

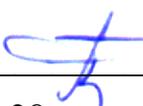
Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования



**Пермский национальный исследовательский
политехнический университет**

УТВЕРЖДАЮ

Проректор по образовательной
деятельности

 А.Б. Петроченков

« 28 » апреля 20 23 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина: Информатика в приложении к отрасли
(наименование)

Форма обучения: очная
(очная/очно-заочная/заочная)

Уровень высшего образования: бакалавриат
(бакалавриат/специалитет/магистратура)

Общая трудоёмкость: 108 (3)
(часы (ЗЕ))

Направление подготовки: 09.03.02 Информационные системы и технологии
(код и наименование направления)

Направленность: Информационные системы и технологии (общий профиль,
СУОС)
(наименование образовательной программы)

1. Общие положения

1.1. Цели и задачи дисциплины

Формирование комплекса знаний, умений и навыков в области применения программных средств автоматизации сборки, тестирования и анализа качества программного кода при разработке информационных систем.

1.2. Изучаемые объекты дисциплины

Назначение, применение программных комплексов для решения задач сборки, тестирования и анализа качества программного кода при разработке информационных систем.

1.3. Входные требования

Знание языков программирования.
Базовые навыки администрирования операционных систем.

2. Планируемые результаты обучения по дисциплине

Компетенция	Индекс индикатора	Планируемые результаты обучения по дисциплине (знать, уметь, владеть)	Индикатор достижения компетенции, с которым соотнесены планируемые результаты обучения	Средства оценки
ОПК-3	ИД-1ОПК-3	Знает назначение и способы применения инструментов контроля версий исходного кода; способы организации параллельной разработки кода при помощи данных инструментов, способы управления виртуальными окружениями Python.	Знает возможности и ограничения современных информационно-коммуникационных технологий; основные принципы и нормативные правовые акты информационной безопасности и защиты информации	Собеседование
ОПК-3	ИД-2ОПК-3	Умеет использовать инструменты контроля версий исходного кода; организовывать параллельную разработку кода при помощи данных инструментов. Умеет создавать автоматизированные тесты и настраивать непрерывную интеграцию изменений. Умеет управлять виртуальными окружениями Python.	Умеет выбирать информационно-коммуникационные технологии, необходимые для эффективного решения стандартных задач в области профессиональной деятельности с учетом требований информационной безопасности	Отчёт по практическому занятию

Компетенция	Индекс индикатора	Планируемые результаты обучения по дисциплине (знать, уметь, владеть)	Индикатор достижения компетенции, с которым соотнесены планируемые результаты обучения	Средства оценки
ОПК-3	ИД-3ОПК-3	Владеет методами автоматизированного тестирования ПО, в том числе веб-сервисов с использованием современных программных средств, методами организации удаленного доступа к сервисам на платформе Unix по защищенным каналам связи.	Владеет информационной и библиографической культурой, навыками обеспечения информационной безопасности и защиты информации при решении стандартных задач в области профессиональной деятельности	Зачет

3. Объем и виды учебной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах	
		Номер семестра	
		5	
1. Проведение учебных занятий (включая проведение текущего контроля успеваемости) в форме:	45	45	
1.1. Контактная аудиторная работа, из них:			
- лекции (Л)	16	16	
- лабораторные работы (ЛР)			
- практические занятия, семинары и (или) другие виды занятий семинарского типа (ПЗ)	27	27	
- контроль самостоятельной работы (КСР)	2	2	
- контрольная работа			
1.2. Самостоятельная работа студентов (СРС)	63	63	
2. Промежуточная аттестация			
Экзамен			
Дифференцированный зачет			
Зачет	9	9	
Курсовой проект (КП)			
Курсовая работа (КР)			
Общая трудоемкость дисциплины	108	108	

4. Содержание дисциплины

Наименование разделов дисциплины с кратким содержанием	Объем аудиторных занятий по видам в часах			Объем внеаудиторных занятий по видам в часах
	Л	ЛР	ПЗ	СРС
5-й семестр				

