работает под контролем руководителя практики</p> <p>2 уровень. Демонстрирует умения и знания в исследовании созданного программного кода с использованием специализированных программных</p>	

Критерии оценивания уровней освоения общих компетенций

Шифр компетенции	Планируемые результаты обучения	Уровни освоения
ОК 01.	Выбирать способы применительно к различным контекстам	<p>1 уровень Не умеет выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.</p> <p>2 уровень Слабо ориентируется в выборе способов решения задач профессиональной деятельности, применительно к различным контекстам</p> <p>3 уровень Умеет выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>4 уровень Свободно и четко выбирает способы решения задач профессиональной деятельности, применительно к различным контекстам</p>
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.	<p>1 уровень Не умеет осуществлять поиск информации, необходимой для эффективного выполнения профессиональных задач и не умеет использовать информационно-коммуникационные технологии в профессиональной деятельности.</p> <p>2 уровень Осуществляет поиск и использование информации, необходимой для эффективного выполнения профессиональных задач по инструкциям руководителя практики уровень, применяет информационно-коммуникационные технологии в профессиональной деятельности только под руководством руководителя практики.</p> <p>3 уровень Способен осуществлять поиск информации, необходимой для эффективного выполнения профессиональных задач и использование информационно-коммуникационных технологий в профессиональной деятельности.</p> <p>4 уровень Проявляет творчество, инициативу при поиске необходимой информации, эффективно использует ее в работе, свободно владеет информационно-коммуникационными технологиями.</p>
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках.	<p>1 уровень Не умеет пользоваться профессиональной документацией на государственном и иностранном языке</p> <p>2 уровень Умеет пользоваться профессиональной документацией на государственном и иностранном языке, работает под руководством руководителя</p> <p>3 уровень Способен пользоваться профессиональной документацией на государственном и иностранном языке</p> <p>4 уровень Свободно пользуются профессиональной документацией на государственном и иностранном языке и использует их в работе</p>

Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный университет генетики, биотехнологии и инженерии
имени Н.И. Вавилова»

Финансово-технологический колледж

**ДНЕВНИК
УЧЕБНОЙ ПРАКТИКИ**

ПРОФЕССИОНАЛЬНЫЙ МОДУЛЬ

ПМ.03 Ревьюирование программных продуктов

по образовательной программе СПО – программе подготовки специалистов средне-
го звена

Специальность 09.02.07 Информационные системы и программирование

(Ф.И.О обучающегося)

курс _____ группа _____

на базе Финансово-технологического колледжа ФГБОУ ВО Вавиловского
университета кабинет технического оснащения для специальности 09.02.07
Информационные системы и программирование.

Время прохождения практики с « » 20 г. по « » 20 г

Руководитель практики от колледжа _____ преподаватель
(Ф.И.О., должность)

Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный университет генетики, биотехнологии и инженерии
имени Н.И. Вавилова»

Финансово-технологический колледж

**ОТЧЕТ
УЧЕБНОЙ ПРАКТИКИ**

ПРОФЕССИОНАЛЬНЫЙ МОДУЛЬ

ПМ.03 Ревьюирование программных продуктов

по образовательной программе СПО – программе подготовки специалистов средне-
го звена

Специальность 09.02.07 Информационные системы и программирование

(Ф.И.О обучающегося)

курс _____ группа _____

на базе Финансово-технологического колледжа ФГБОУ ВО Вавиловского
университета кабинет с компьютерами и установленным программным
обеспечением

Время прохождения практики с « » 20 г. по « » 20 г

Руководитель практики от колледжа
(Ф.И.О., должность)

Оценка по практике: дифференцированный зачет (_____)

Саратов

ИНСТРУКЦИОННО-ТЕХНОЛОГИЧЕСКАЯ КАРТА № 1

Междисциплинарный курс/профессиональный модуль: ПМ.03 Ревьюирование программных продуктов/ УП. 03. Учебная практика по модулю.

Специальность: 09.02.07 Информационные системы и программирование

Группа: _____

Дата проведения: _____

Учебное время: 6 часов

Место проведения: кабинет технического оснащения для специальности 09.02.07 Информационные системы и программирование.

Преподаватель: _____

Наименование работы: Согласовать порядок выполнения заданий с руководителем практики от колледжа. Пройти инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности. Работа с проектной документацией, разработанной с использованием графических языков спецификаций

Цели работы и образовательные результаты в соответствии с ФГОС СПО:

образовательные:

- частичное освоение ПК3.1., ПК 3.2

воспитательные:

- частичное формирование ОК01, ОК 09

развивающие:

- частичное формирование ОК02, ОК 09

Термины и понятия: в соответствии с рабочей программой.

Техника безопасности: проведение первичного инструктажа в соответствии с инструкцией по технике безопасности, правилами СанПиН и др.

