

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Комарова Светлана Юриевна

Должность: Проректор по образовательной деятельности

Дата подписания: 2020.08.14 10:07:17

Уникальный программный ключ:

43ba42f5deae4140b1f449a708a781028071237781a11207c4a4549f2095d7a

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Омский государственный аграрный университет имени П.А. Столыпина»**

Экономический факультет

ОПОП по направлению 19.04.01 Биотехнология

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по освоению учебной дисциплины

Б1.О.10 Программирование на языке Python для научных задач

Направленность (профиль)

«Биоинформатика и анализ данных в агробиотехнологиях»

Обеспечивающая преподавание дисциплины кафедры	математических и естественнонаучных дисциплин
Разработчики, старшие преподаватели	Л. В. Ламонина О. Б. Смирнова

СОДЕРЖАНИЕ

Введение	3
1. Место учебной дисциплины в подготовке	4
2. Структура учебной работы, содержание и трудоёмкость основных элементов дисциплины	11
2.1. Организационная структура, трудоёмкость и план изучения дисциплины	11
2.2. Содержание дисциплины по разделам	12
3. Общие организационные требования к учебной работе обучающегося, условия допуска к экзамену / зачету	13
3.1. Организация занятий и требования к учебной работе обучающегося	13
3.2. Условия допуска к экзамену по дисциплине	13
4. Лекционные занятия	14
5. Лабораторные занятия по курсу и подготовка обучающегося к ним	15
6. Общие методические рекомендации по изучению отдельных разделов дисциплины	16
7. Общие методические рекомендации по оформлению и выполнению отдельных видов ВАРС	20
7.1. Рекомендации по выполнению РАР	20
7.2. Рекомендации по самостоятельному изучению тем	21
8. Текущий контроль успеваемости	24
9. Промежуточная (семестровая) аттестация	27
9.1 Нормативная база проведения промежуточной аттестации по результатам изучения дисциплины	27
9.2. Основные характеристики промежуточной аттестации по итогам изучения дисциплины для экзамена	27
9.3. Заключительное тестирование по итогам изучения дисциплины	27
9.4 Процедура проведения экзамена	28
10. Информационное и методическое обеспечение учебного процесса по дисциплине	30
Приложение 1 Форма титульного листа РАР	31

ВВЕДЕНИЕ

1. Настоящее издание является основным организационно-методическим документом учебно-методического комплекса по дисциплине в составе основной профессиональной образовательной программы высшего образования (ОПОП ВО). Оно предназначено стать для них методической основой по освоению данной дисциплины.

2. Содержательной основой для разработки настоящих методических указаний послужила Рабочая программа дисциплины, утвержденная в установленном порядке.

3. Методические аспекты развиты в учебно-методической литературе и других разработках, входящих в состав УМК по данной дисциплине.

4. Доступ обучающихся к электронной версии Методических указаний по изучению дисциплины, обеспечен в электронной информационно-образовательной среде университета.

При этом в электронную версию могут быть внесены текущие изменения и дополнения, направленные на повышение качества настоящих методических указаний.

Уважаемые обучающиеся!

Приступая к изучению новой для Вас учебной дисциплины, начните с вдумчивого прочтения разработанных для Вас кафедрой специальных методических указаний. Это поможет Вам вовремя понять и правильно оценить ее роль в Вашем образовании.

Ознакомившись с организационными требованиями кафедры по этой дисциплине и соизмерив с ними свои силы, Вы сможете сделать осознанный выбор собственной тактики и стратегии учебной деятельности, уберечь самих себя от неразумных решений по отношению к ней в начале семестра, а не тогда, когда уже станет поздно. Используя эти указания, Вы без дополнительных осложнений подойдете к промежуточной аттестации по этой дисциплине. Успешность аттестации зависит, прежде всего, от Вас. Ее залог – ритмичная, целенаправленная, вдумчивая учебная работа, в целях обеспечения которой и разработаны эти методические указания.

1. Место учебной дисциплины в подготовке выпускника

Учебная дисциплина относится к дисциплинам ОПОП университета, состав которых определяется вузом и требованиями ФГОС.

Цель дисциплины – формирование системы знаний по программированию для решения научных задач, формирование навыков использования существующих программных библиотек и моделей, создания программных реализаций на основе языка программирования Python с учетом основных принципов разработки скриптов, овладение основными принципами программирования на высокоуровневом языке Python и прикладными аспектами его применения.

В ходе освоения дисциплины обучающийся должен:

- иметь целостное представление о современных тенденциях программирования, технологиях и инструментах его реализации, современных трендов в области программирования, о языке программирования Python и сферах его применения, структуре программных кодов;

- владеть: навыками программирования, использования существующих программных библиотек и моделей, создания программных реализаций на основе языка программирования Python;

- знать: основные принципы программирования на высокоуровневом языке Python и прикладные аспекты его применения

- уметь: реализовывать различные алгоритмы на высокоуровневом языке программирования Python к решению задач.

1.1. Перечень компетенций с указанием этапов их формирования в результате освоения учебной дисциплины

Компетенции, в формировании которых задействована дисциплина		Код и наименование индикатора достижений компетенции	Компоненты компетенций, формируемые в рамках данной дисциплины (как ожидаемый результат ее освоения)		
код	наименование		знать и понимать	уметь делать (действовать)	владеть навыками (иметь навыки)
Общепрофессиональные компетенции					
ОПК-3	Способен разрабатывать алгоритмы и участвовать в разработке программ в сфере своей профессиональной деятельности	ИД-2 _{ОПК-3} Пишет скрипты на Python для обработки и анализа биоинформатических данных; работает с командной строкой для запуска биоинформатических инструментов	Знает синтаксис, типы данных, структуры управления, основные принципы создания скриптов на Python, основные биоинформатические форматы файлов (FASTA / FASTQ, SAM / BAM / CRAM, VCF; GFF / GTF / BED), для обработки и анализа их данных; основные приемы работы с командной строкой для запуска биоинформатических инструментов	Умеет писать скрипты на Python для обработки и анализа биоинформатических данных; запускать биоинформатические инструменты из командной строки, читать и интерпретировать выходные данные и сообщения об ошибках в терминале	Владеет навыками программной реализации обработки и анализа биоинформатических данных посредством написания скриптов, находить "узкие места" в скриптах и ускорять их работу, владение инструментами, позволяющие решать задачи в сфере своей профессиональной деятельности, оптимизируя рабочие процессы, обработки большие объемы данных и корректной обработки возможных ошибок
ОПК -4	Способен выбирать и использовать современные инструментальные методы и технологии, осваивать новые методы и технику исследований для решения конкретных задач профессиональной деятельности	ИД-3 _{ОПК-4} Выбирает и изучает новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности	Знает, как выбирать и изучает новое программное обеспечение или ключевые библиотеки Python для анализа данных: pandas: для эффективной обработки табличных данных; NumPy: для векторизованных вычислений; Biopython: специализированные инструменты для работы с биологическими данным, основы статистического анализа и машинного обучения	Умеет выбирать и новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности, а также парсить и обрабатывать биоинформатические файлы, используя стандартные средства или библиотеку Biopython, извлекать нужную информацию из файлов	Владеет навыками выбора программного обеспечения или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности и его применения. Владеет навыками ускорять работу за счет использования векторизации (NumPy), эффективных структур данных, написания чистого, читаемого и поддерживаемого кода, добавлять документацию и комментарии

1.2. Описание показателей, критериев и шкал оценивания в рамках дисциплины

Индекс и название компетенции	Код индикатора достижений компетенции	Индикаторы компетенции	Показатель оценивания – знания, умения, навыки (владения)	Уровни сформированности компетенций				Формы и средства контроля формирования компетенций
				компетенция не сформирована	минимальный	средний	высокий	
				Оценки сформированности компетенций				
				2	3	4	5	
				Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»	
				Характеристика сформированности компетенции				
				Компетенция в полной мере не сформирована. Имеющихся знаний, умений и навыков недостаточно для решения практических (профессиональных) задач	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения стандартных практических (профессиональных) задач	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач	
Критерии оценивания								
ОПК -3 Способен разрабатывать алгоритмы и участвовать в разработке программ в сфере своей профессиональной деятельности	ИД-2 _{ОПК-3} Пишет скрипты на Python для обработки и анализа биоинформатических данных; работает с командной строкой для запуска биоинформатических инструментов	Полнота знаний	Знает синтаксис, типы данных, структуры управления, основные принципы создания скриптов на Python, основные биоинформатические форматы файлов (FASTA / FASTQ, SAM / BAM / CRAM, VCF; GFF / GTF / BED), для обработки и анализа их данных; основные приемы работы с командной строкой для запуска биоинформатических инструментов	Не знает синтаксис, типы данных, структуры управления, основные принципы создания скриптов на Python, основные биоинформатические форматы файлов (FASTA / FASTQ, SAM / BAM / CRAM, VCF; GFF / GTF / BED), для обработки и анализа их данных; основные приемы работы с командной строкой для запуска биоинформатических инструментов	Знает синтаксис, типы данных, структуры управления, основные принципы создания скриптов на Python, основные биоинформатические форматы файлов (FASTA / FASTQ, SAM / BAM / CRAM, VCF; GFF / GTF / BED), для обработки и анализа их данных; основные приемы работы с командной строкой для запуска биоинформатических инструментов частично	Знает основы синтаксиса, типы данных, структуры управления, основные принципы создания скриптов на Python, основные биоинформатические форматы файлов (FASTA / FASTQ, SAM / BAM / CRAM, VCF; GFF / GTF / BED), для обработки и анализа их данных; основные приемы работы с командной строкой для запуска биоинформатических инструментов, но допускает несущественные ошибки	Знает основы синтаксиса, типы данных, структуры управления, основные принципы создания скриптов на Python, основные биоинформатические форматы файлов (FASTA / FASTQ, SAM / BAM / CRAM, VCF; GFF / GTF / BED), для обработки и анализа их данных; основные приемы работы с командной строкой для запуска биоинформатических инструментов.	Тестовые задания, расчетно-аналитическая работа

