

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Комарова Светлана Юриевна

Должность: Проректор по образовательной деятельности

Дата подписания: 2020.09.11 10:01:00

Уникальный программный ключ:

43ba42f5deae4140b1f449a708a7810280712377a81a41207c4a4549f2095d7a

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

Экономический факультет

ОПОП по направлению 19.04.01 Биотехнология

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Б1.0.10 Программирование на языке Python для научных задач

Направленность (профиль)

«Биоинформатика и анализ данных в агробiotехнологиях»

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

ВВЕДЕНИЕ

1. Фонд оценочных средств по дисциплине является обязательным обособленным приложением к Рабочей программе.
2. Фонд оценочных средств является составной частью нормативно-методического обеспечения системы оценки качества освоения обучающимися указанной дисциплины.
3. При помощи ФОС осуществляется контроль и управление процессом формирования обучающимися компетенций, из числа предусмотренных ФГОС ВО в качестве результатов освоения дисциплины.
4. Фонд оценочных средств по дисциплине включает в себя: оценочные средства, применяемые для входного контроля; оценочные средства, применяемые в рамках индивидуализации выполнения и контроля фиксированных видов ВАРС; оценочные средства, применяемые для текущего контроля и оценочные средства, применяемые при промежуточной аттестации по итогам изучения дисциплины.
5. Разработчиками фонда оценочных средств по дисциплине являются преподаватели кафедры математических и естественнонаучных дисциплин, обеспечивающей изучение обучающимися дисциплины в университете. Содержательной основой для разработки ФОС послужила Рабочая программа дисциплины.

1. ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ
учебной дисциплины модуля, персональный уровень достижения которых проверяется с использованием представленных в п. 3 оценочных средств

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

ЧАСТЬ 2. ОБЩАЯ СХЕМА ОЦЕНИВАНИЯ ХОДА И РЕЗУЛЬТАТОВ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Общие критерии оценки и реестр применяемых оценочных средств

2.1. Обзорная ведомость-матрица оценивания хода и результатов изучения дисциплины в рамках педагогического контроля

Категория контроля и оценки	Режим контрольно-оценочных мероприятий				
	самооценка	взаимооценка	Оценка со стороны		Комиссионная оценка
			преподавателя	представителя производства	
	1	2	3	4	5
Индивидуализация выполнения*, контроль фиксированных видов ВАРС:					
– расчетно-аналитическая работа		Обсуждение теоретических выводов по результатам РАР	Рецензирование		
Текущий контроль:					
– самостоятельное изучение тем	Вопросы для самоконтроля		Конспект, опрос		
– в рамках лабораторных занятий и подготовки к ним	Вопросы для самоподготовки		Конспект, опрос		
– в рамках общеуниверситетской системы контроля успеваемости			Фронтальный контроль текущей успеваемости по контрольным неделям, установленным в университете		
Промежуточная аттестация* обучающихся по итогам изучения дисциплины			Экзамен		
* данным знаком помечены индивидуализируемые виды учебной работы					

2.2 Общие критерии оценки хода и результатов изучения учебной дисциплины

1. Формальный критерий получения обучающимися положительной оценки по итогам изучения дисциплины:	
1.1. Предусмотренная программа изучения дисциплины обучающимся выполнена полностью до начала процесса промежуточной аттестации	1.2. По каждой из предусмотренных программой видов работ по дисциплине обучающийся успешно отчитался перед преподавателем, демонстрируя при этом должный (не ниже минимально приемлемого) уровень сформированности элементов компетенций
2. Группы неформальных критериев качественной оценки работы обучающегося в рамках изучения дисциплины:	
2.1. Критерии оценки качества хода процесса изучения обучающимся программы дисциплины (текущей успеваемости)	2.2. Критерии оценки качества выполнения конкретных видов ВАРС
2.3. Критерии оценки качественного уровня итоговых результатов изучения дисциплины	2.4. Критерии аттестационной оценки качественного уровня результатов изучения дисциплины

2.3 РЕЕСТР
элементов фонда оценочных средств по учебной дисциплине

Группа оценочных средств	Оценочное средство или его элемент
	Наименование
1. Средства для индивидуализации выполнения, контроля фиксированных видов ВАРС	Расчетно-аналитическая работа
	Критерий и шкала оценки результатов выполнения расчетно-графической работы
2. Средства для текущего контроля	Вопросы для самостоятельного изучения темы
	Общий алгоритм самостоятельного изучения темы
	Критерии оценки самостоятельного изучения темы
	Вопросы для самоподготовки по темам лабораторных занятий
	Критерии оценки самоподготовки по темам лабораторных занятий
3. Средства для промежуточной аттестации по итогам изучения дисциплины	Примерные тестовые вопросы для проведения итогового контроля
	Критерии оценки ответов на тестовые вопросы итогового контроля
	Нормативная база проведения промежуточной аттестации обучающихся по результатам изучения дисциплины
	Основные характеристики промежуточной аттестации обучающихся по итогам изучения дисциплины

2.4. 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 _{опк3} 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 _{ОПК-3} 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-Зопк-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-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 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-3ОПК-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

ЧАСТЬ 3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующих этапы формирования компетенций

Часть 3.1. Типовые контрольные задания, необходимые для оценки знаний, умений, навыков

3.1.1 . Средства для индивидуализации выполнения, контроля фиксированных видов 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, 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.