Информационно-методическое обеспечение: см. в соответствии с видами работ

- **раздаточный материал:** опорный конспект, учебная литература

Материально-техническое обеспечение:

- **наглядные пособия:** мультимедийные презентации

- **технические средства (оборудование):** компьютер с установленным программным обеспечением

Порядок выполнения работы

№ п/п	Задания, последовательность выполнения операций	Норма времени, мин.	Инструктивные указания
1.	Инструктаж на рабочем месте	10 мин	Методические указания
2.	Написать ответы на вопросы.	20 мин	Заполнить приложение 1 Написать ответы на вопросы
3.	Настройка системы контроля версий Git	20 мин	Заполнить приложение 1 Перечислить стадии установки
4.	Создание Git-репозитория..	80 мин	Заполнить приложение 1 Перечислить стадии создания репозитория и сделать скриншоты экрана
5.	Рассмотреть ТЗ и по нему создать с использованием графических языков спецификации UML диаграммы	120 мин	Заполнить приложение 1 Рассмотреть ТЗ и по нему создать с использованием графических языков спецификации UML диаграммы (диаграмму вариантов использования, диаграмму классов, диаграмму действий)
6.	Отчет о выполненной работе	20 мин	Отчет о выполненной работе

Задание 1. Написать ответы на вопросы

1. СКВ- это _____

2. Какие проблемы решает Git?

3. Для разработки какой операционной системы используется Git?

4. Репозиторий- это _____

5. В чем разница между GoogleDrive/DropBox/YandexDisk и репозиторием Git?

Задание 2. Перечислить стадии установки системы контроля версий Git:

- Скачать с сайта _____,
 Установить _____,
 Указать настройки _____,

Задание 3. Создание Git- репозитория.

1. Запустите GitBash из меню Пуск,
2. Наберите команду ls.,
3. Создайте папку (непосредственно из-под Windows), назовите TMP,
4. В Bash перейти в эту папку. Для этого используем команду cd (changedirectory): \$
cd /d/tmp/
5. С помощью команды pwd посмотрите, какая директория текущая в данный момент.
Наберите команду и проверьте, где вы находитесь.
6. Задаем настройки Git.
7. Инициализируйте репозиторий в текущей пустой папке
6. Выполните команду ls. с ключом -a:
7. Проверьте в каком состоянии в данный момент находится наш репозиторий

Задание 4. Рассмотреть ТЗ и по нему создать с использованием графических языков спецификации UML диаграммы (диаграмму вариантов использования, диаграмму классов, диаграмму действий)

Диаграмма вариантов использования

Диаграмма классов

Диаграмма действий

ИНСТРУКЦИОННО-ТЕХНОЛОГИЧЕСКАЯ КАРТА № 2

Междисциплинарный курс/профессиональный модуль: ПМ.03 Ревьюирование программных продуктов/ УП. 03. Учебная практика по модулю.

Специальность: 09.02.07 Информационные системы и программирование

Группа: _____

Дата проведения: _____

Учебное время: 6 часов

Место проведения: кабинет технического оснащения для специальности 09.02.07 Информационные системы и программирование.

Преподаватель: _____

Наименование работы: Работа с проектной документацией, разработанной с использованием графических языков спецификаций

Прогнозирование затрат, сроков и качества с помощью стандартных метрик.

Цели работы и образовательные результаты в соответствии с ФГОС СПО:

образовательные:

- частичное освоение ПКЗ.1., ПК 3.2

воспитательные:

- частичное формирование ОК01

развивающие:

- частичное формирование ОК02, ОК 09

Термины и понятия: в соответствии с рабочей программой.

Техника безопасности: проведение первичного инструктажа в соответствии с инструкцией по технике безопасности, правилами СанПиН и др.

Информационно-методическое обеспечение: см. в соответствии с видами работ

- **раздаточный материал:** опорный конспект, учебная литература

Материально-техническое обеспечение:

- **наглядные пособия:** мультимедийные презентации

- **технические средства (оборудование):** компьютер с установленным программным обеспечением

Порядок выполнения работы

№ п/п	Задания, последовательность выполнения операций	Норма времени, мин.	Инструктивные указания
1.	Знакомство с заданием	5 мин	Методические указания
2.	Написать ответы на вопросы.	10 мин	Заполнить приложение 2 Написать ответы на вопросы
3.	Разработать техническое задание по темам	120 мин	Заполнить приложение 2 Разработать техническое задание по заданной предметной области
4.	Создать проект. Ввести наименование задач.	30 мин	Заполнить приложение 2 Создать проект. Ввести наименование задач.
5.	Создать лист ресурсов	100 мин	Заполнить приложение 2 Создать лист ресурсов
6.	Отчет о выполненной работе	5 мин	Отчет о выполненной работе

Задание 1. Ответьте на вопросы.

1. Что обозначает аббревиатура UML?

2. Кому и зачем нужен UML?

3. Что послужило причиной возникновения UML?

4. Для чего используют UML на практике?

5. Из чего состоит UML?

6. Из каких элементов состоит модель?

7. Какова общая структура модели?

Задание 2. Разработай техническое задание

Техническое задание Приложение 2.1

Задание 3. Создать проект. Ввести наименование задач.

Таблица 1. Перечень задач

Назначение задачи	Длительность	Ресурсы
	__ дней	
	__ день	
	- __ день	
	__ день	
	__ день	
	__ день	
	__ дней	

	___ день	
	___ день	
	___ день	
	___ день	
	___ дней	
	___ дней	

2. Создать лист ресурсов.