Индекс и название компетенции	Код индикатора достижений компетенции	Индикаторы компетенции	Показатель оценивания – знания, умения, навыки (владения)	Уровни сформированности компетенций				Формы и средства контроля формирования компетенций
				компетенция не сформирована	минимальный	средний	высокий	
				Оценки сформированности компетенций				
				2	3	4	5	
				Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»	
				Характеристика сформированности компетенции				
				Компетенция в полной мере не сформирована. Имеющихся знаний, умений и навыков недостаточно для решения практических (профессиональных) задач	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения стандартных практических (профессиональных) задач	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач	
Критерии оценивания								
ОПК -3 Способен разрабатывать алгоритмы и участвовать в разработке программ в сфере своей профессиональной деятельности	ИД-2 _{ОПК3} Пишет скрипты на Python для обработки и анализа биоинформатических данных; работает с командной строкой для запуска биоинформатических инструментов	Наличие умений	Умеет писать скрипты на Python для обработки и анализа биоинформатических данных; запускать биоинформатические инструменты из командной строки, читать и интерпретировать выходные данные и сообщения об ошибках в терминале	Не умеет писать скрипты на Python для обработки и анализа биоинформатических данных; запускать биоинформатические инструменты из командной строки, читать и интерпретировать выходные данные и сообщения об ошибках в терминале	Умеет частично писать скрипты на Python для обработки и анализа биоинформатических данных; запускать биоинформатические инструменты из командной строки, читать и интерпретировать выходные данные и сообщения об ошибках в терминале	Умеет писать скрипты на Python для обработки и анализа биоинформатических данных; запускать биоинформатические инструменты из командной строки, читать и интерпретировать выходные данные и сообщения об ошибках в терминале, допуская незначительные ошибки	Умеет писать скрипты на Python для обработки и анализа биоинформатических данных; запускать биоинформатические инструменты из командной строки, читать и интерпретировать выходные данные и сообщения об ошибках в терминале	Тестовые задания, расчетно-аналитическая работа
		Наличие навыков (владение опытом)	Владеет навыками программной реализации обработки и анализа биоинформатических данных посредством написания скриптов, находить "узкие места" в скриптах и ускорять их работу, владение инструментами, позволяющие решать задачи в сфере своей профессиональной деятельности, оптимизируя рабочие процессы, обработки большие объемы данных и корректной обработки возможных ошибок	Не владеет навыками программной реализации обработки и анализа биоинформатических данных посредством написания скриптов, находить "узкие места" в скриптах и ускорять их работу, владение инструментами, позволяющие решать задачи в сфере своей профессиональной деятельности, оптимизируя рабочие процессы, обработки большие объемы данных и корректной обработки возможных ошибок	Владеет навыками программной реализации обработки и анализа биоинформатических данных посредством написания скриптов, находить "узкие места" в скриптах и ускорять их работу, владение инструментами,но допускает грубые ошибки	Владеет навыками программной реализации обработки и анализа биоинформатических данных посредством написания скриптов, находить "узкие места" в скриптах и ускорять их работу, владение инструментами, но допускает несущественные ошибки	Владеет навыками программной реализации обработки и анализа биоинформатических данных посредством написания скриптов, находить "узкие места" в скриптах и ускорять их работу, владение инструментами, позволяющие решать задачи в сфере своей профессиональной деятельности, оптимизируя рабочие процессы, обработки большие объемы данных и корректной обработки возможных ошибок	

Индекс и название компетенции	Код индикатора достижений компетенции	Индикаторы компетенции	Показатель оценивания – знания, умения, навыки (владения)	Уровни сформированности компетенций				Формы и средства контроля формирования компетенций
				компетенция не сформирована	минимальный	средний	высокий	
				Оценки сформированности компетенций				
				2	3	4	5	
				Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»	
				Характеристика сформированности компетенции				
				Компетенция в полной мере не сформирована. Имеющихся знаний, умений и навыков недостаточно для решения практических (профессиональных) задач	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения стандартных практических (профессиональных) задач	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач	
Критерии оценивания								
ОПК-4. Способен выбирать и использовать современные инструментальные методы и технологии, осваивать новые методы и технику исследований для решения конкретных задач профессиональной деятельности	ИД-3 _{ОПК-4} Выбирает и изучает новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности	Полнота знаний	Знает как выбирать и изучает новое программное обеспечение или ключевые библиотеки Python для анализа данных: pandas: для эффективной обработки табличных данных; NumPy: для векторизованных вычислений; Biopython: специализированные инструменты для работы с биологическими данным, основы статистического анализа и машинного обучения	Не знает как выбирать и изучает новое программное обеспечение или ключевые библиотеки Python для анализа данных: pandas: для эффективной обработки табличных данных; NumPy: для векторизованных вычислений; Biopython: специализированные инструменты для работы с биологическими данным, основы статистического анализа и машинного обучения	Знает как выбирать и изучает новое программное обеспечение или ключевые библиотеки Python для анализа данных: pandas: для эффективной обработки табличных данных; NumPy: для векторизованных вычислений	Знает как выбирать и изучает новое программное обеспечение или ключевые библиотеки Python для анализа данных: pandas: для эффективной обработки табличных данных; NumPy: для векторизованных вычислений; Biopython: специализированные инструменты для работы с биологическими данным, основы статистического анализа и машинного обучения	Знает как выбирать и изучает новое программное обеспечение или ключевые библиотеки Python для анализа данных: pandas: для эффективной обработки табличных данных; NumPy: для векторизованных вычислений; Biopython: специализированные инструменты для работы с биологическими данным, основы статистического анализа и машинного обучения	Тестовые задания, расчетно-аналитическая работа

Индекс и название компетенции	Код индикатора достижений компетенции	Индикаторы компетенции	Показатель оценивания – знания, умения, навыки (владения)	Уровни сформированности компетенций				Формы и средства контроля формирования компетенций
				компетенция не сформирована	минимальный	средний	высокий	
				Оценки сформированности компетенций				
				2	3	4	5	
				Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»	
				Характеристика сформированности компетенции				
				Компетенция в полной мере не сформирована. Имеющихся знаний, умений и навыков недостаточно для решения практических (профессиональных) задач	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения стандартных практических (профессиональных) задач	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач	
Критерии оценивания								
ОПК-4 Способен выбирать и использовать современные инструментальные методы и технологии, осваивать новые методы и технику исследований для решения конкретных задач профессиональной деятельности	ИД-3опк-4 Выбирает и изучает новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности	Наличие умений	Умеет выбирать и новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности, а также парсить и обрабатывать биоинформатические файлы, используя стандартные средства или библиотеку Biopython, извлекать нужную информацию из файлов.	Не умеет выбирать и новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности, а также парсить и обрабатывать биоинформатические файлы, используя стандартные средства или библиотеку Biopython, извлекать нужную информацию из файлов	Умеет выбирать и некоторое программное обеспечение или библиотеки, следуя официальной документации и научным публикациям, достаточным для решения стандартной конкретной прикладной задачи профессиональной деятельности, а также парсить и обрабатывать биоинформатические файлы, используя некоторые стандартные средства или библиотеку Biopython, извлекать нужную информацию из файлов	Умеет выбирать новое программное обеспечение или библиотеки, следуя официальной документации и научным публикациям, на уровне достаточном для решения конкретной прикладной задачи профессиональной деятельности, а также парсить и обрабатывать биоинформатические файлы, используя основные стандартные средства или библиотеку Biopython, извлекать нужную информацию из файлов	Умеет самостоятельно выбирать новое программное обеспечение или библиотеки, следуя официальной документации и научным публикациям, для решения конкретной прикладной задачи профессиональной деятельности, а также парсить и обрабатывать биоинформатические файлы, используя стандартные средства или библиотеку Biopython, извлекать нужную информацию из файлов	Тестовые задания, расчетно-аналитическая работа