Наименование разделов дисциплины с кратким содержанием	Объем аудиторных занятий по видам в часах			Объем внеаудиторных занятий по видам в часах
	Л	ЛР	ПЗ	СРС
Системы онтроля версий програмного кода.	2	0	3	8
Системы контроля версий исходного кода, Git. Использование сервисов контроля версий исходного кода для организации параллельной разработки.				
Работа с проектами в Python.	2	0	4	9
Виртуальные окружения Python. Работа с виртуальными окружениями python при помощи интегрированных сред разработки.				
Тестирование програмного обеспечения.	2	0	4	9
Тестирование програмного обеспечения. Автоматизация unit-тестирования програмного обеспечения на python. Анализ покрытия тестами исходного кода.				
Контроль качества кода. Непрерывная интеграция кода.	2	0	4	9
Системы непрерывной интеграции изменений кода. Системы контроля качества исходного кода.				
Разработка веб-сервисов.	4	0	4	10
Разработка веб-сервисов с использованием современных средств разработки.				
Автоматизированное тестирование веб-сервисов.	2	0	4	9
Автоматизированное тестирование веб-сервисов с использованием современных средств разработки.				
Шифрование информации при организации удаленного доступа.	2	0	4	9
Основы организации удаленного доступа к Unix системам. Шифрование информации при организации удаленного доступа.				
ИТОГО по 5-му семестру	16	0	27	63
ИТОГО по дисциплине	16	0	27	63

Тематика примерных практических занятий

№ п.п.	Наименование темы практического (семинарского) занятия
1	Базовые операции с репозиторием исходного кода git при помощи GitKraken. Организация параллельной разработки с использованием системы контроля версий исходного кода git.
2	Базовые операции с виртуальными окружениями python при помощи PyCharm. Создание виртуальных окружений, установка модулей.
3	Написание сценариев для тестирования програмного обеспечения. Тестирование програмного обеспечения при помощи pytest.
4	Построение процесса непрерывной интеграции при помощи Github Actions. Анализ качества програмного кода при помощи инструментов Codacy, Codecov.
5	Создание веб-сервиса на языке программирования Python.

№ п.п.	Наименование темы практического (семинарского) занятия
6	Автоматизация тестирования веб-сервиса при помощи современных средств.
7	Организация удаленного доступа к виртуальной машине с ОС Ubuntu Server.

5. Организационно-педагогические условия

5.1. Образовательные технологии, используемые для формирования компетенций

<p>Проведение лекционных занятий по дисциплине основывается на активном методе обучения, при котором учащиеся не пассивные слушатели, а активные участники занятия, отвечающие на вопросы преподавателя. Вопросы преподавателя нацелены на активизацию процессов усвоения материала, а также на развитие логического мышления. Преподаватель заранее намечает список вопросов, стимулирующих ассоциативное мышление и установление связей с ранее освоенным материалом.</p> <p>Практические занятия проводятся на основе реализации метода обучения действием: определяются проблемные области, формируются группы. При проведении практических занятий преследуются следующие цели: применение знаний отдельных дисциплин и креативных методов для решения проблем и принятия решений; отработка у обучающихся навыков командной работы, межличностных коммуникаций и развитие лидерских качеств; закрепление основ теоретических знаний.</p> <p>Проведение лабораторных занятий основывается на интерактивном методе обучения, при котором обучающиеся взаимодействуют не только с преподавателем, но и друг с другом. При этом доминирует активность учащихся в процессе обучения. Место преподавателя в интерактивных занятиях сводится к направлению деятельности обучающихся на достижение целей занятия.</p> <p>При проведении учебных занятий используются интерактивные лекции, групповые дискуссии, ролевые игры, тренинги и анализ ситуаций и имитационных моделей.</p>
--

5.2. Методические указания для обучающихся по изучению дисциплины

<p>При изучении дисциплины обучающимся целесообразно выполнять следующие рекомендации:</p> <ol style="list-style-type: none"> 1. Изучение учебной дисциплины должно вестись систематически. 2. После изучения какого-либо раздела по учебнику или конспектным материалам рекомендуется по памяти воспроизвести основные термины, определения, понятия раздела. 3. Особое внимание следует уделить выполнению отчетов по практическим занятиям, лабораторным работам и индивидуальным комплексным заданиям на самостоятельную работу. 4. Вся тематика вопросов, изучаемых самостоятельно, задается на лекциях преподавателем. Им же даются источники (в первую очередь вновь изданные в периодической научной литературе) для более детального понимания вопросов, озвученных на лекции.