Таблица 2. Лист ресурсов

Назначение ресурса	Тип	Единицы измерения материалов	Стандартная ставка	Ставка сверхурочных	Затраты на исполъз.
Постановщик	Трудовой		_____ р./ч	_____ р./ч	0,00р.
Программист 1	Трудовой		_____ р./ч	_____ р./ч	_____р.
Программист 2	Трудовой		_____ р./ч	0,00 р./ч	_____р.
Бумага	Материальный	шт.	_____ р.		_____р.

Выделить фазы, вехи, критические задачи. Сформировать собственный настраиваемый отчет

ИНСТРУКЦИОННО-ТЕХНОЛОГИЧЕСКАЯ КАРТА 3

Междисциплинарный курс/профессиональный модуль: ПМ.03 Ревьюирование программных продуктов/ УП. 03. Учебная практика по модулю.

Специальность: 09.02.07 Информационные системы и программирование

Группа: _____

Дата проведения: _____

Учебное время: 6 часов

Место проведения: кабинет технического оснащения для специальности 09.02.07 Информационные системы и программирование.

Преподаватель: _____

Наименование работы: Оптимизация программного кода с использованием специализированных программных средств

Цели работы и образовательные результаты в соответствии с ФГОС СПО:
образовательные:

- частичное освоение ПК3.1., ПК 3.2

воспитательные:

- частичное формирование ОК01, ОК 09

развивающие:

- частичное формирование ОК02, ОК 09

Термины и понятия: в соответствии с рабочей программой.

Техника безопасности: проведение первичного инструктажа в соответствии с инструкцией по технике безопасности, правилами СанПиН и др.

Информационно-методическое обеспечение: см. в соответствии с видами работ

- **раздаточный материал:** опорный конспект, учебная литература

Материально-техническое обеспечение:

- **наглядные пособия:** мультимедийные презентации

- **технические средства (оборудование):** компьютер с установленным программным обеспечением

Порядок выполнения работы

№ п/п	Задания, последовательность выполнения операций	Норма времени, мин.	Инструктивные указания
1.	Знакомство с заданием	10 мин	Методические указания
2.	Профилирование и анализ программного кода	60 мин	Заполнить приложение 3 Установите инструменты профилирования и провести его. Выполнить анализ памяти
3.	Оптимизация кода	130 мин	Заполнить приложение 3 Оптимизируйте код, основываясь на результатах профилирования
4.	Верификация и бенчмаркинг	60 мин	Заполнить приложение 3 Создайте тесты для проверки корректности. Проведите сравнительный бенчмаркинг. Проанализируйте компромиссы.
5.	Отчет о выполненной работе	10 мин	Отчет о выполненной работе

Задание 1: Профилирование и анализ

1. Установите инструменты профилирования:

```
bash
pip install memory_profiler line_profiler snakeviz
```

2. Проведите базовое профилирование:

Используйте cProfile для анализа времени выполнения:

```
bash
python -m cProfile -o profile_stats.prof main.py
```

Визуализируйте результаты с помощью snakeviz:

```
bash
snakeviz profile_stats.prof
```

Определите 3 самые "тяжёлые" функции по времени выполнения.

3. Проведите построчное профилирование:

Добавьте декоратор @profile к методам calculate_statistics и find_correlations

Запустите:

```
bash
kernprof -l -v main.py
```

Определите самые медленные строки в этих методах.

4. Анализ использования памяти:

- Используйте memory_profiler для отслеживания потребления памяти
- Найдите места с максимальным потреблением памяти

Задание 2: Оптимизация кода

Оптимизируйте код, основываясь на результатах профилирования:

1. Устраните избыточные операции:

- Уберите ненужные копии данных
- Оптимизируйте вычисление статистик, избегая повторных расчётов

2. Оптимизируйте алгоритмы:

- Замените медленные реализации на более эффективные
- Используйте встроенные функции Python и библиотеки (например, statistics, numpy если можно)

3. Улучшите вычисление корреляций:

- Оптимизируйте или замените медленный алгоритм поиска корреляций
- Устраните избыточные циклы и условия

4. Минимизируйте аллокации памяти:

- Уменьшите количество создаваемых промежуточных списков
- Используйте генераторы там, где это уместно

Задание 3: Верификация и бенчмаркинг

1. Создайте тесты для проверки корректности:

- Напишите unit-тесты, проверяющие, что оптимизированный код возвращает те же результаты
- Используйте pytest или unittest

2. Проведите сравнительный бенчмаркинг:

- Сравните производительность до и после оптимизаций
- Используйте timeit или создайте простой скрипт для сравнения
- Измерьте ускорение (speedup) для разных размеров данных

3. Проанализируйте компромиссы:

- Оцените, насколько оптимизации повлияли на читаемость кода
- Предложите дальнейшие пути оптимизации, если потребуется

Требования к результату

1. **Оптимизированный код** в файле `optimized_data_analyzer.py`
2. **Отчёт по профилированию** в формате Markdown (`profiling_report.md`), содержащий:
 - Результаты профилирования (топ-5 самых медленных функций)
 - Графики или таблицы сравнения производительности
 - Описание найденных проблем и применённых оптимизаций
3. **Тесты** в файле `test_optimized.py`
4. **Скрипт для бенчмаркинга** `benchmark.py`
5. **Краткий итоговый отчёт** (`summary.md`) с:
 - Общим ускорением (в процентах или разях)
 - Самыми эффективными оптимизациями
 - Рекомендациями по дальнейшему улучшению