Индекс и название компетенции	Код индикатора достижений компетенции	Индикаторы компетенции	Показатель оценивания – знания, умения, навыки (владения)	Уровни сформированности компетенций				Формы и средства контроля формирования компетенций
				компетенция не сформирована	минимальный	средний	высокий	
				Оценки сформированности компетенций				
				2	3	4	5	
				Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»	
				Характеристика сформированности компетенции				
				Компетенция в полной мере не сформирована. Имеющихся знаний, умений и навыков недостаточно для решения практических (профессиональных) задач	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения стандартных практических (профессиональных) задач	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для решения сложных практических (профессиональных) задач	
Критерии оценивания								
ОПК-4 Способен выбирать и использовать современные инструментальные методы и технологии, осваивать новые методы и технику исследований для решения конкретных задач профессиональной деятельности	ИД-3_ОПК-4 Выбирает и изучает новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности	Наличие навыков (владение опытом)	Владеет навыками выбора программного обеспечения или библиотеки, следуя официальной документации и научным публикациям, для решения конкретной прикладной задачи профессиональной деятельности и его применения. Владеет навыками ускорять работу за счет использования векторизации (NumPy), эффективных структур данных, написания чистого, читаемого и поддерживаемого кода, добавлять документацию и комментарии.	Не владеет навыками выбора программного обеспечения или библиотеки, следуя официальной документации и научным публикациям, для решения конкретной прикладной задачи профессиональной деятельности и его применения. Не владеет навыками ускорять работу за счет использования векторизации (NumPy), эффективных структур данных, написания чистого, читаемого и поддерживаемого кода, добавлять документацию и комментарии	Владеет некоторыми навыками выбора программного обеспечения или библиотеки, следуя официальной документации и научным публикациям, достаточными для решения конкретной прикладной задачи профессиональной деятельности и его применения. Владеет основными навыками достаточными, чтобы ускорять работу за счет использования векторизации (NumPy), стандартных структур данных, написания чистого, читаемого и поддерживаемого кода, добавлять документацию и комментарии	Владеет основными навыками выбора программного обеспечения или библиотеки следуя официальной документации и научным публикациям, достаточными для решения стандартной прикладной задачи профессиональной деятельности и его применения. Владеет основными навыками, достаточными для того, чтобы ускорять работу за счет использования векторизации (NumPy), стандартных структур данных, написания чистого, читаемого и поддерживаемого кода, добавлять документацию и комментарии	Владеет навыками самостоятельного выбора программного обеспечения или библиотеки, следуя официальной документации и научным публикациям, для решения конкретной прикладной задачи профессиональной деятельности и его применения. Владеет навыками ускорять работу за счет использования векторизации (NumPy), эффективных структур данных, написания чистого, читаемого и поддерживаемого кода, добавлять документацию и комментарии	Тестовые задания, расчетно-аналитическая работа

2. Структура учебной работы, содержание и трудоёмкость основных элементов дисциплины

2.1. Организационная структура, трудоемкость и план изучения дисциплины

Дисциплина изучается во 2 семестре 1 курса.

Продолжительность семестра 17 недель.

Вид учебной работы	Трудоемкость, час	
	2 семестр, 1 курс*	
	Очная форма обучения	
1. Контактная работа	78	
1.1. Аудиторные занятия, всего	40	
– лекции	8	
– практические занятия (включая семинары)	-	
– лабораторные работы	32	
1.2. Консультации (в соответствии с учебным планом)	38	
2. Внеаудиторная академическая работа	30	
2.1 Фиксированные виды внеаудиторных самостоятельных работ:	10	
Выполнение и сдача индивидуального задания в виде**		
– расчетно-аналитической работы	10	
2.2 Самостоятельное изучение тем/вопросов программы	8	
2.3 Самоподготовка к аудиторным занятиям	6	
2.4 Самоподготовка к участию и участие в контрольно-оценочных мероприятиях, проводимых в рамках текущего контроля освоения дисциплины (за исключением учтённых в пп. 2.1 – 2.2):	6	
3. Подготовка и сдача экзамена по итогам освоения дисциплины	36	
ОБЩАЯ трудоемкость дисциплины:	Часы	144
	Зачетные единицы	4

Примечание:

* – **семестр** – для очной и очно-заочной формы обучения, **курс** – для заочной формы обучения;

** – КР/КП, реферата/эссе/презентации, контрольной работы (для обучающихся заочной формы обучения), расчетно-графической (расчетно-аналитической) работы и др.;

2.2. Укрупнённая содержательная структура учебной дисциплины и общая схема её реализации в учебном процессе

Номер и наименование раздела дисциплины. Укрупненные темы раздела		Общая	Трудоемкость раздела и ее распределение по видам учебной работы, час.						ВАРС		Формы текущего контроля успеваемости и промежуточной аттестации	№№ компетенций, на формирование которых ориентирован раздел
			Контактная работа				Консультации (в соответствии с учебным планом)					
			Аудиторная работа			всего		Фиксированные виды				
			Всего	лекции	занятия							
				практические (всех форм)	лабораторные							
Очная форма обучения												
1	Python как инструмент научных вычислений и биоинформатики	32	12	2	-	10	10	10	4	Тестовые задания, РАР	ОПК-3 ОПК-4	
	Обзор экосистемы Python для науки: NumPy, SciPy, pandas, matplotlib, seaborn, Biopython											
	Окружение: conda/mamba, Jupyter Lab, VS Code											
	Основы синтаксиса для анализа данных: списковые включения, функции, работа с файлами (FASTA, FASTQ, GFF, VCF)											
2	Высокопроизводительные вычисления и векторизация	38	10	2	-	8	18	10	2	Тестовые задания, РАР	ОПК-3 ОПК-4	
	NumPy. Векторизация и циклы											
	Работа с большими массивами: memory mapping, типы данных. Введение в Numba и Cython для ускорения кода											
	Параллельные вычисления: multiprocessing, concurrent.futures, joblib											
3	Обработка биологических данных. Визуализация научных данных	38	18	4	-	14	10	10	4	Тестовые задания, РАР	ОПК-3 ОПК-4	
	Парсинг геномных аннотаций (GFF3, GTF) и вариаций (VCF)											
	Анализ последовательностей: GC-состав, поиск мотивов, трансляция. Работа с выравниваниями (SAM/BAM) через pysam. Генерация признаков для машинного обучения из биологических данных											
	Принципы визуализации для биоинформатики: Manhattan plot. Интерактивные графики: plotly, altair. Визуализация геномных треков (pyGenomeTracks, Hi-C plots). Создание publication-ready фигур											
	Промежуточная аттестация	36	x	x	x	x	x	x	x	Экзамен		
Итого по дисциплине		144	78	8	-	32	38	30	10			

3. Общие организационные требования к учебной работе обучающегося

3.1. Организация занятий и требования к учебной работе обучающегося

Организация занятий по дисциплине носит циклический характер. По трем разделам предусмотрена взаимоувязанная цепочка учебных работ: лекция – самостоятельная работа обучающихся (аудиторная и внеаудиторная). На занятиях студенческая группа получает задания и рекомендации.

Для своевременной помощи обучающимся при изучении дисциплины кафедрой организуются индивидуальные и групповые консультации, устанавливается время приема выполненных работ.

Учитывая статус дисциплины к её изучению, предъявляются следующие организационные требования:

- обязательное посещение обучающимся всех видов аудиторных занятий;
- ведение конспекта в ходе лекционных занятий;
- качественная самостоятельная подготовка к лабораторным занятиям, активная работа на них;
- активная, ритмичная самостоятельная аудиторная и внеаудиторная работа обучающегося, своевременная сдача преподавателю отчетных документов по аудиторным и внеаудиторным видам работ;
- в случае наличия пропущенных обучающимся занятий, необходимо получить консультацию по подготовке и оформлению отдельных видов заданий.

Для успешного освоения дисциплины, обучающемуся предлагаются учебно-информационные источники в виде учебной, учебно-методической литературы по всем разделам.

3.2. Условия допуска к экзамену (при наличии)

Экзамен является формой контроля, который выставляется обучающемуся согласно «Положения о текущем контроле успеваемости, промежуточной аттестации обучающихся по программам высшего образования (бакалавриат, специалитет, магистратура) и среднего профессионального образования в ФГБОУ ВО Омский ГАУ», выполнившему в полном объеме все перечисленные в п.2-3 требования к учебной работе, прошедший все виды тестирования, выполнения реферата с положительной оценкой. В случае неполного выполнения указанных условий по уважительной причине, обучающемуся могут быть предложены индивидуальные задания по пропущенному учебному материалу.

4. Лекционные занятия

Для изучающих дисциплину читаются лекции в соответствии с планом, представленным в таблице 3.

Таблица 3 – Лекционный курс

№		Тема лекции. Основные вопросы темы	Трудоемкост ь по разделу, час.	Применяемые интерактивные формы обучения
раздела	лекции			
1	1	Python как инструмент научных вычислений и биоинформатики	2	-
		Обзор экосистемы Python для науки: NumPy, SciPy, pandas, matplotlib, seaborn, Biopython		
		Установка окружения: conda/mamba, Jupyter Lab, VS Code		
		Основы синтаксиса для анализа данных: списковые включения, функции, работа с файлами (FASTA, FASTQ, GFF, VCF)		
2	2	Высокопроизводительные вычисления и векторизация	2	Лекция с разбором конкретных ситуаций
		NumPy. Векторизация и циклы.		
		Работа с большими массивами: memory mapping, типы данных. Введение в Numba и Cython для ускорения кода.		
		Параллельные вычисления: multiprocessing, concurrent.futures, joblib.		
3	3-4	Обработка биологических данных. Визуализация научных данных	4	Лекция с разбором конкретных ситуаций
		Парсинг геномных аннотаций (GFF3, GTF) и вариаций (VCF)		
		Анализ последовательностей: GC-состав, поиск мотивов, трансляция. Работа с выравниваниями (SAM/BAM) через pysam. Генерация признаков для машинного обучения из биологических данных		
		Принципы визуализации для биоинформатики: Manhattan plot. Интерактивные графики: plotly, altair. Визуализация геномных треков (pyGenomeTracks, Hi-C plots). Создание publication-ready фигур		
Общая трудоемкость лекционного курса			8	

Всего лекций по дисциплине:	час.	Из них в интерактивной форме:	час.
ОЧНАЯ форма обучения	8	ОЧНАЯ форма обучения	6

5. Лабораторные занятия по дисциплине и подготовка к ним

Лабораторные занятия по курсу проводятся в соответствии с планом, представленным в таблице 4.

