

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»
(АНОО ВО «УНИВЕРСИТЕТ «СИРИУС»)**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

«Программирование»

Уровень образования:	высшее образование – программа специалитета
Специальность:	06.05.01 Биоинженерия и биоинформатика
Направленность (профиль):	Биоинженерия

1. Трудоемкость дисциплины (модуля): 4 з.е.

2. Место дисциплины в учебном плане: дисциплина «Программирование» входит в Блок 1. «Дисциплины (модули)», обязательную часть, раздел «Профессиональная подготовка» и изучается в 5-8 модулях (3-4 семестры).

3. Цель дисциплины (модуля): формирование у студентов теоретических знаний и практических навыков программирования на основных языках программирования, а также освоение базовых приёмов работы в операционной системе Linux и серверной среде.

4. Задачи дисциплины (модуля):

Задачи дисциплины: формирование у студентов практических навыков разработки алгоритмов и программ на языке высокого уровня (Python) для решения прикладных задач в области анализа биологических данных, а также освоение базовых приёмов работы в командной среде Linux.

5. Перечень разделов (тем) дисциплины и их краткое содержание:

Обучение студентов навыкам работы в операционной среде Linux и навыкам программирования на языке Python. Дисциплина необходима для изучения различных аспектов биоинформатики.

Раздел	Содержание
Раздел 1. Linux для биолога: от терминала к кластеру	Введение в устройство Linux и изучение командной строки Linux, базовых операций с файлами и директориями. Работа с удалённым сервером по SSH, передача данных, использование менеджеров сессий (tmux, screen). Фильтрация текстовых файлов с помощью стандартных утилит (grep, cut, sort). Создание и управление окружениями в conda. Основы работы с вычислительным кластером и запуск заданий через SLURM.
Раздел 2. Введение в Python.	Краткая история дисциплины. Настройка окружения, установка языка программирования работа с IDE.
Раздел 3. Введение в основные типы данных.	Будут разобраны: числа, булевы типы данных, строки, списки, множества, словари, кортежи. Особенности этих типов данных и работы с ними. Операции и приоритет операций, условные операторы.
Раздел 4. Основы работы на языке программирования Python.	Работа с файлами, циклы, генераторы списков, кортежей и словарей, основы регулярных выражения. Введение в функциональное программирование. Функции, lambda функции, способы передачи параметров функциям, рекурсивные функции.
Раздел 5. Основы анализа и визуализации данных в Python	Будут разобраны основы работы с такими библиотеками для анализа и визуализации данных как: NumPy, Pandas, Matplotlib, Seaborn, Plotly.

6. Образовательные результаты освоения дисциплины (модуля):

Формируемые компетенции (код компетенции, формулировка)	Планируемые результаты обучения по дисциплине (индикаторы достижения компетенций)
ОПК-2. Способен использовать специализированные знания фундаментальных разделов математики,	ИОПК-2.1 Применяет фундаментальные знания, полученные в области математических и естественных наук, и использует их в

физики, химии и биологии для проведения исследований в области биоинженерии, биоинформатики и смежных дисциплин (модулей)	<p>профессиональной деятельности, понимает смысл физических, химических и биологических моделей, явлений и процессов</p> <p>ИОПК-2.2 Использует и адаптирует существующие методы математики, физики, химии и биологии для решения прикладных задач в области биоинженерии, биоинформатики</p>
ОПК-5. Способен использовать современные методы и технологии программирования, а также инструменты и среды разработки программного обеспечения в профессиональной деятельности в области биоинформатик	<p>ИОПК-5.1 Обучающийся демонстрирует способность разрабатывать и применять скрипты на языке Python для обработки и анализа биологических данных в стандартных биоинформатических форматах (FASTA, GFF, TSV, CSV).</p> <p>ИОПК-5.2 Обучающийся умеет работать в командной строке Linux, использовать инструменты оболочки и интегрировать внешние биоинформатические программы в автоматизированный анализ данных.</p>
ОПК-6. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	<p>ИОПК-6.1 Знает основные алгоритмические конструкции и языки программирования</p> <p>ИОПК-6.2 Создает компьютерные программы с использованием современных систем программирования</p>

7. Оценочные и методические материалы

7.1. Оценочные материалы для организации текущего контроля

Проверочные работы (ПР)

- Форма: тестовая, электронная
- Место и время проведения: во время контактной работы в аудитории, согласно расписанию
- Примеры вопросов:
 - Какие действия выполняют команды ls, cp, mv, touch, rm, echo, cat в командной строке Bash? Приведите примеры использования.
 - Как с помощью Bash создать новый файл, вывести в него текст и затем отобразить его содержимое?
 - Чем отличается установка пакетов через APT от использования менеджера Conda? В каких случаях предпочтительно использовать Conda?
 - Как создать и активировать виртуальное окружение с помощью Conda? Как добавить в него необходимые пакеты?
 - Как с помощью команды find найти все файлы с определённым расширением в заданной директории и её поддиректориях?
 - В чём принципиальные отличия между виртуальной машиной и контейнером? Какие плюсы и минусы у каждого подхода?
 - Что такое Docker-образ и контейнер? Как создать контейнер из образа и как создать образ из Dockerfile?