Критерии оценки

- **Качество анализа** (использование инструментов профилирования) — 3 балла
- **Эффективность оптимизаций** (измеримое ускорение) — 3 балла
- **Сохранение функциональности** (все тесты проходят) — 2 балла
- **Качество кода** (читаемость, комментарии, PEP8) — 1 балл
- **Полнота отчёта** — 1 балл

Советы

1. Начните с профилирования — не оптимизируйте "вслепую"
2. Фокусируйтесь на самых медленных частях кода (правило 80/20)
3. Рассмотрите возможность использования `numpy` для числовых операций
4. Проверяйте каждую оптимизацию на корректность
5. Документируйте изменения и их влияние на производительность

Пример ожидаемого улучшения

Исходный код (`data_size=10000`):

- Вычисление статистик: ~0.8 сек
- Анализ корреляций: ~1.5 сек
- **Общее время:** ~2.3 сек

После оптимизации (цель):

- Вычисление статистик: < 0.1 сек
- Анализ корреляций: < 0.3 сек
- **Общее время:** < 0.4 сек (ускорение в 5+ раз)

ИНСТРУКЦИОННО-ТЕХНОЛОГИЧЕСКАЯ КАРТА 4

Междисциплинарный курс/профессиональный модуль: ПМ.03 Ревьюирование программных продуктов/ УП. 03. Учебная практика по модулю.

Специальность: 09.02.07 Информационные системы и программирование

Группа: _____

Дата проведения: _____

Учебное время: 12 часов

Место проведения: кабинет технического оснащения для специальности 09.02.07 Информационные системы и программирование.

Преподаватель: _____

Наименование работы: Использование методов и технологий тестирования и ревьюирования кода и проектной документации

Цели работы и образовательные результаты в соответствии с ФГОС СПО:

образовательные:

- частичное освоение ПК3.1., ПК 3.2

воспитательные:

- частичное формирование ОК01, ОК 09

развивающие:

- частичное формирование ОК02, ОК 09

Термины и понятия: в соответствии с рабочей программой.

Техника безопасности: проведение первичного инструктажа в соответствии с инструкцией по технике безопасности, правилами СанПиН и др.

Информационно-методическое обеспечение: см. в соответствии с видами работ

- **раздаточный материал:** опорный конспект, учебная литература

Материально-техническое обеспечение:

- **наглядные пособия:** мультимедийные презентации

- **технические средства (оборудование):** компьютер с установленным программным обеспечением

Порядок выполнения работы

№ п/п	Задания, последовательность выполнения операций	Норма времени, мин.	Инструктивные указания
1.	Знакомство с заданием	10 мин	Методические указания
2.	Тестирование кода	130 мин	Заполнить приложение 4 Провести различные виды тестирования
3.	Ревью кода	130 мин	Заполнить приложение 4 Проведите code review предоставленного кода. Создайте отчет в определенном формате
4.	Ревью документации	120 мин	Заполнить приложение 4 Проанализируйте и улучшите документацию
5.	CI/CD и автоматизация	120 мин	Заполнить приложение 4 Создайте конфигурацию для автоматизированного тестирования
6.	Отчет о выполненной работе	20 мин	Отчет о выполненной работе

Всестороннее тестирование, ревью кода и документации для проекта DataAnalyzer

Ваш коллега написал оптимизированную версию `DataAnalyzer` из предыдущего задания. Теперь вам необходимо провести полный цикл обеспечения качества: протестировать код, провести ревью и оценить документацию.

Оптимизированный код для анализа (предоставленный коллегой)

```
import numpy as np
from typing import Dict, List, Tuple
import json
from dataclasses import dataclass
from statistics import mean as stats_mean
import time
from functools import lru_cache

@dataclass
class AnalysisResult:
    """Результат анализа данных."""
    mean: float
    median: float
    std_dev: float
    variance: float
    min_val: float
    max_val: float
    range_val: float
    cv_percentage: float
    outlier_count: int
    outlier_percentage: float

    def to_dict(self) -> Dict:
        """Конвертация в словарь."""
        return {
            'mean': self.mean,
            'median': self.median,
            'std_dev': self.std_dev,
            'variance': self.variance,
            'min': self.min_val,
            'max': self.max_val,
            'range': self.range_val,
            'cv_percentage': self.cv_percentage,
            'outlier_count': self.outlier_count,
            'outlier_percentage': self.outlier_percentage
        }
```

```

class OptimizedDataAnalyzer:
    """Оптимизированный анализатор данных с использованием NumPy."""

    def __init__(self, data_size: int = 10000, use_numpy: bool = True):
        """
        Инициализация анализатора.

        Args:
            data_size: Размер генерируемых данных
            use_numpy: Использовать NumPy для вычислений
        """
        self.data_size = data_size
        self.use_numpy = use_numpy
        self.data = self._generate_data()

    def _generate_data(self) -> np.ndarray:
        """Генерация случайных данных с использованием NumPy."""
        print(f"[INFO] Генерация {self.data_size} случайных чисел...")
        return np.random.uniform(0, 1000, self.data_size)

    @lru_cache(maxsize=1)
    def calculate_statistics(self) -> AnalysisResult:
        """
        Вычисление статистических показателей.
        Использует кэширование для повторных вызовов.
        """
        print("[INFO] Вычисление статистик...")

        if self.use_numpy:
            return self._calculate_with_numpy()
        else:
            return self._calculate_without_numpy()

    def _calculate_with_numpy(self) -> AnalysisResult:
        """Вычисление статистик с использованием NumPy."""
        data = self.data

        # Основные статистики
        mean_val = np.mean(data)
        median_val = np.median(data)
        std_val = np.std(data, ddof=1)
        var_val = np.var(data, ddof=1)
        min_val = np.min(data)
        max_val = np.max(data)
        range_val = max_val - min_val