Таблица 4 – Примерный тематический план лабораторных занятий по разделам дисциплины

раздела	№		Тема лабораторной работы	Трудовые мкость ЛР, час	Связь с ВАРС		Применяемые интерактивные формы обучения*
	ЛЗ*	ЛР*			предусм отрена самоподг отовка к занятию +/-	Защита отчета о ЛР во внеауди торное время +/-	
1	1-2	1	Синтаксис Python и автоматизация задач. Парсинг FASTA-файла без внешних библиотек. Расчёт GC-состава, поиск ORF. Написание скрипта для фильтрации FASTQ.	4	-	-	-
	3	2	NumPy: от основ к реальным задачам. Скользящее среднее по профилю покрытия (маскирование, операции с осями).	2	+	-	Работа в малых группах
	4-5	3	Генерация синтетических данных: моделирование ошибок секвенирования, добавление известных мутаций. Матричное попарное выравнивание профилей метилирования (ковариационные матрицы, собственные векторы).	4	+	-	-
2	6	4	Биоинформатические библиотеки. Biopython: чтение GenBank, извлечение CDS, транскрипция и трансляция. pysam: работа с BAM-файлом (глубина покрытия в заданных координатах, фильтрация по MAPQ).	2	+	-	Работа в малых группах
	7	5	Pandas + pyarrow: построение таблицы «образец → ген → число редких вариантов» из VCF	2	+	-	-
	8-9	6	Объединение данных: аннотация вариантов из VCF с помощью аннотационного GFF/	4	+	-	-
3	10-12	8	Визуализация для биологии. Построение volcano plot (log2FC vs -log10 p-value) для RNA-seq. Тепловая карта корреляций экспрессии генов (seaborn.clustermap). Визуализация структурных вариантов (circos-plot через plotly или просто линейный трек)	6	+	-	Работа в малых группах
	13-16	9	Решение научных задач. Разработка конвейера: «от FASTQ до матрицы признаков» (качество → GC-состав → покрытие по генам → визуализация). Оптимизация конвейера (профилирование, векторизация, параллельные вычисления) и оформление отчёта в формате Jupyter Notebook.	8	-	-	-
Итого ЛР		9	Общая трудоемкость ЛР	32	x		

Подготовка обучающихся к лабораторным занятиям осуществляется с учетом общей структуры учебного процесса. На лабораторных занятиях осуществляется входной и текущий аудиторный контроль в виде опроса, по основным понятиям дисциплины.

Подготовка к лабораторным занятиям подразумевает выполнение домашнего задания к очередному занятию по заданиям преподавателя, выдаваемым в конце предыдущего занятия, а также изучение теоретического материала.

Для осуществления работы по подготовке к занятиям, необходимо ознакомиться с методическими указаниями по дисциплине, внимательно ознакомиться с литературой и электронными ресурсами, с рекомендациями по подготовке, вопросами для самоконтроля.

6. Общие методические рекомендации по изучению отдельных разделов дисциплины

При изучении конкретного раздела дисциплины, из числа вынесенных на лекционные и практические занятия, обучающемуся следует учитывать изложенные ниже рекомендации. Обратите на них особое внимание при подготовке к аттестации.

Работа по теме прежде всего предполагает ее изучение по учебнику или пособию. Следует обратить внимание на то, что в любой теории, есть либо неубедительные, либо чересчур абстрактные, либо сомнительные положения. Поэтому необходимо вырабатывать самостоятельные суждения, дополняя их аргументацией, что и следует демонстрировать на семинарах. Для выработки самостоятельного суждения важным является умение работать с научной литературой. Поэтому работа по теме кроме ее изучения по учебнику, пособию предполагает также поиск по теме научных статей в научных журналах. Такими журналами являются: Программные продукты и системы. – Тверь : Научно-исследовательский институт «Центрпрограммсистем», 1988. – Выходит 4 раза в год. – ISSN 2311-2735. – Текст : электронный. – URL: <https://eivis.ru/browse/publication/64086>. Выбор статьи, относящейся к теме, лучше делать по последним в году номерам, где приводится перечень статей, опубликованных за год.

Самостоятельная подготовка предполагает использование ряда методов.

1. Конспектирование. Конспектирование позволяет выделить главное в изучаемом материале и выразить свое отношение к рассматриваемой автором проблеме.

Техника записей в конспекте индивидуальна, но есть ряд правил, которые могут принести пользу его составителю: начиная конспект, следует записать автора изучаемого произведения, его название, источник, где оно опубликовано, год издания. Порядок конспектирования:

- а) внимательное чтение текста;
- б) поиск в тексте ответов на поставленные в изучаемой теме вопросы;
- в) краткое, но четкое и понятное изложение текста;
- г) выделение в записи наиболее значимых мест;
- д) запись на полях возникающих вопросов, понятий, категорий и своих мыслей.

2. Записи в форме тезисов, планов, аннотаций, формулировок определений. Все перечисленные формы помогают быстрой ориентации в подготовленном материале, подборе аргументов в пользу или против какого-либо утверждения.

3. Словарь понятий и категорий. Составление словаря помогает быстрее осваивать новые понятия и категории, увереннее ими оперировать. Подобный словарь следует вести четко, разборчиво, чтобы удобно было им пользоваться. Из приведенного в УМК глоссария нужно к каждому семинару выбирать понятия, относящиеся к изучаемой теме, объединять их логической схемой в соответствии с вопросами семинарского занятия.

Раздел 1. Python как инструмент научных вычислений и биоинформатики

Краткое содержание

1. Обзор экосистемы Python для науки: NumPy, SciPy, pandas, matplotlib, seaborn, Biopython.
2. Установка окружения: conda/mamba, Jupyter Lab, VS Code.
3. Основы синтаксиса для анализа данных: списковые включения, функции, работа с файлами (FASTA, FASTQ, GFF, VCF).

Вопросы для самоконтроля по разделу:

1. Опишите основные возможности библиотеки NumPy.
2. Перечислите основные этапы работы с библиотекой NumPy
3. Опишите основные атрибуты при работе с массивами в библиотеке NumPy.
4. Что такое векторизация операций в NumPy? Приведите пример, где векторизованный код работает быстрее обычного цикла *for*.
5. Опишите особенности применения библиотеки SciPy.
6. Охарактеризуйте основные структуры данных в библиотеке pandas: *Series* и *DataFrame*. Как создать их из словаря или списка?
7. Какие методы используются для чтения биологических файлов форматов CSV/TSV в *DataFrame*? Как обрабатывать файлы с нестандартными разделителями?
8. Как выполнять фильтрацию, сортировку и группировку данных в объекте *DataFrame*? Приведите примеры кода.
9. Как работать с пропущенными данными (NaN) в pandas? Какие есть методы для их обнаружения и заполнения?

10. Перечислите основные методы для работы со списками (например, `append()`, `pop()`, `insert()`, `remove()`).
11. Какие библиотеки входят в основной стек для научных вычислений на Python (SciPy Stack)? Кратко опишите назначение каждой из них: *NumPy*, *SciPy*, *pandas*, *Matplotlib*.
12. В чем разница между списками (*list*) и массивами NumPy (*np.array*)? Когда следует использовать каждый из этих типов данных?
13. Как установить необходимые научные пакеты Python? Опишите использование менеджера пакетов *pip* и дистрибутива *Anaconda/Miniconda*.
14. Что такое виртуальное окружение (*virtualenv*, *conda env*) и зачем оно нужно при работе над проектами по биоинформатике?
15. Опишите возможности библиотек: *matplotlib*, *seaborn*, *Biopython*.
16. Охарактеризуйте работу с файлами (FASTA, FASTQ, GFF, VCF).

Раздел 2. Высокопроизводительные вычисления и векторизация

Краткое содержание

1. NumPy. Векторизация и циклы.
2. Работа с большими массивами: *memory mapping*, типы данных. Введение в *Numba* и *Cython* для ускорения кода.
3. Параллельные вычисления: *multiprocessing*, *concurrent.futures*, *joblib*.

Вопросы для самоконтроля по разделу:

1. Что такое векторизация в контексте Python и NumPy? В чем заключается ее основной принцип и почему она работает быстрее, чем использование циклов `for`?
2. Что такое "broadcasting" (вещание) в NumPy? Приведите пример операции с массивами разной формы, где применяется broadcasting.
3. Каковы основные причины низкой производительности "чистого" Python при обработке больших объемов данных? (Подсказка: рассмотрите понятия GIL, динамической типизации и интерпретации).
4. Сравните производительность (с точки зрения скорости и читаемости) трех способов возведения элементов списка в квадрат: (а) цикл `for`, (б) генератор списков (*list comprehension*), (в) векторизованная операция NumPy.
5. Что такое "временная сложность" (*Big O notation*) алгоритма? Какова временная сложность поиска элемента в списке и в словаре?
6. Как профилировать код на Python? Назовите инструменты (например, *cProfile*, `%timeit` в *Jupyter*) и объясните, какую информацию они предоставляют.
7. Опишите концепцию "ленивых вычислений" (*lazy evaluation*). Как она реализована в библиотеках *Dask* и как помогает экономить память?
8. Что такое GIL (Global Interpreter Lock) в CPython? Как он влияет на многопоточность для задач, связанных с вычислениями на CPU?
9. В чем принципиальная разница между модулями *threading* и *multiprocessing* в Python? Когда следует использовать каждый из них?
10. Опишите модель параллелизма на основе процессов. Как происходит обмен данными между процессами (например, через *Queue* или *Pipe*)?
11. Как работает библиотека *concurrent.futures*? Сравните использование *ThreadPoolExecutor* и *ProcessPoolExecutor*.
12. Для каких задач предназначена библиотека *Dask*? Как она позволяет масштабировать вычисления с NumPy/pandas на несколько ядер или даже на кластер?
13. Сравните *Dask DataFrame* и *pandas DataFrame*. В чем их сходства и ключевые различия в контексте работы с большими данными?
14. Что такое библиотека *Ray*? В чем ее отличие от *Dask* и для каких современных задач (например, машинное обучение) она особенно эффективна?
15. Как можно использовать GPU для ускорения вычислений на Python? Назовите основные библиотеки (например, *CuPy*, *PyTorch*, *TensorFlow*) и объясните принцип их работы.
16. Напишите векторизованный код на NumPy, который создает массив из 10 миллионов случайных чисел и вычисляет их среднее значение.
17. Перепишите следующий код с использованием векторизации:

```
result = []
for i in range(len(a)):
    result.append(a[i] * b[i])
```

(где *a* и *b* — одномерные массивы).