3.1.2 Средства для текущего контроля

ВОПРОСЫ для самостоятельного изучения 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 (.yml) 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 стэков, 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 информации 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%).

ВОПРОСЫ для самоподготовки к лабораторным занятиям

Общий алгоритм самоподготовки

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.

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.

Шкала и критерии оценивания самоподготовки по темам занятий

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

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

Тестовые задания

Оценочные средства для проведения тестирования (текущего, рубежного, итогового и пр.) в рамках дисциплины представлены в разделе 4.

3.1.3. Средства для промежуточной аттестации по итогам изучения дисциплины

ВОПРОСЫ для подготовки к итоговому контролю

Конечно, вот 25 экзаменационных вопросов по дисциплине «Программирование на языке Python для научных задач», составленных с учётом специфики направления «Биоинформатика и анализ данных в агробиотехнологиях». Вопросы охватывают как основы языка, так и ключевые библиотеки для работы с биологическими данными.

Примерный перечень 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.

ПРИМЕР ЭКЗАМЕНАЦИОННОГО БИЛЕТА

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

Омский государственный аграрный университет имени П.А. Столыпина

Кафедра математических и естественнонаучных дисциплин

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»
(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.

Заведующий кафедрой _____

Утвержден на заседании кафедры _____, протокол № _____
(наименование) (Дата)

ПЛАНОВАЯ 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.

Нормативная база проведения промежуточной аттестации обучающихся по результатам изучения дисциплины:	
1) действующее «Положение о текущем контроле успеваемости, промежуточной аттестации обучающихся по программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры и среднего профессионального образования в ФГБОУ ВО Омский ГАУ»	
Основные характеристики промежуточной аттестации обучающихся по итогам изучения дисциплины	
Цель промежуточной аттестации -	установление уровня достижения каждым обучающимся целей обучения по данной дисциплине
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 проведения экзамена определяется графиком сдачи экзаменов, утверждаемым приказом ректора
Форма экзамена -	смешанный (письменный, устный)
Время проведения экзамена	Время и место проведения экзамена определяется графиком сдачи экзаменов, утверждаемым деканом факультета

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.

4. ОЦЕНОЧНЫЕ СРЕДСТВА сформированности компетенции

ПК-3 Способен разрабатывать алгоритмы и участвовать в разработке программ в сфере своей профессиональной деятельности

ИД-2 Пишет скрипты на Python для обработки и анализа биоинформатических данных; работает с командной строкой для запуска биоинформатических инструментов

Тип заданий: выбор одного варианта правильного ответа из нескольких предложенных / выбор нескольких правильных вариантов из предложенных вариантов ответов

1. Модуль Biopython, который используется для работы с трехмерными структурами белков (PDB)

Bio.Seq
Bio.SeqIO
+Bio.PDB
Bio.Align

2. Подход в Biopython позволяющий конвертировать последовательности между форматами FASTA, GenBank и GFF/GBK

+Bio.Seq
Bio.PDB
Bio.Entrez
Bio.SeqUtils

3. Библиотека, которая является стандартом де-факто для построения графиков в научном программировании на Python

Seaborn
+Matplotlib
Plotly
Bokeh

4. Библиотека Pandas предназначена для ...

решения дифференциальных уравнений
+работы со структурированными табличными данными (DataFrame)
символьных вычислений
параллельных вычислений

5. Использование Jupyter Notebook/JupyterLab в научной среде необходимо для ...

разработки крупных веб-приложений
+интерактивной разработки, визуализации данных и документирования исследований
управления базами данных
компиляции кода

6. Магическая команда %timeit в Jupyter Notebook

показывает текущее время
+измеряет время выполнения небольшой части кода за несколько проходов для получения точного результата
запускает отладчик
сохраняет состояние ядра

Тип заданий: установление правильной последовательности в предложенных вариантах ответов / установление соответствия между элементами в предложенных вариантах ответов

7. Установите соответствие библиотек Pythona и их основное назначение

УКАЖИТЕ СООТВЕТСТВИЕ ДЛЯ КАЖДОГО ЭЛЕМЕНТА ЗАДАНИЯ

NumPy	Проведение вычислений с многомерными массивами и матрицами
Pandas	Работа с табличными данными, структура DataFrame
Matplotlib	Построение статических, интерактивных и анимированных визуализаций
	Решение научных и инженерных задач (оптимизация, интеграция, статистика)

8. Установите соответствие типам графиков в Matplotlib их описанию

УКАЖИТЕ СООТВЕТСТВИЕ ДЛЯ КАЖДОГО ЭЛЕМЕНТА ЗАДАНИЯ

Линейный (plot)	Строит
Гистограмма (hist)	Показывает распределение частотного признака
Точечный/Диаграмма рассеяния (scatter)	Отображает зависимость одной переменной от другой точками на плоскости
	Столбчатая диаграмма используется для сравнения категориальных данных высотой столбцов