```

```
cv_percentage = (std_val / mean_val * 100) if mean_val != 0 else 0
```

```
# Выбросы (правило 3 сигм)
```

```
outliers_mask = np.abs(data - mean_val) > 2 * std_val
```

```
outlier_count = np.sum(outliers_mask)
```

```
outlier_percentage = (outlier_count / len(data)) * 100
```

```
return AnalysisResult(
```

```
    mean=float(mean_val),
```

```
    median=float(median_val),
```

```
    std_dev=float(std_val),
```

```
    variance=float(var_val),
```

```
    min_val=float(min_val),
```

```
    max_val=float(max_val),
```

```
    range_val=float(range_val),
```

```
    cv_percentage=float(cv_percentage),
```

```
    outlier_count=int(outlier_count),
```

```
    outlier_percentage=float(outlier_percentage)
```

```
)
```

```
def _calculate_without_numpy(self) -> AnalysisResult:
```

```
    """Вычисление статистик без NumPy (резервный метод)."""
```

```
    data = self.data.tolist()
```

```
    mean_val = stats_mean(data)
```

```
    sorted_data = sorted(data)
```

```
    n = len(sorted_data)
```

```
# Медиана
```

```
if n % 2 == 0:
```

```
    median_val = (sorted_data[n//2 - 1] + sorted_data[n//2]) / 2
```

```
else:
```

```
    median_val = sorted_data[n//2]
```

```
# Дисперсия и стандартное отклонение
```

```
var_val = sum((x - mean_val) ** 2 for x in data) / (n - 1)
```

```
std_val = var_val ** 0.5
```

```
min_val = min(data)
```

```
max_val = max(data)
```

```
range_val = max_val - min_val
```

```
cv_percentage = (std_val / mean_val * 100) if mean_val != 0 else 0
```

```
# Выбросы
```

```
outlier_count = sum(1 for x in data if abs(x - mean_val) > 2 * std_val)
```

```
outlier_percentage = (outlier_count / n) * 100
```

```

return AnalysisResult(
    mean=float(mean_val),
    median=float(median_val),
    std_dev=float(std_val),
    variance=float(var_val),
    min_val=float(min_val),
    max_val=float(max_val),
    range_val=float(range_val),
    cv_percentage=float(cv_percentage),
    outlier_count=int(outlier_count),
    outlier_percentage=float(outlier_percentage)
)

```

```

def find_correlations(self, sample_size: int = 100) -> List[Dict]:

```

```

    """

```

Поиск корреляций между случайными парами чисел.

Args:

sample_size: Количество проверяемых пар

Returns:

Список словарей с результатами корреляций

```

    """

```

```

print(f"[INFO] Поиск корреляций для {sample_size} пар...")

```

```

if len(self.data) < 20:

```

```

    return []

```

```

correlations = []

```

```

for _ in range(min(sample_size, 1000)):

```

```

    # Выбор случайных подвыборок

```

```

    idx1 = np.random.randint(0, len(self.data) - 20)

```

```

    idx2 = np.random.randint(0, len(self.data) - 20)

```

```

    sample1 = self.data[idx1:idx1 + 20]

```

```

    sample2 = self.data[idx2:idx2 + 20]

```

```

    # Вычисление корреляции

```

```

    corr = self._calculate_correlation(sample1, sample2)

```

```

    correlations.append({

```

```

        'index_pair': (int(idx1), int(idx2)),

```

```

        'correlation': float(corr),

```

```

        'sample_sizes': (20, 20)

```

```

    })

    # Сортировка по абсолютному значению корреляции
    correlations.sort(key=lambda x: abs(x['correlation']), reverse=True)
    return correlations[:10]

def _calculate_correlation(self, x: np.ndarray, y: np.ndarray) -> float:
    """Вычисление корреляции Пирсона."""
    if self.use_numpy:
        return np.corrcoef(x, y)[0, 1] if len(x) > 1 and len(y) > 1 else 0.0

    # Ручной расчет без NumPy
    n = min(len(x), len(y))
    if n < 2:
        return 0.0

    x = x[:n]
    y = y[:n]

    mean_x = np.mean(x) if self.use_numpy else sum(x) / n
    mean_y = np.mean(y) if self.use_numpy else sum(y) / n

    numerator = sum((x[i] - mean_x) * (y[i] - mean_y) for i in range(n))
    denom_x = sum((xi - mean_x) ** 2 for xi in x)
    denom_y = sum((yi - mean_y) ** 2 for yi in y)

    if denom_x == 0 or denom_y == 0:
        return 0.0

    return numerator / ((denom_x * denom_y) ** 0.5)

def save_results(self, filename: str):
    """Сохранение результатов анализа в файл."""
    stats = self.calculate_statistics()
    correlations = self.find_correlations()

    results = {
        'metadata': {
            'data_size': self.data_size,
            'use_numpy': self.use_numpy,
            'timestamp': time.strftime('%Y-%m-%d %H:%M:%S')
        },
        'statistics': stats.to_dict(),
        'top_correlations': correlations[:5]
    }