18. Опишите шаги, которые необходимо выполнить для запуска Python-скрипта на удаленном HPC-кластере с помощью Slurm: от написания кода до получения результатов.
19. Почему `pandas.apply()` часто работает медленнее, чем векторизованная операция через NumPy?
20. В чем разница между `np.array` и `np.ndarray`?

Раздел 3. Обработка биологических данных. Визуализация научных данных

Краткое содержание

1. Парсинг геномных аннотаций (GFF3, GTF) и вариаций (VCF).
2. Анализ последовательностей: GC-состав, поиск мотивов, трансляция. Работа с выравниваниями (SAM/BAM) через `pySAM`. Генерация признаков для машинного обучения из биологических данных.
3. Принципы визуализации для биоинформатики: Manhattan plot. Интерактивные графики: `plotly`, `altair`. Визуализация геномных треков (`pyGenomeTracks`, Hi-C plots). Создание publication-ready фигур.

Вопросы для самоконтроля по разделу:

1. Назовите основные типы графиков, используемых в биоинформатике. Для каких задач применяются гистограммы, диаграммы рассеяния (*scatter plot*) и тепловые карты (*heatmap*)?
2. Как построить простой график функции (например, распределение GC-состава) с помощью Matplotlib? Из каких компонентов состоит базовый график (осей, линии, подписи)?
3. Чем библиотека Seaborn отличается от Matplotlib? В чем ее преимущество при визуализации статистических данных?
4. Как можно визуализировать результаты анализа дифференциальной экспрессии генов? (Например, с помощью графика вулкана — Volcano Plot).
5. Для чего используется библиотека Biopython? Назовите несколько основных модулей и их функций (например, работа с последовательностями, парсинг форматов FASTA, GenBank).
6. Как с помощью Biopython прочитать файл в формате FASTA и получить доступ к идентификатору и последовательности записи?
7. Что такое формат SAM/BAM и какие задачи он решает? Какая основная Python-библиотека используется для работы с этими файлами?
8. Какую роль играют Jupyter Notebook / JupyterLab в анализе биологических данных? Почему они так популярны среди ученых?
9. Объясните концепцию "Snakemake" или "Nextflow". Зачем нужны системы управления рабочими процессами (workflow management systems) в биоинформатических пайплайнах?
10. Напишите однострочник на Python, который считает количество строк в файле (аналог команды `wc -l`).
11. Вычислите среднее значение и стандартное отклонение для заданного числового массива, используя только NumPy.
12. Как профилировать код на Python, чтобы найти узкие места в производительности? Назовите хотя бы один инструмент.
13. Что такое "List Comprehension" и когда его применение предпочтительнее использования цикла `for`?
14. Опишите типичный сценарий анализа данных РНК-секвенирования (RNA-seq), указав, на каком этапе какие Python-инструменты были бы задействованы (от контроля качества до функционального аннотирования).
15. Напишите цикл `for`, который выводит на экран все четные числа от 0 до 20 включительно.
16. Создайте функцию, которая принимает список чисел и возвращает их сумму.
17. Напишите программу, которая запрашивает у пользователя имя и возраст, а затем выводит приветственное сообщение с использованием *f-строки*.
18. Даны данные об успеваемости 10 студентов по 3 дисциплинам. Напишите код, который найдет и выведет предмет с самой высокой оценкой.
19. Напишите код, который, создает список, содержащий квадраты чисел от 1 до 10.
20. Опишите возможности библиотеки Pandas.

Процедура оценивания

После изучения каждого раздела проводится контроль. Контроль осуществляется с целью определения качества проведения образовательных услуг по дисциплине, для оценки степени достижения обучающимися состояния, определяемого целевыми установками дисциплины, а также для формирования корректирующих мероприятий. Контроль осуществляется по разделам дисциплины в соответствии с планом. Контроль состоит из выполнения заданий, работ на лабораторных занятиях, тестовых заданий.

Шкала и критерии оценивания

Оценка «зачтено» выставляется, если студент на основе самостоятельного изученного материала, смог всесторонне раскрыть теоретическое содержание вопросов, владеет методиками при решении практических задач.

Оценка «не зачтено» выставляется, если студент не смог раскрыть теоретическое содержание вопросов, не владеет методиками при решении практических задач или выполнил несамостоятельно.

7. Общие методические рекомендации по оформлению и выполнению отдельных видов ВАРС

7.1 Рекомендации по выполнению индивидуального задания расчетно-аналитической работы (РАР)

Учебные цели, на достижение которых ориентировано выполнение реферата:

1. Получить целостное представление о систематизации и закреплении теоретических знаний.
2. Развитие практических навыков по решению задач с применением программирования.
3. Выработка навыков анализа статистических и аналитических данных и формулирования выводов по полученным результатам.

Выполнение расчетно-аналитической работы проводится по конкретному варианту задания, который необходимо уточнить у преподавателя.

Типовые задачи расчетно-аналитической работы

Прикладная задача профессиональной деятельности с применением среды программирования Python.

Задания для выполнения:

1. По исходным данным научной задачи и заданиям к ней написать программный код на языке программирования Python с применением его библиотек, провести его тестирование на заданных исходных данных и в случае необходимости выполнить отладку программы или ускорить его работу;
2. Сформулировать пояснительную записку по процессу выполнения решения задачи и анализу полученных результатов средствами прикладных программ;
3. Для решения поставленной задачи использовать библиотеки Python (рекомендуемые): Biopython (парсинг FASTA, обратная комплементарность, трансляция).
 - numpy / pandas (обработка данных, скользящие окна).
 - matplotlib / seaborn (построение графиков).
 - scipy (статистические тесты: t-критерий, Z-score для выявления выбросов).
 - Встроенные модули: math, collections, re, os, sys.

Правила оформления расчетно-аналитической работы

Выполнение расчетно-графической работы проводится обучающимся по конкретному варианту N задания, который необходимо уточнить у преподавателя. Индивидуализация решения РАР обеспечивается за счёт разных исходных данных.

Отчет выполнения РАР обучающийся должен представить в виде файла текстового документа. Решение каждой задачи должно быть оформлено с нового раздела и содержать условие задачи, листинг программы, скриншот отладки программы и проверку результатов, с использованием табличного процессора.

Процедура оценивания

По итогам выполнения РАР проводятся следующие контрольные мероприятия: преподавателю для проверки сдается оформленная РАР в ЭИОС или в электронном виде на занятиях. Если имеются замечания по работе, то обучающийся исправляет указанные ошибки и отправляет работу на повторное рецензирование. При достаточно большом количестве замечаний по работе, проводится разбор ошибок на аудиторном занятии.

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ

Оценка «зачтено» выставляется, если решение заданий оформлено грамотно, в частности методы решения, формы его записи и формы записи ответа могут быть разными. Обоснованно получен верный программный код или полученный код дает неверный ответ из-за негрубой ошибки, но при этом имеется верная последовательность всех шагов решения или допущена единичная ошибка, возможно, приведшая к неверному ответу, но при этом имеется верная последовательность всех шагов решения.

Оценка «не зачтено» выставляется, если решение заданий оформлено неграмотно, получен неверный программный код из-за неверной последовательности всех шагов решения, или решено не самостоятельно.

7.2. Рекомендации по самостоятельному изучению тем

Преподаватель в начале изучения дисциплины выдает темы для самостоятельного изучения, определяет форму представления отчетных материалов в виде конспекта.

ВОПРОСЫ

для самостоятельного изучения темы

«Менеджеры пакетов. pip (для PyPI) и conda (для Anaconda)»

1. Какие фундаментальные различия между pip и conda в подходе к управлению пакетами и средами разработки? Какие задачи каждый из них выполняет лучше?
2. Как работает разрешение зависимостей в pip и как он обрабатывает зависимости, конфликты версий и зависимости «платформенно-зависимого» кода?
3. Что такое виртуальные окружения в Python и как их использовать с pip (venv/venv + pip) vs. управляемые окружения в conda? Какие плюсы и минусы каждого подхода?
4. Как создаются и активируются окружения conda и какие файлы конфигурации (.yaml) используются для воспроизводимости сред?
5. Как устанавливать конкретные версии пакетов, roll back версий и избегать конфликтов при обновлениях в pip и conda? Приведите примеры команд.
6. Что такое каналы (repositories) в conda и как управлять источниками пакетов (defaults, conda-forge)? Какие риски и преимущества у разных источников?
7. Как управлять зависимостями между пакетами с помощью conda и как использовать окружения для изолирования проектов с различными версиями Python и библиотек?
8. Какие механизмы кэширования, зеркалирования и фиксации версий пакетов существуют в pip и conda для воспроизводимости (pip freeze, Pipfile.lock, poetry.lock, конда-каналы и environment.yml)?
9. Как обрабатывать бинарные зависимые пакеты и нативный код в рамках pip и conda: различия в сборке, компиляции и наличии предсобранных «wheel» пакетов vs. нативных бинарников из conda-forge?
10. Какие стратегии тестирования и воспроизводимости вы применяете при работе с двумя менеджерами пакетов: сценарии миграции между ними, совместное использование инструментов (pip + conda) и советы по устойчивости окружений в научных проектах?