6. Перечень учебно-методического и информационного обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Печатная учебно-методическая литература

№ п/п	Библиографическое описание (автор, заглавие, вид издания, место, издательство, год издания, количество страниц)	Количество экземпляров в библиотеке
1. Основная литература		

1	Бочкарёв С. В., Шмидт И. А. Корпоративные информационные системы : учебное пособие. Пермь : Изд-во ПГТУ, 2010. 363 с.	48
2	Меняев М. Ф. Информатика и основы программирования : учебное пособие. Москва : Омега-Л, 2005. 464 с.	13
3	Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем : учебник. Санкт-Петербург : Питер, 2002. 463 с.	12
4	Теория информации и кодирование / Самсонов Б. Б., Плохов Е. М., Филоненков А. И., Кречет Т. В. Ростов-на-Дону : Феникс, 2002. 287 с.	10
5	Фридланд А. Я. Информатика: процессы, системы, ресурсы : монография. М. : БИНОМ. Лаб. знаний, 2003. 232 с.	11
2. Дополнительная литература		
2.1. Учебные и научные издания		
1	Воеводин В. В. Вычислительная математика и структура алгоритмов : 10 лекций о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно, чтобы успешно преодолевать эти трудности : учебник для вузов / В. В. Воеводин. - Москва: Изд-во МГУ, 2010.	34
2	Гергель В. П. Современные языки и технологии параллельного программирования : учебник для вузов / В. П. Гергель. - Москва: Изд-во МГУ им. М. В. Ломоносова, 2012.	35
3	Инструменты параллельного программирования в системах с общей памятью : учебник для вузов / К. В. Корняков [и др.]. - Москва: Изд-во МГУ, 2010.	34
4	Линев А. В. Технологии параллельного программирования для процессоров новых архитектур : учебник для вузов / А. В. Линев, Д. К. Боголепов, С. И. Бастраков. - Москва: Изд-во МГУ, 2010.	34
5	Павловская Т. А. C/C++. Программирование на языке высокого уровня : для магистров и бакалавров учебник для вузов. Санкт-Петербург [и др.] : Питер, 2021. 460 с. 37,410 усл. печ. л.	50
2.2. Периодические издания		
	Не используется	
2.3. Нормативно-технические издания		
	Не используется	
3. Методические указания для студентов по освоению дисциплины		
	Не используется	
4. Учебно-методическое обеспечение самостоятельной работы студента		
	Не используется	

6.2. Электронная учебно-методическая литература

Вид литературы	Наименование разработки	Ссылка на информационный ресурс	Доступность (сеть Интернет / локальная сеть; авторизованный / свободный доступ)
Дополнительная литература	Журнал «Математическое моделирование»	https://www.elibrary.ru/title_about.asp?id=7877	сеть Интернет; авторизованный доступ

6.3. Лицензионное и свободно распространяемое программное обеспечение, используемое при осуществлении образовательного процесса по дисциплине

Вид ПО	Наименование ПО
Операционные системы	MS Windows 7 (подп. Azure Dev Tools for Teaching до 27.03.2022)
Офисные приложения.	Microsoft Office Professional 2007. лиц. 42661567
Прикладное программное обеспечение общего назначения	Dr.Web Enterprise Security Suite, 3000 лиц, ПНИПУ ОЦНИТ 2017

6.4. Современные профессиональные базы данных и информационные справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Наименование	Ссылка на информационный ресурс
База данных научной электронной библиотеки (eLIBRARY.RU)	https://elibrary.ru/
База данных уязвимостей CVE Mitre	https://cve.mitre.org/
Научная библиотека Пермского национального исследовательского политехнического университета	http://lib.pstu.ru/
Электронно-библиотечная система Лань	https://e.lanbook.com/
Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru/
Информационные ресурсы Сети КонсультантПлюс	http://www.consultant.ru/

7. Материально-техническое обеспечение образовательного процесса по дисциплине

Вид занятий	Наименование необходимого основного оборудования и технических средств обучения	Количество единиц
Лекция	Ноутбук	1
Лекция	Проектор	1
Практическое занятие	Персональный компьютер	20