```

```

with open(filename, 'w') as f:
    json.dump(results, f, indent=2)

print(f"[INFO] Результаты сохранены в {filename}")

# main_optimized.py
"""
Основной скрипт для демонстрации работы оптимизированного анализатора.
"""

from optimized_data_analyzer import OptimizedDataAnalyzer
import time

def main():
    print("=== Оптимизированный анализ данных ===")

    # Тестирование разных конфигураций
    configs = [
        {'size': 1000, 'numpy': True},
        {'size': 10000, 'numpy': True},
        {'size': 10000, 'numpy': False},
        {'size': 50000, 'numpy': True}
    ]

    for config in configs:
        print(f"\n--- Конфигурация: size={config['size']}, numpy={config['numpy']} ---")

        # Создание анализатора
        start_time = time.time()
        analyzer = OptimizedDataAnalyzer(
            data_size=config['size'],
            use_numpy=config['numpy']
        )
        init_time = time.time() - start_time

        # Вычисление статистик
        start_time = time.time()
        stats = analyzer.calculate_statistics()
        stats_time = time.time() - start_time

        # Анализ корреляций
        start_time = time.time()
        correlations = analyzer.find_correlations(sample_size=50)
        corr_time = time.time() - start_time

    # Вывод результатов

```

```

print(f"Время инициализации: {init_time:.4f} сек")
print(f"Время вычисления статистик: {stats_time:.4f} сек")
print(f"Время анализа корреляций: {corr_time:.4f} сек")
print(f"Общее время: {init_time + stats_time + corr_time:.4f} сек")
print(f"Среднее: {stats.mean:.2f}, Медиана: {stats.median:.2f}")
print(f"Найдено корреляций: {len(correlations)}")

# Сохранение результатов
analyzer.save_results(f"analysis_results_{config['size']}_{config['numpy']}.json")

print("\n=== Анализ завершён ===")

if __name__ == "__main__":
    main()

```

Документация проекта DataAnalyzer

Описание

Проект предназначен для анализа больших массивов числовых данных. Обеспечивает вычисление статистических показателей и поиск корреляций.

Установка

```

```bash
pip install numpy

```

### Использование

```

python

from data_analyzer import DataAnalyzer

analyzer = DataAnalyzer(data_size=10000)
stats = analyzer.calculate_statistics()

```

### DataAnalyzer

Класс для анализа данных.

#### Методы:

- `calculate_statistics()`: Вычисляет статистики
- `find_correlations()`: Находит корреляции

## ЗАДАНИЕ 1: Тестирование кода

### 1.1. Модульное тестирование (Unit Testing)

Создайте comprehensive test suite:

```
```python
# test_optimized.py - шаблон для начала
import pytest
import numpy as np
from optimized_data_analyzer import OptimizedDataAnalyzer, AnalysisResult

class TestOptimizedDataAnalyzer:
    """Тесты для OptimizedDataAnalyzer."""

    def test_initialization(self):
        """Тест инициализации анализатора."""
        # TODO: Проверка корректной инициализации
        pass

    def test_data_generation(self):
        """Тест генерации данных."""
        # TODO: Проверка размера и типа данных
        pass

    @pytest.mark.parametrize("use_numpy", [True, False])
    def test_statistics_calculation(self, use_numpy):
        """Тест вычисления статистик."""
        # TODO: Проверка корректности вычислений
        # Сравнение с numpy/stats для верификации
        pass

    def test_correlation_range(self):
        """Тест диапазона корреляции."""
        # TODO: Корреляция должна быть в диапазоне [-1, 1]
        pass

    def test_edge_cases(self):
        """Тест граничных случаев."""
        # TODO: Пустые данные, один элемент, все одинаковые значения
        pass

    def test_performance(self):
        """Тест производительности."""
        # TODO: Измерение времени выполнения для разных размеров
        pass
```
```

## Задачи:

1. Напишите минимум 10 различных тестов, покрывающих:
  - Нормальные случаи использования
  - Граничные случаи
  - Ошибочные входные данные
  - Производительность

### 1.2. Интеграционное тестирование

Создайте интеграционные тесты:

```
python

test_integration.py
"""
Интеграционные тесты для полного workflow.
"""

def test_complete_workflow():
 """Тест полного рабочего процесса."""
 # TODO: Создание анализатора -> вычисление статистик ->
 # поиск корреляций -> сохранение результатов -> загрузка
 pass

def test_numpy_vs_no_numpy():
 """Сравнение результатов с numpy и без."""
 # TODO: Результаты должны быть близки (с допустимой погрешностью)
 pass
```

### 1.3. Нагрузочное тестирование

Создайте тесты производительности:

```
python

test_performance.py
"""
Тесты производительности.
"""

import time
import pytest

@pytest.mark.performance
class TestPerformance:

 @pytest.mark.parametrize("data_size", [1000, 10000, 50000])
```

```

def test_statistics_performance(self, data_size, benchmark):
 """Бенчмарк вычисления статистик."""
 # TODO: Использовать pytest-benchmark или timeit
 pass

def test_memory_usage(self):
 """Тест использования памяти."""
 # TODO: Использовать memory_profiler
 pass

```

## Задание 2 Ревью кода

Проведите code review предоставленного кода. Создайте отчет в формате:

Code Review Report: [optimized\\_data\\_analyzer.py](#)

### Общая оценка

[Ваша оценка качества кода]

#### Положительные аспекты

1. [Что сделано хорошо]
2. [Хорошие практики]
3. [Удачные решения]

#### Проблемы и предложения

##### Категория 1: Производительность

1. **\*\*Проблема\*\***: [Описание]
  - **\*\*Файл/строка\*\***: optimized\_data\_analyzer.py:XX
  - **\*\*Предложение\*\***: [Как исправить]
  - **\*\*Критичность\*\***: High/Medium/Low

##### Категория 2: Читаемость и поддержка

1. **\*\*Проблема\*\***: [Описание]
  - **\*\*Файл/строка\*\***: ...
  - **\*\*Предложение\*\***: ...
  - **\*\*Критичность\*\***: ...

##### Категория 3: Безопасность

1. **\*\*Проблема\*\***: [Описание]
  - **\*\*Файл/строка\*\***: ...
  - **\*\*Предложение\*\***: ...
  - **\*\*Критичность\*\***: ...

##### Категория 4: Архитектура и дизайн

## 1. **\*\*Проблема\*\***: [Описание]

- **\*\*Файл/строка\*\***: ...
- **\*\*Предложение\*\***: ...
- **\*\*Критичность\*\***: ...

### Рекомендации по исправлению

1. [Приоритетные исправления]
2. [Долгосрочные улучшения]

### Критерии для review:

1. **Производительность**: Оптимальность алгоритмов, использование кэширования
2. **Читаемость**: Имена переменных, комментарии, структура кода
3. **Поддержка**: Модульность, тестируемость, расширяемость
4. **Безопасность**: Обработка ошибок, валидация входных данных
5. **Следование стандартам**: PEP 8, типизация, документация

### Задание 3: Ревью документации

Проанализируйте и улучшите документацию:

### Documentation Review Report

#### Текущее состояние

[Оценка существующей документации]

#### Недостатки

##### 1. **Отсутствующая информация**:

- [Что отсутствует]
- **Важность**: High/Medium/Low

##### 2. **Неточная информация**:

- [Что неверно]
- **Исправление**: [Как исправить]

##### 3. **Плохая структура**:

- [Проблемы с организацией]
- **Предложение**: [Как улучшить]

### Предлагаемая структура документации

docs/

```
|— README.md
|— INSTALLATION.md
|— USAGE.md
|— API_REFERENCE.md
|— EXAMPLES.md
```

└─ CONTRIBUTING.md  
└─ CHANGELOG.md

Конкретные улучшения

Примеры использования

[Добавьте разнообразные примеры]

Рекомендации по инструментам

1. Использовать Sphinx + autodoc для генерации документации
2. Добавить docstring ко всем публичным методам
3. Создать интерактивные примеры в Jupyter Notebook

Задание 4: CI/CD и автоматизация

Создайте конфигурацию для автоматизированного тестирования:

```
name: CI

on: [push, pull_request]

jobs:
 test:
 runs-on: ubuntu-latest

 strategy:
 matrix:
 python-version: ['3.8', '3.9', '3.10']

 steps:
 - uses: actions/checkout@v2

 - name: Set up Python
 uses: actions/setup-python@v2
 with:
 python-version: ${ { matrix.python-version } }

 - name: Install dependencies
 run: |
 python -m pip install --upgrade pip
 pip install -r requirements.txt
 pip install pytest pytest-cov

 - name: Run tests
 run: |
```

```
pytest --cov=optimized_data_analyzer --cov-report=xml
```

- **name:** Upload coverage

**uses:** codecov/codecov-action@v2

- **name:** Code quality

**run:** |