ВОПРОСЫ

для самостоятельного изучения темы

«NumPy и SciPy библиотеки языка Python в научных и инженерных расчетах с поддержкой высокоуровневых математических функций»

1. Особенности и назначение библиотек NumPy и SciPy для языка Python
2. Приведите примеры задач, решаемые с помощью библиотеки NumPy
3. Приведите примеры задач, решаемые с помощью библиотеки SciPy
4. Создайте словарь, где ключи — названия фруктов, а значения — их количество. Выведите все фрукты, количество которых больше 5.
5. Какие базовые концепции лежат в основе NumPy (ndarray, dtype, broadcasting, векторизация) и чем они отличаются от обычных списков Python по памяти и скорости выполнения?
6. Как выполнять эффективную работу с массивами: создание, преобразование форм, индексация, срезы, копирование vs. представления, и как избегать ненужных копирований?
7. Какие способы решения систем линейных уравнений, вычисления собственных значений/векторов и факторизации матриц предоставляет NumPy/SciPy, и в каких случаях полезнее использовать LU/Cholesky/QR разложения?
8. Как работать с линейной алгеброй в SciPy (mod scipy.linalg) и чем отличается функционал по сравнению с NumPy, включая специальные функции для матриц разреженности?
9. Как применять высокоуровневые математические функции SciPy: оптимизацию (минимум/максимум), интегрирование (одномерное и многомерное), интерполяцию и аппроксимацию данных?
10. Какие алгоритмы численной интеграции и дифференциальных уравнений доступны в SciPy (odeint, solve_ivp) и как выбрать метод с учётом устойчивости и шага по времени?
11. Как использовать SciPy для анализа сигналов и обработки изображений: частотный спектр (FFT), фильтры, сглаживание, интерполяция изображений и преобразования Фурье?

12. Какие средства для работы с данными в SciPy и NumPy подходят для статистического анализа и инструментов моделирования (распределения, регрессия, бутстрэп, Байесовская инфраструктура в SciPy/Statsmodels)?

13. Как работать с разреженной линейной алгеброй: создание и операции над разреженными матрицами (CSR/CSC), разреженные раскладки и эффективные алгоритмы умножения и решения систем?

14. Как реализовать эффективные вычисления на больших массивах: векторизация vs. использование функций из numpy/scipy.sparse, параллелизм через multiprocessing/Dask, и какие профилировочные инструменты помогут локализовать «узкие места» производительности?

ВОПРОСЫ

для самостоятельного изучения темы

«Объектно-ориентированное программирование, итераторы и декораторы кода Python. Тестирование, профилирование кода. Обработка исключительных ситуаций»

1. Объектно-ориентированное программирование (ООП)
2. Итераторы и генераторы
3. Декораторы кода
4. Обработка исключительных ситуаций
5. Тестирование и профилирование кода.
6. Какие принципы SOLID применимы к проектированию классов в Python, и как они влияют на читаемость, расширяемость и тестируемость объектов?
7. Как реализовать итератор и итерируемый объект в Python (iter, next), чем отличается итератор от генератора и как это влияет на память и производительность?
8. Что такое паттерн "генератор-выражение" и как использовать его совместно с итераторами для ленивых вычислений и экономии памяти?
9. Какие подходы и техники применяются для реализации декораторов функций и методов: простой декоратор, декоратор с сохранением метаданных (functools.wraps), и как декораторы влияют на стек вызовов и инструменты дебаггинга?
10. Как реализовать параметризованные декораторы и декораторы с аргументами, включая хранение состояния между вызовами и влияние на тестирование?
11. Какие типы тестирования применимы к объектно-ориентированному дизайну: модульное тестирование классов, тесты поведения (mock/stub), тестирование взаимодействий между объектами и паттернами мокирования/фиктивности?
12. Как проводить профилирование кода Python: какие инструменты выбрать (cProfile, line_profiler, memory_profiler, Py-Spy) и как интерпретировать результаты для устранения узких мест?
13. Какие подходы к обработке исключений в Python оптимальны: когда ловить конкретные исключения vs. универсально "except Exception", как сохранять контекст исключения и зачем использовать raise from, а также как реализовать собственные исключения для ясности API?
14. Как проектировать тестируемые и устойчивые к ошибкам API классов: контрактный дизайн (type hints, докстринги), валидация входных данных в конструкторе и методах, обработка крайних случаев и явная сигнализация ошибок через исключения.
15. Какие стратегии мониторинга и устойчивости к ошибкам применимы в больших проектах: журналирование (logging), уровни логирования, трассировка стеков, тестирование регрессий на трассах исключений и интеграционные тесты для сценариев использования.

ВОПРОСЫ

для самостоятельного изучения темы

«Jupyter Notebook / JupyterLab. Интерактивная среда в Data Science и биоинформатике»

1. Что такое Jupyter Notebook и чем он отличается от JupyterLab? Опишите ключевые особенности и преимущества каждого из них.
2. Что такое Jupyter Notebook и JupyterLab, чем отличается интерактивная среда от обычной IDE и какие преимущества это дает для анализа данных и биоинформатики?
3. Как создать и организовать проекты в JupyterLab: структура рабочих директорий, использование версий файлов (.ipynb), и принципы модульности кода (ноутбуки vs. модули Python).
4. Как устанавливать и управлять средами окружения для Jupyter (conda/venv), а также как избегать конфликтов зависимостей при работе с биоинформатическими пакетами (Biopython, scikit-bio и т. п.)?
5. Какие инструменты визуализации доступны в Jupyter (Matplotlib, Seaborn, Plotly, Bokeh) и как реализовать интерактивные графики для биоинформатических данных (геномные области, экспрессия генов, временные ряды)?

6. Как реализовать воспроизводимые научные отчёты и ноутбуки: экспорты в HTML/Notebook, экспорт в PDF/LaTeX, интеграция с авто-генерацией документации и отчётов для публикаций?
7. Какие подходы к управлению версиями кода и данных в рамках Jupyter (Git, Jupyter Book, nbdev) особенно полезны в биоинформатике для отслеживания изменений и совместной работы?
8. Какие языки программирования помимо Python поддерживаются в Jupyter? Приведите примеры (например, R, Julia) и объясните, как добавить поддержку нового языка.
9. Опишите жизненный цикл ячейки: режим команд (Command Mode) и режим редактирования (Edit Mode), назначение основных сочетаний клавиш в каждом режиме.
10. Почему не рекомендуется использовать системный интерпретатор Python для работы над проектами в Jupyter? Сравните использование Jupyter Notebook для исследовательского анализа данных (EDA) и для написания производственного кода. Какие есть плюсы и минусы в обоих случаях?
11. Приведите конкретные примеры использования Jupyter в биоинформатике: Парсинг и первичный анализ файлов FASTA, FASTQ.
12. Работа с геномными аннотациями в формате GFF/BED.
13. Визуализация результатов секвенирования РНК (например, построение вулкан-плота).
14. Как библиотеки визуализации (matplotlib, seaborn, plotly) интегрируются с Jupyter? В чем преимущество использования %matplotlib inline? Что такое "интерактивные графики" и как их создавать?
15. Опишите типичный рабочий процесс Data Scientist'a в Jupyter: от загрузки сырых данных до построения финальной модели и сохранения артефактов.

Общий алгоритм самостоятельного изучения темы

1. Ознакомиться с рекомендованной учебной литературой и электронными ресурсами по теме (ориентируясь на вопросы для самоконтроля).
2. На этой основе составить развёрнутый план изложения темы
3. Выбрать форму отчетности конспектов (план – конспект, текстуальный конспект, свободный конспект, конспект – схема)
4. Оформить отчётный материал в установленной форме в соответствии методическими рекомендациями
5. Провести самоконтроль освоения темы по вопросам, выданным преподавателем
6. Предоставить отчётный материал преподавателю по согласованию с ведущим преподавателем
7. Подготовиться к предусмотренному контрольно-оценочному мероприятию по результатам самостоятельного изучения темы
8. Принять участие в указанном мероприятии, пройти итоговое тестирование в установленное для внеаудиторной работы время

Рекомендации по написанию конспекта

Конспект – учебно-теоретический материал, изложенный в компактной форме, который отражает основные вопросы изучаемой темы дисциплины. Конспект представляет собой тезисы наиболее существенной информацией темы в компактной форме и может включать схему логических связей, таблицы и т.п. Содержание конспекта должно соответствовать теме, в конспекте представлять систематизированное, логически последовательное изложение теоретического материала в компактной форме. Конспект выполняется каждым обучающимся письменно в отдельной тетради или электронном виде. Конспект по отдельно выбранной теме должен состоять из названия темы, источника, краткого изложения материала. Титульный лист размещен в приложении 1.