8. Фонд оценочных средств дисциплины

Описан в отдельном документе

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Пермский национальный исследовательский политехнический
университет»**

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

(фонд оценочных средств)

для проведения промежуточной аттестации обучающихся по дисциплине
«Информатика в приложении к отрасли»
Приложение к рабочей программе дисциплины

Направление подготовки:	09.03.02 Информационные системы и технологии
Направленность (профиль) образовательной программы:	Цифровые технологии и интеллектуальные системы управления
Квалификация выпускника:	«Бакалавр»
Выпускающая кафедра:	Вычислительная математика, механика и биомеханика
Форма обучения:	Очная
Форма промежуточной аттестации:	Зачет

Оценочные материалы (фонд оценочных средств) для проведения промежуточной аттестации обучающихся по дисциплине является частью (приложением) к рабочей программе дисциплины. Для проведения промежуточной аттестации обучающихся по дисциплине устанавливает формы и процедуры текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине.

1. Перечень контролируемых результатов обучения по дисциплине, объекты оценивания и виды контроля

Согласно РПД освоение учебного материала дисциплины запланировано в течение одного семестра (1-го семестра учебного плана) и разбито на 7 учебных модулей. В каждом модуле предусмотрены аудиторские лекционные и лабораторные занятия, а также самостоятельная работа студентов. В рамках освоения учебного материала дисциплины формируются компоненты компетенций *знать, уметь, владеть*, указанные в РПД, которые выступают в качестве контролируемых результатов обучения по дисциплине (табл. 1.1).

Контроль уровня усвоенных знаний, освоенных умений и приобретенных владений осуществляется в рамках текущего, рубежного и промежуточного контроля при изучении теоретического материала, сдаче отчетов по лабораторным работам и зачета. Виды контроля сведены в таблицу 1.1.

Таблица 1.1. Перечень контролируемых результатов обучения по дисциплине

Контролируемые результаты обучения по дисциплине (ЗУВы)	Вид контроля					
	Текущий		Промежуточный / рубежный		Итоговый	
	С	ТО	ОЛР	Т/КР/КИЗ		Зачет
Усвоенные знания						
З.1 Знает назначение и способы применения инструментов контроля версий исходного кода; способы организации параллельной разработки кода при помощи данных инструментов, способы управления виртуальными окружениями Python.	С1					ТВ
Освоенные умения						
У.1 Умеет использовать инструменты контроля версий исходного кода; организовывать параллельную разработку кода при помощи данных инструментов. Умеет создавать автоматизированные тесты и настраивать непрерывную интеграцию изменений. Умеет управлять виртуальными окружениями Python.				ЛР 1-7		ПЗ
Приобретенные владения						
В.1 Владеет методами автоматизированного тестирования ПО, в том числе веб-сервисов с использованием современных программных средств, методами организации удаленного доступа к сервисам на платформе Unix по защищенным каналам связи.				ЛР 1-7		ПЗ

С – собеседование по теме; ТО – коллоквиум (теоретический опрос); ИЗ – индивидуальное задание; ЛР – выполнение лабораторных работ (оценка умений и навыков); ОЛР – отчет по лабораторной работе; РК – рубежный контроль; ТВ – теоретический вопрос; ПЗ – практическое задание; КИЗ – комплексное индивидуальное задание на самостоятельную работу; КЗ – комплексное задание экзамена.

Итоговой оценкой достижения результатов обучения по дисциплине является промежуточная аттестация в виде зачета, проводимая по результатам текущего, промежуточного и рубежного контроля.

1. Виды контроля, типовые контрольные задания и шкалы оценивания

результатов обучения

Текущий контроль успеваемости имеет целью обеспечение максимальной эффективности учебного процесса, управление процессом формирования заданных компетенций обучаемых, повышение мотивации к учебе и предусматривает оценивание хода освоения дисциплины. В соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования – программам бакалавриата, специалитета и магистратуры в ПНИПУ предусмотрены следующие виды и периодичность текущего контроля успеваемости обучающихся:

- входной контроль, проверка исходного уровня подготовленности обучаемого и его соответствия предъявляемым требованиям для изучения данной дисциплины;
- текущий контроль усвоения материала (уровня освоения компонента «знать» заданных компетенций) на каждом групповом занятии и контроль посещаемости лекционных занятий;
- промежуточный и рубежный контроль освоения обучаемыми отдельных компонентов «знать», «уметь» заданных компетенций путем опросов, контрольных работ (индивидуальных домашних заданий), защиты отчетов по лабораторным работам, рефератов, эссе и т.д.