9. Установите соответствие объектам библиотеки Pandas их описанию

УКАЖИТЕ СООТВЕТСТВИЕ ДЛЯ КАЖДОГО ЭЛЕМЕНТА ЗАДАНИЯ

Series	Одномерный помеченный массив, похожий на словарь или вектор
DataFrame	Таблица с маркированными строками и столбцами
Panel	Трехмерная структура данных (устаревший тип, заменен MultiIndex)
	График функции или изменение данных во времени линией

Тип заданий: открытого типа (самостоятельный ввод обучающимся правильного ответа в виде термина, краткого определения, цифрового значения) / Практико-ориентированные задания (кейсы)

10. Для решения задачи: Создайте одномерный массив X из 200 точек на отрезке от -5 до 5 написан код

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-5, 5, 200)
```

Укажите номер строки, в которой допущена ошибка.

ОТВЕТ ЗАПИШИТЕ ЦЕЛЫМ ЧИСЛОМ

+2

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 Выбирает и изучает новое программное обеспечение или библиотеки следуя официальной документации и научным публикациям для решения конкретной прикладной задачи профессиональной деятельности

Тип заданий: выбор одного варианта правильного ответа из нескольких предложенных / выбор нескольких правильных вариантов из предложенных вариантов ответов

11. В библиотеке Matplotlib для отображения графика в интерактивном окне (а не внутри Jupyter Notebook) используется команда...

```
%matplotlib inline
plt.show()
plt.plot()
plt.savefig()
```

12. Функция для чтения данных из файла формата Excel в объект DataFrame

```
+pd.read_excel()  
pd.read_csv()  
pd.load_data()  
pd.import_table()
```

13. Функция из NumPy, которая генерирует массив из N равномерно распределенных чисел на отрезке [start, stop)

```
`np.linspace(start, stop, N)`  
`np.random.rand(N)`  
+`np.arange(start, stop, step)`
```

14. Метод для вычисления среднего значения по всем элементам массива A в NumPy

```
+`A.mean()`  
`np.sum(A) / len(A)`  
`np.average(A)`  
`np.random.randint(start, stop, N)`
```

15. Атрибут в Pandas для изменения названий столбцов DataFrame

```
+`.columns`  
`.index`  
`.values`  
`.head()`
```

16. В Jupyter Notebook команда `%timeit` служит для...

+измерения времени выполнения одной строки или блока кода с высокой точностью
остановки выполнения ячейки
запуска отладчика
сохранения состояния ядра

Тип заданий: установление правильной последовательности в предложенных вариантах ответов / установление соответствия между элементами в предложенных вариантах ответов

17. Модули SciPy и решаемые задачи. Установите соответствие между модулем и его назначением.

УКАЖИТЕ СООТВЕТСТВИЕ ДЛЯ КАЖДОГО ЭЛЕМЕНТА ЗАДАНИЯ

scipy.integrate	численное интегрирование и решение ОДУ
scipy.optimize	поиск минимума/максимума функций
scipy.stats	работа со случайными величинами и статистическими распределениями
	измерение времени выполнения выражения

18. Магические команды Jupyter Notebook. Установите соответствие между командой и её действием.

УКАЖИТЕ СООТВЕТСТВИЕ ДЛЯ КАЖДОГО ЭЛЕМЕНТА ЗАДАНИЯ

%timeit	измеряет время выполнения выражения
%debug	запускает встроенный отладчик
%matplotlib inline	включает отображение графиков прямо в ячейке ноутбука
	поиск минимума/максимума функций

19. Атрибуты и методы объекта DataFrame. Установите соответствие между методом и его результатом.

УКАЖИТЕ СООТВЕТСТВИЕ ДЛЯ КАЖДОГО ЭЛЕМЕНТА ЗАДАНИЯ

.shape	кортеж (количество строк, количество столбцов)
.head()	возвращает первые n строк таблицы
.dropna()	удаляет строки или столбцы с пропущенными значениями
	запускает встроенный отладчик

Тип заданий: открытого типа (самостоятельный ввод обучающимся правильного ответа в виде термина, краткого определения, цифрового значения) / Практико-ориентированные задания (кейсы)

20. Для решения задачи в файле *experiment.txt* записаны результаты эксперимента. Прочитайте файл.

Был написан код:

```
import matplotlib.pyplot as plt
times = []
values = []
try:
    with open('experiment.txt', 'r') as file:
        for line in file:
            t, v = map(float, line.strip().split())
            times.append(t)
            values.append(v)
except FileNotFoundError:
    print("Файл experiment.txt не найден!")
```

Укажите номер строки, в которой неверно указан оператор

ОТВЕТ ЗАПИШИТЕ ЦЕЛЫМ ЧИСЛОМ
+8

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ

Низкий уровень сформированности компетенций – 60% и менее правильно выполненных заданий.

Базовый уровень сформированности компетенций – от 61 до 70% правильно выполненных заданий.

Повышенный уровень сформированности компетенций – от 71 до 80% правильно выполненных заданий.

Высокий уровень сформированности компетенций – от 81 до 100% правильно выполненных заданий.