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ самостоятельного изучения темы

Оценка «зачтено» выставляется, если студент на основе самостоятельного изученного материала, смог раскрыть основное теоретическое содержание темы и выполнил предложенные тестовые задания (не менее 60%).

Оценка «не зачтено» выставляется, если студент не смог всесторонне раскрыть основное теоретическое содержание темы и выполнил предложенные тестовые задания (менее 60%).

8. Текущий контроль успеваемости

В течение семестра, проводится текущий контроль успеваемости по дисциплине, к которому обучающийся должен быть подготовлен.

Отсутствие пропусков аудиторных занятий, активная работа на практических занятиях, общее выполнение графика учебной работы являются основанием для получения положительной оценки по текущему контролю.

В качестве текущего контроля использован тестовый контроль. Тест состоит из небольшого количества элементарных вопросов по основным разделам дисциплины: неправильные решения разбираются на следующем занятии; частота тестирования определяется преподавателем.

ВОПРОСЫ и ЗАДАЧИ для самоподготовки к лабораторным занятиям

В процессе подготовки к лабораторному занятию обучающийся изучает представленные ниже вопросы по темам. На занятии обучающийся демонстрирует свои знания по изученным вопросам в форме устного ответа. Представляет реферат. Для усвоения материала по теме занятия обучающийся решает задачи.

Общий алгоритм самоподготовки

Тема 1. Синтаксис Python и автоматизация задач. Парсинг FASTA-файла без внешних библиотек. Расчёт GC-состава, поиск ORF. Написание скрипта для фильтрации FASTQ .

1. Синтаксис Python и автоматизация.
 2. Парсинг FASTA-файла без внешних библиотек.
 3. Расчёт GC-состава.
 4. Поиск ORF (открытых рамок считывания).
 5. Написание скрипта для фильтрации FASTQ.
 6. Проанализировать структуру скрипта для фильтрации FASTQ
- Этот пример иллюстрирует базовый подход к обработке данных секвенирования нового поколения (NGS) с использованием только стандартных инструментов Python.

```
def filter_fastq(input_file, output_file, min_length=50, min_quality=20):
    with open(input_file, 'r') as infile, open(output_file, 'w') as outfile:
        while True:
            header = infile.readline().strip()
            if not header:
                break # Конец файла
            seq = infile.readline().strip()
            _ = infile.readline() # Разделитель '+'
            qual = infile.readline().strip()

            # Фильтрация по длине
            if len(seq) < min_length:
                continue

            # Фильтрация по качеству
            avg_qual = sum(ord(c) - 33 for c in qual) / len(qual)
            if avg_qual < min_quality:
                continue

            # Запись прошедшей фильтрацию записи
            outfile.write(f"{header}\n{seq}\n+\n{qual}\n")
```

Тема 2. NumPy: от основ к реальным задачам. Скользящее среднее по профилю покрытия (маскирование, операции с осями)

1. Основы NumPy и векторизация
2. Профиль покрытия (Coverage Profile)
3. Маскирование (Masking) и булевы индексы
4. Операции с осями (Axis Operations)
5. Скользящее среднее (Moving Average)

Тема 3. Генерация синтетических данных: моделирование ошибок секвенирования, добавление известных мутаций. Матричное попарное выравнивание профилей метилирования (ковариационные матрицы, собственные векторы)

1. Генерация синтетических данных
2. Матричное попарное выравнивание профилей метилирования. Ковариационные матрицы. Собственные векторы и значения (PCA). Попарное выравнивание через расстояние

Тема 4. Биоинформатические библиотеки. Biopython: чтение GenBank, извлечение CDS, транскрипция и трансляция. pysam: работа с BAM-файлом (глубина покрытия в заданных координатах, фильтрация по MAPQ)

1. Biopython: работа с аннотациями и последовательностями.
2. pysam: работа с BAM-файлами.
3. Проанализировать решение задачи: Определить, сколько ридов покрывает каждую позицию в определённом участке генома (например, в экзоне гена).
Метод `.count()`: Позволяет быстро подсчитать общее количество ридов, перекрывающих указанный геномный интервал.
Подсчёт ридов в интервале chr1:1000-2000
`coverage = bamfile.count(contig="chr1", start=1000, stop=2000)`
Метод `.pileup()`: Более мощный инструмент, который позволяет пройти по каждой позиции в заданном интервале и получить доступ ко всем ридам, покрывающим эту позицию. Это необходимо для более сложного анализа (например, поиска SNP).
4. Фильтрация по MAPQ. Глобальная фильтрация.

Тема 5. Pandas + pyarrow: построение таблицы «образец → ген → число редких вариантов» из VCF.

1. Структура и чтение VCF-файла
2. Подготовка данных в Pandas
3. Агрегация и построение матрицы
4. Оптимизация производительности

Тема 6. Объединение данных: аннотация вариантов из VCF с помощью аннотационного GFF.

1. Форматы данных: VCF и GFF. Назначения и ключевые слова для аннотаций
2. Стратегии объединения данных. Поиск по интервалам. Прямой перебор
3. Процесс аннотации: шаг за шагом
4. Сложности и нюансы

Тема 7. Визуализация для биологии. Построение volcano plot ($\log_2FC$ vs $-\log_{10} p\text{-value}$) для RNA-seq. Тепловая карта корреляций экспрессии генов (`seaborn.clustermap`). Визуализация структурных вариантов (`circos-plot` через `plotly` или просто линейный трек).

1. Volcano Plot (Вулканический график) для RNA-seq.
2. Тепловая карта корреляций экспрессии генов.
3. Визуализация структурных вариантов (SV)
4. Линейный трек (Genome Browser-like plot)
5. Круговая диаграмма (Circos Plot)
6. Визуализация SV: Для отображения варианта (например, транслокации между chr1 и chr2) рисуется кривая (`ribbon`), которая начинается на треке одной хромосомы и заканчивается на треке другой.

Тема 8. Решение научных задач. Разработка конвейера: «от FASTQ до матрицы признаков» (качество → GC-состав → покрытие по генам → визуализация). Оптимизация конвейера (профилирование, векторизация, параллельные вычисления) и оформление отчёта в формате Jupyter Notebook.

1. Разработка конвейера: «от FASTQ до матрицы признаков».
2. Оптимизация конвейера.
3. Оформление отчёта в формате Jupyter Notebook.

Оценка «зачтено» выставляется обучающемуся, если он правильно выполнил лабораторную работу и при выполнении работы были допущены несущественные недочеты.

Оценка «не зачтено» выставляется студенту, если он не выполнил лабораторную работу и допустил множество ошибок при её выполнении.

9. Промежуточная (семестровая) аттестация по курсу

9.1. Нормативная база проведения промежуточной аттестации обучающихся по результатам изучения дисциплины:	
1) действующее «Положение о текущем контроле успеваемости, промежуточной аттестации обучающихся по программам высшего образования (бакалавриат, специалитет, магистратура) и среднего профессионального образования в ФГБОУ ВО Омский ГАУ»	
9.2. Основные характеристики промежуточной аттестации обучающихся по итогам изучения дисциплины	
Цель промежуточной аттестации –	Установление уровня достижения каждым обучающимся целей обучения по данной дисциплине, изложенных в п.2.2 настоящей программы
Форма промежуточной аттестации –	Экзамен
Место экзамена в графике учебного процесса:	1. Подготовка к экзамену и сдача экзамена осуществляется за счёт учебного времени (трудоемкости), отведённого на экзаменационную сессию для обучающихся, сроки которой устанавливаются приказом по университету 2. Дата, время и место проведения экзамена определяется графиком сдачи экзаменов, утверждаемым деканом выпускающего факультета
Форма экзамена –	<i>Смешанный (письменный, устный)</i>
Процедура проведения экзамена –	Представлена в фонде оценочных средств по дисциплине (см. Приложение 9)
Экзаменационная программа по учебной дисциплине:	1. Представлена в фонде оценочных средств по дисциплине (см. Приложение 9). 2. Охватывает разделы (в соответствии с п. 4.1 настоящего документа)
Методические материалы, определяющие процедуры оценивания знаний, умений, навыков:	Представлены в фонде оценочных средств по дисциплине (см. Приложение 9)

9.3. Заключительное тестирование по итогам изучения дисциплины

По итогам изучения дисциплины, обучающиеся проходят заключительное тестирование. Тестирование является формой контроля, направленной на проверку владения терминологическим аппаратом и конкретными знаниями в области фундаментальных и прикладных дисциплин.

Подготовка к заключительному тестированию по итогам изучения дисциплины

Тестирование осуществляется по всем темам и разделам дисциплины, включая темы, выносимые на самостоятельное изучение. Процедура тестирования ограничена во времени и предполагает максимальное сосредоточение обучающегося на выполнении теста, содержащего несколько тестовых заданий.

Тестирование проводится в электронной виде. Тест включает в себя 20 вопросов. Время, отводимое на выполнение теста – 40 минут. В каждый вариант теста включаются вопросы разных типов (одиночный и множественный выбор, открытые (ввод ответа с клавиатуры), на упорядочение, соответствие и др.). На тестирование выносятся вопросы из каждого раздела дисциплины.

9.4. Процедура проведения экзамена

При явке на экзамен, обучающийся обязан иметь при себе зачетную книжку, которую он предъявляет экзаменатору в начале экзамена. Экзамен проводится в смешанной форме (устной и письменной форме), по билетам, составленным в соответствии с программой курса. Экзаменатору предоставляется право задавать обучающемуся вопросы в соответствии с учебной программой.