Рубежный контроль по дисциплине проводится на следующей неделе после прохождения модуля дисциплины, а промежуточный – во время каждого контрольного мероприятия внутри модулей дисциплины;

- межсессионная аттестация, единовременное подведение итогов текущей успеваемости не менее одного раза в семестр по всем дисциплинам для каждого направления подготовки (специальности), курса, группы;
- контроль остаточных знаний.

2.1. Текущий контроль усвоения материала

Текущий контроль усвоения материала в форме собеседования или выборочного теоретического опроса студентов проводится по каждой теме. Результаты по 4-х балльной шкале оценивания заносятся в книжку преподавателя и учитываются в виде интегральной оценки при проведении промежуточной аттестации.

2.2. Промежуточный и рубежный контроль

Промежуточный и рубежный контроль для комплексного оценивания усвоенных знаний, усвоенных умений и приобретенных владений (табл. 1.1) проводится в форме защиты лабораторных работ (после изучения каждого модуля учебной дисциплины).

2.2.1. Защита лабораторных работ

Всего запланировано 7 лабораторных работ. Типовые темы лабораторных работ приведены в РПД. Защита лабораторной работы проводится индивидуально каждым студентом или группой студентов.

Критерии оценки лабораторных работ

Оценка «пять» ставится, если студент выполнил общую и индивидуальную часть задания успешно, показав в целом систематическое применение полученных знаний, умений и владений, студент справился с

систематизацией, анализом информации и оформлением отчетной документации.

Оценка «четыре» ставится, если студент выполнил общую и индивидуальную часть задания успешно, показав в целом систематическое, но сопровождающееся отдельными ошибками применение полученных знаний, умений и владений, студент справился с систематизацией, анализом информации и оформлением отчетной документации.

Оценка «три» ставится, если студент выполнил общую и индивидуальную часть задания с существенными неточностями. При выполнении задания показал удовлетворительное владение навыками применения полученных знаний и умений, студент неполностью справился с систематизацией, анализом информации и оформлением отчетной документации.

Оценка «два» ставится, если студент допустил много ошибок или не выполнил задание.

2.3. Промежуточная аттестация (итоговый контроль)

2.3.1. Процедура промежуточной аттестации без дополнительного аттестационного испытания

Промежуточная аттестация проводится в форме зачета. Зачет по дисциплине основывается на результатах текущего, промежуточного и рубежного контроля по данной дисциплине.

Критерии выведения итоговой оценки за компоненты компетенций при проведении промежуточной аттестации в виде зачета приведены в общей части ФОС образовательной программы.

2.3.2.1. Типовые вопросы и задания для зачета по дисциплине

Типовые вопросы для контроля усвоенных знаний:

1. Контроль версий программного кода. Определение, назначение, инструменты.
2. Непрерывная интеграция кода. Определение, основные составляющие процесса.
3. Тестирование программного кода. Валидация и верификация. Определение, примеры.

Типовое практическое задание для контроля освоенных умений приведено в Приложении 1.

2.3.2.2. Шкалы оценивания результатов обучения на зачете

Оценка результатов обучения по дисциплине в форме уровня сформированности компонентов *знать, уметь, владеть* заявленных компетенций проводится по 4-х балльной шкале оценивания.

Типовые шкала и критерии оценки результатов обучения при сдаче зачета для компонентов *знать, уметь и владеть* приведены в общей части ФОС образовательной программы.

3. Критерии оценивания уровня сформированности компонентов и компетенций

3.1. Оценка уровня сформированности компонентов компетенций

При оценке уровня сформированности компетенций в рамках выборочного контроля при зачете считается, что *полученная оценка за компонент проверяемой в билете компетенции обобщается на соответствующий компонент всех компетенций, формируемых в рамках данной учебной дисциплины.*

Общая оценка уровня сформированности всех компетенций проводится путем агрегирования оценок, полученных студентом за каждый компонент формируемых компетенций, с учетом результатов текущего контроля в виде интегральной оценки по 5-ти балльной шкале. Все результаты контроля заносятся в оценочный лист и заполняются преподавателем по итогам промежуточной аттестации.