```
pip install black flake8 mypy
```

```
black --check optimized_data_analyzer.py
```

```
flake8 optimized_data_analyzer.py
```

```
mypy optimized_data_analyzer.py
```

### Задачи:

1. Создайте requirements.txt и setup.py
2. Настройте автоматический запуск тестов при push/pull request
3. Добавьте проверки code quality
4. Настройте отчет о покрытии кода тестами

Требования к результату

1. **Полный набор тестов** (минимум 15 тестов):
  - test\_optimized.py - модульные тесты
  - test\_integration.py - интеграционные тесты
  - test\_performance.py - тесты производительности
  - test\_edge\_cases.py - тесты граничных случаев
2. **Отчеты о ревью:**
  - code\_review\_report.md - подробный review кода
  - documentation\_review.md - анализ и предложения по документации
3. **Улучшенная документация:**
  - docs/ директория с полной документацией
4. **Конфигурация CI/CD:**
5. **Итоговый отчет**

Критерии оценки

  - **Полнота тестового покрытия** (разные типы тестов, edge cases)
  - **Качество code review** (глубина анализа, конкретные предложения)
  - **Улучшение документации** (полнота, ясность, структура)
  - **Автоматизация** (CI/CD, инструменты quality assurance)
  - **Профессиональный подход** (системность, воспроизводимость) - 1 балл

Дата сдачи работы: \_\_\_\_\_

Оценка уровня сформированности компетенций: ПК3.1., ПК 3.2.– частично освоена, ОК01–02., ОК 09– полностью сформированы

Оценка: \_\_\_\_\_

Подпись преподавателя: \_\_\_\_\_

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

### Виды выполняемых работ

**Согласовать порядок выполнения заданий с руководителем практики от колледжа. Пройти инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности Работа с проектной документацией, разработанной с использованием графических языков спецификаций.**

1. Михеева, Е. В. Информационные технологии в профессиональной деятельности : учеб. пособие / Е. В. Михеева. - 14-е изд., стер. - М. : Академия, 2016. - 384 с.
2. Гохберг, Г. С. Информационные технологии : учебник / Г. С. Гохберг, А. В. Зафиевский, А. А. Короткин. - 9-е изд., перераб. и доп. - М. : Академия, 2014. - 240 с.

### Виды выполняемых работ

**Работа с проектной документацией, разработанной с использованием графических языков спецификаций. Прогнозирование затрат, сроков и качества с помощью стандартных метрик. Оптимизация программного кода с использованием специализированных программных средств**

1. Рудаков А. В. Технология разработки программных продуктов: учебник для студ. учреждений сред. проф. образования / А. В. Рудаков. - 12-е изд., стер. - Москва: Издательский центр "Академия", 2018. - 208 с. ISBN 978-5-4468-7402-6
2. Поколодина Е. В. Ревьюирование программных модулей : учебник для студ. учреждений сред. проф. образования / Е. В. Поколодина, Н. А. Долгова, Д. В. Ананьев. - М. : Издательский центр «Академия», 2020. - 208 с. ISBN 978-5-4468- 8609-8
3. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru>

### Виды выполняемых работ

**Использование методов и технологий тестирования и ревьюирования кода и проектной документации**

1. Рудаков А. В. Технология разработки программных продуктов: учебник для студ. учреждений сред. проф. образования / А. В. Рудаков. - 12-е изд., стер. - Москва: Издательский центр "Академия", 2018. - 208 с. ISBN 978-5-4468-7402-6
2. Поколодина Е. В. Ревьюирование программных модулей : учебник для студ. учреждений сред. проф. образования / Е. В. Поколодина, Н. А. Долгова, Д. В. Ананьев. - М. : Издательский центр «Академия», 2020. - 208 с. ISBN 978-5-4468- 8609-8
3. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru>

### Виды выполняемых работ

**Использование методов и технологий тестирования и ревьюирования кода и проектной документации**

1. Рудаков А. В. Технология разработки программных продуктов: учебник для студ. учреждений сред. проф. образования / А. В. Рудаков. - 12-е изд., стер. - Москва: Издательский центр "Академия", 2018. - 208 с. ISBN 978-5-4468-7402-6
2. Поколодина Е. В. Ревьюирование программных модулей : учебник для студ. учреждений сред. проф. образования / Е. В. Поколодина, Н. А. Долгова, Д. В. Ананьев. - М. : Издательский центр «Академия», 2020. - 208 с. ISBN 978-5-4468- 8609-8

3. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru>

## РЕЦЕНЗИЯ НА ОТЧЁТ УЧЕБНОЙ ПРАКТИКИ

Обучающийся \_\_\_\_\_  
(фамилия, имя, отчество)

Курс \_\_\_\_\_

Группа \_\_\_\_\_

Специальность \_\_\_\_\_

Профессиональный модуль ПМ.03 Ревьюирование программных продуктов

1. Место прохождения учебной практики на базе \_\_\_\_\_

ФГБОУ ВО «Саратовский государственный университет генетики, биотехнологии и инженерии имени Н.И. Вавилова»

2. Аттестационный лист обучающегося по практике от руководителя практики от колледжа (представлен /не представлен) подтверждает \_\_\_\_\_ уровень освоения профессиональных компетенций.

3. Характеристика по освоению профессиональных и общих компетенций обучающегося по практике от руководителя практики от колледжа (представлена /не представлена) подтверждает \_\_\_\_\_ уровень освоения компетенций.

4. Дневник учебной практики (представлен /не представлен).  
Соответствие видов и объёмов выполненных работ и отведённому на каждый вид работ времени, предусмотренных заданиями на учебную практику по профессиональному модулю (соответствует / не соответствует).

5. Отчет учебной практике (представлен /не представлен).  
Доказательства выполнения работ, предусмотренных заданиями на учебную практику по профессиональному модулю (представлены / представлены не в полном объеме/ не представлены).  
Обучающийся верно оформил не менее \_\_\_\_\_% представленных заданий.  
Отчётная документация отражает \_\_\_\_\_ уровень теоретической и практической подготовки обучающегося, (не содержит/ содержит) недочётов по содержанию и/или оформлению.

Недостатки/ достоинства отчёта о прохождении практики

6. (Обеспечены/ не обеспечены) доказательства личного участия обучающегося в составлении и оформлении представленной документации.

Вывод: отчет учебной практики заслуживает оценки \_\_\_\_\_

Руководитель практики от колледжа: \_\_\_\_\_