Перечень примерных вопросов к экзамену

1. Базовые типы данных и структуры в Python (int, float, str, bool). Ключевое отличие списков (list), кортежей (tuple) и множеств (set). Приведите примеры их использования при хранении биологических данных (например, список последовательностей, набор уникальных генов).
2. Функции в Python. Механизм передачи аргументов (по значению или по ссылке) для изменяемых и неизменяемых объектов.
3. Работа с файлами. Стандартный способ открытия, чтения и записи файлов с помощью конструкции with ... as. Считывание файла в формате FASTA и извлечение из него идентификаторов и последовательностей.
4. Обработка ошибок: блоки try/except. Примеры обработки ошибки, которая может возникнуть при чтении некорректного биологического файла (например, файл не найден или имеет неверный формат).
5. Библиотека NumPy. Отличие обычных списков Python от массивов numpy.ndarray.
6. Векторизация операций в NumPy и ее важность для научных вычислений. Сравните скорость поэлементного сложения двух больших массивов с помощью цикла for и с использованием векторизованной операции NumPy.
7. Библиотека Pandas. Опишите два основных объекта библиотеки Pandas: Series и DataFrame. Как можно создать DataFrame из словаря? Представьте, что у вас есть таблица с результатами полевого опыта (урожайность сортов в разных блоках). Как бы вы представили эти данные в виде DataFrame?
8. Манипуляции с данными в Pandas. Основные методы для фильтрации строк и столбцов в DataFrame (использование квадратных скобок, методов .loc[], .iloc[], функций .query()). Покажите, как выбрать из таблицы данные только по определённому сорту растения.
9. Агрегация и группировка в Pandas: Для чего используется метод .groupby()? Напишите код, который рассчитывает среднюю урожайность для каждого сорта растений из таблицы, представленной в виде DataFrame.
10. Визуализация данных с Matplotlib. Модуль pyplot из библиотеки Matplotlib. Продвинутая визуализация:
11. Основы BioPython. BioPython и его основное применение в биоинформатике. Как с помощью BioPython можно парсить файл в формате GenBank и извлекать информацию о названии гена и его последовательности?
12. Отличие методов чтения FASTA и FASTQ файлов в Biopython и их типичные применения.
13. Парсинг FASTQ-файлов. Опишите структуру формата FASTQ.
14. Случайные числа и моделирование. Модуль random. Как можно использовать генерацию случайных чисел для создания симуляции генетического дрейфа в популяции или для бутстрэп-анализа (bootstrap)?
15. Линейная алгебра в SciPy. Какие задачи линейной алгебры позволяет решать модуль scipy.linalg? Приведите гипотетический пример применения (например, решение системы уравнений в кинетической модели ферментативной реакции).
16. Статистический анализ в SciPy. Модуль scipy.stats. Статистические тесты в модуле.
17. Кластерный анализ. Основные понятия?
18. Jupyter Notebook/Lab. Назначение среды Jupyter? Перечислите её ключевые преимущества для исследовательской работы в области биоинформатики (интерактивность, интеграция кода, текста и графиков).
19. Основные типы данных в Python. Приведите примеры использования в задачах биотехнологии.
20. Понятие библиотеки NumPy. Опишите её ключевые преимущества для научных вычислений по сравнению со стандартными списками Python.
21. Инструменты разработки (IDE). Специализированная среда разработки (например, PyCharm) и ее отличие от простого текстового редактора. Функции IDE для написания надёжного научного кода.

**Экзамен по дисциплине
«Б1.0.10 Программирование на языке Python для научных задач»**

Типовая структура экзаменационного билета

по дисциплине «Б1.0.10 Программирование на языке Python для научных задач»
(специальность/направление подготовки - 19.04.01 Биотехнология)

1. Теоретический вопрос 1.
2. Теоретический вопрос 2.
3. Практическое задание на использование среды программирования Python и его библиотек.

Бланк экзаменационного билета

Образец

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Омский государственный аграрный университет имени П.А. Столыпина»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ ____

по дисциплине «Программирование на языке Python для научных задач» для направления
19.04.01 Биотехнология.

1. Основные типы данных в Python. Приведите примеры использования в задачах биотехнологии.
2. Понятие библиотеки NumPy. Опишите её ключевые преимущества для научных вычислений по сравнению со стандартными списками Python.
3. Практическое задание:
Напишите программу на Python, которая считывает из файла data.txt список чисел (по одному числу в строке), представляющих оптическую плотность (OD600) бактериальной культуры в разные моменты времени, и вычисляет среднее значение OD600.

**ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ
ответов на вопросы промежуточного контроля**

Результаты экзамена определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» и объявляются в день экзамена.

Оценку «отлично» выставляют обучающемуся, глубоко и прочно освоившему теоретический и практический материал дисциплины. Ответ должен быть логичным, грамотным. Обучающемуся необходимо показать знание не только основного, но и дополнительного материала, быстро ориентироваться, отвечая на дополнительные вопросы. Обучающийся должен свободно справляться с поставленными задачами, правильно обосновывать принятые решения.

Оценку «хорошо» заслуживает обучающийся, твердо знающий программный материал дисциплины, грамотно и, по существу, излагающий его. Не следует допускать существенных неточностей при ответах на вопросы, необходимо правильно применять теоретические положения при решении практических задач, владеть определенными навыками и приемами их выполнения.

Оценку «удовлетворительно» получает обучающийся, который имеет знания только основного материала, но не усвоил его детали, испытывает затруднения при решении практических задач. В ответах на поставленные вопросы обучающимся допущены неточности, даны недостаточно правильные формулировки, нарушена последовательность в изложении программного материала.

Оценка «неудовлетворительно» говорит о том, что обучающийся не знает значительной части материала по дисциплине, допускает существенные ошибки в ответах, не может решить практические задачи или решает их с затруднениями.

Выставление оценки осуществляется с учетом описания показателей, критериев и шкал оценивания компетенций по дисциплине, представленных в таблице 1.2

10. Информационное и методическое обеспечение учебного процесса по дисциплине

В соответствии с действующими государственными требованиями для реализации учебного процесса по дисциплине обеспечивающей кафедрой разрабатывается и постоянно совершенствуется учебно-методический комплекс (УМКД), соответствующий данной рабочей программе и прилагаемый к ней. При разработке УМКД кафедра руководствуется установленными университетом требованиями к его структуре, содержанию и оформлению. В состав УМКД входят перечисленные ниже и другие источники учебной и учебно-методической информации, средства наглядности.

Электронная версия актуального УМКД, адаптированная для обучающихся, выставляется в электронной информационно-образовательной среде университета.

ПЕРЕЧЕНЬ литературы, рекомендуемой для изучения дисциплины Б1.0.10 Программирование на языке Python для научных задач	
Автор, наименование, выходные данные	Доступ
Борзунов, С. В. Языки программирования. Python: решение сложных задач : учебное пособие для вузов / С. В. Борзунов, С. Д. Кургалин. – 2-е изд., стер. – Санкт-Петербург : Лань, 2026. – 192 с. – ISBN 978-5-507-51226-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL : https://e.lanbook.com/book/508367 . – Режим доступа : для авториз. пользователей	e.lanbook.com
Гегечкори, Е. Т. Программирование на языке Python : учебное пособие / Е. Т. Гегечкори. – Омск : ОмГТУ, 2023. – 172 с. – ISBN 978-5-8149-3617-2. – Текст : электронный // Лань : электронно-библиотечная система. – URL : https://e.lanbook.com/book/421673 . – Режим доступа : для авториз. пользователей.	e.lanbook.com
Демидова, Л. А. Кластерный анализ. Python : учебное пособие / Л. А. Демидова. – Москва : РТУ МИРЭА, 2022. – 103 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL : https://e.lanbook.com/book/240092 . – Режим доступа : для авториз. пользователей	e.lanbook.com
Панов, М. А. Анализ данных с использованием языка программирования Python : учебное пособие / М. А. Панов. – Екатеринбург : УрГЭУ, 2024. – 329 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL : https://e.lanbook.com/book/481577 . – Режим доступа : для авториз. пользователей	e.lanbook.com
Инженерные технологии и системы : научный журнал. – Саранск : ФГБОУ ВПО "МГУ им. Н.П. Огарёва" – ISSN 2658-6525. – Текст : электронный. – URL : https://znanium.ru/catalog/magazines/issues?ref=eca003ec-77e5-11e9-9e8a-90b11c31de4c . – Режим доступа : по подписке	https://znanium.ru
Программные продукты и системы. – Тверь : Научно-исследовательский институт «Центрпрограммсистем», 1988. – . – Выходит 4 раза в год. – ISSN 2311-2735. – Текст : электронный. – URL : https://eivis.ru/browse/publication/64086	https://eivis.ru/

Пример оформления титульного листа расчетно-аналитической работы

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ П. А. СТОЛЫПИНА»
(ФГБОУ ВО ОМСКИЙ ГАУ)

Факультет ТС в АПК

Кафедра математических и естественнонаучных дисциплин

**Расчетно-аналитическая работа
по дисциплине
«Программирование на языке Python для научных задач»**

ОПОП по направлению подготовки 19.04.01 Биотехнология.
Профиль (направленность) «Биоинформатика и анализ данных в
агробiotехнологиях»

**Тема: Прикладная задача профессиональной
деятельности с применением среды программирования
Python**

Номер варианта: 15

Исполнитель: студент группы
... очной формы обучения
ФИО

Проверил: канд. экон. наук, доцент
кафедры ...
ФИО

ОМСК 2027