Форма оценочного листа и требования к его заполнению приведены в общей части ФОС образовательной программы.

При формировании итоговой оценки промежуточной аттестации в виде зачета используются типовые критерии, приведенные в общей части ФОС образовательной программы.

Типовое практическое задание для контроля освоенных умений

Задача 1

Реализовать небольшую библиотеку функций. В библиотеку включить следующие функции:

- определения n чисел Фибоначчи – функция принимает n , возвращает список из чисел
- сортировку пузырьком функция принимает список чисел, возвращает его отсортированную копию
- калькулятор функция принимает 3 аргумента: число 1, число 2 и знак действия: +, -, *, / выполняет действие и возвращает результат

Реализовать unit-тесты данных функций при помощи библиотеки `pytest`. Реализовать тестирование реакции программы на корректные и не корректные входные данные. Определить классы эквивалентности и граничные значения для входных данных. Код сопроводить комментариями.

Задача 2

Создание репозитория на сервисе Github через браузер.

1. Создайте репозиторий **lab01** на сервисе Github через браузер.
2. Клонировать репозиторий на компьютер при помощи `GitKraken`
3. Создайте проект в `PyCharm` в данном репозитории
4. Добавьте папку с виртуальным окружением **venv** в исключения системы контроля версий
5. Создайте файл **main.py** в директории проекта с программой, которая реализует алгоритм Евклида – алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел.
6. Создайте коммит, отправьте его на сервер Github
7. Откройте в браузере репозиторий, убедитесь, что созданная программа находится в репозитории.

Задача 3

При помощи Web-версии сервиса Postman построить запросы к сервису Github.

Должны быть реализованы следующие запросы:

- Вывод списка репозиториях какой-либо организации, чьи репозитории расположены на Github
- Вывод списка репозиториях текущего пользователя
- Создание нового репозитория для текущего пользователя
- Удаление репозитория

Авторизацию производить при помощи токена. Имя пользователя, название репозитория, а также название организации и токен хранить в виде `environment variable`. **Токен хранить в виде `secret variable`!**

Задача 4

Построить процесс непрерывной интеграции изменений в программе на Python, при помощи Github Actions. Результаты тестов и анализ покрытия кода должны быть сохранены в виде артефакта. Также, анализ покрытия, должен быть выгружаться на сервис Codacy. Должна присутствовать интеграция с сервисом анализа качества кода Codacy. При успешном выполнении тестов код должен быть сохранен в виде артефакта. Процесс непрерывной интеграции должен содержать следующие задачи:

1. Тестирование проекта
 - a. Выгрузку репозитория
 - b. Установку Python версии 3.8 x64
 - c. Установку необходимых модулей

- d. Проведение модульных тестов программы с анализом покрытия кода тестами и генерацией html отчета с результатами тестов
 - e. Генерацию html отчета по качеству покрытия кода тестами
 - f. Выгрузку результата проведения тестов и анализа покрытия кода тестами в виде артефакта Github Actions (архив с html файлами)
 - g. Генерацию xml отчета (для Codacy) по качеству покрытия кода тестами
 - h. Выгрузку xml отчета на сервис Codacy
2. Публикацию проекта в виде архива на сервисе Github в случае успешного прохождения тестов
- a. Выгрузку репозитория
 - b. Выгрузку исходного кода в виде артефакта Github Actions

Токен доступа к сервису Codacy должен быть сохранен в виде secret в параметрах репозитория! Код с захаркоженным токеном в публичном доступе не принимается В выгруженный в виде артефакта код не должно попасть ничего, кроме кода!

Критерии оценки индивидуального комплексного задания

Оценка «зачтено» ставится, если студент выполнил все задания, либо успешно выполнил более 50% объема работы. Продемонстрировал достаточный уровень владения навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала.

Оценка «незачет» ставится, если студент выполнил часть заданий контрольной работы с существенными неточностями, либо успешно выполнил менее 50% объема работы. Продемонстрировал недостаточный уровень владения умениями и навыками при решении профессиональных задач в рамках усвоенного учебного материала.