- Какие IDE существуют для работы с языком программирования Python?
- Какие отличительные особенности языка программирования Python?
- На какие группы можно разделить типы данных в языке Python?
- Какие бывают числовые типы данных?
- Какие логические конструкции можно использовать в Python?
- С какого индекса начинается индексация строк в Python?
- Перечислите основные методы строк
- В чем основное различие между списками и кортежами?
- Что такое распаковка кортежа?
- Перечислите способы пересечения множеств
- Какую структуру имеют словари?
- Перечислите способы записи информации в файл
- В чем отличие функций и функций-генераторов?
- Основные парадигмы объектно-ориентированного программирования
- В чем отличие DataFrame от Series?

Критерии оценки:

Критерий	Зачтено	Не зачтено
Процент правильных ответов	Процент правильных ответов на вопросы более 80%	Процент правильных ответов менее 80%

Домашние работы:

Примерный перечень заданий для контрольных работ:

1. Домашнее задание №1 выдается студентам в одном варианте и состоит из 5 задач. Каждой задаче присвоен свой балл. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - файл со скриптом.

Задача 1. Скрипт на Bash

Написать bash-скрипт, который из файла геномной аннотации (gtf-файл) достаёт “левые” границы белок-кодирующих генов :

1. Создает директорию results
2. Преобразует файл .gff в таблицу аннотации таким образом, что:
 1. остаются только строки со значением gene в 3 колонке.
 2. Остаются только те строки, в 9 колонке которых содержится “gene_type=unprocessed_pseudogene”.
 3. остаются только столбцы: 1 (хромосома), 4 (координата начала), 5 (координата конца), 7, 9.
 4. от столбца 9 остается только название гена
3. Модифицирует полученную таблицу так, что:
 1. Если ген располагается на “+” цепи, то координата старта остаётся неизменной, а координата конца = координате старта + 1 (т. к. Правая граница не включается)
 2. Если ген располагается на “-” цепи, то координата старта = координате конца, а к координате конца необходимо прибавить 1 (т. к. Правая граница не включается)
4. Записывает полученную таблицу в файл result.tsv в директории results

Задача 2. Числа Фибоначчи

Как известно, числа Фибоначчи определяются следующим образом:

$F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$

Вам нужно написать скрипт, который считывает из stdin число N и выводит в stdout число FN.

Задача 3. Требуется реализовать программу, умножающую квадратные матрицы целых чисел используя алгоритм Штрассена (или его модификацию). Скрипт должен читать две матрицы из стандартного потока ввода (для чтения можно использовать функцию `input` или `numpy.loadtxt`).

Первой строкой на вход скрипту подается целое число - размер перемножаемых квадратных матриц, далее следуют $2 * n$ строк описывающих строки матриц. Числа в строках разделены пробелами.

Ваша программа должна выдать результирующую матрицу на стандартный поток вывода (можете и спользовать `print`), числа строки результирующей матрицы должны быть разделены ненулевым количеством пробелов или табуляций, и каждая строка матрицы должна выводиться на новой строке.

Критерии оценки:

1. Корректность выполнения заданий — до 4 баллов.
2. Качество и грамотность написанного кода — до 2 баллов.

7.2. Оценочные материалы для организации промежуточной аттестации

- Форма проведения: письменная (синхронная), в очном формате в зависимости от расписания. Проведение осуществляется в рамках зачетной недели в третьем и четвертом семестре.

Место проведения: учебная аудитория

Пример экзаменационных вопросов:

1. Опишите структуру файловой системы Linux. Какие значения имеют каталоги `/home`, `/etc`, `/bin`, `/var`, `/tmp`, `/root`? Как перейти между каталогами и создавать их с помощью командной строки?
2. Что такое стандартные потоки ввода, вывода и ошибок в Bash? Как перенаправить вывод команды в файл, объединить потоки или передать вывод одной команды как ввод другой?
3. Объясните, что такое переменная окружения PATH. Как её просмотреть и изменить? Как повлияет её изменение на запуск команд?
4. Что такое процессы в Linux? Как просматривать список процессов и их характеристики? Чем отличаются команды `ps`, `top`, `htop`? Как определить, какой процесс занимает больше всего памяти?
5. Что такое Conda? В чём преимущества использования виртуальных окружений Conda по сравнению с системными пакетными менеджерами (например, APT)? Как создать и активировать окружение с нужной версией Python и установить в него пакет?
6. Опишите, как работает процесс создания Docker-образа с помощью Dockerfile. Какие инструкции используются в Dockerfile? Как опубликовать готовый образ на Docker Hub?
7. Какие функция принятия запроса от пользователя существуют в Python? Какие можно использовать для вывода информации в командную строку?
8. Вам дана случайная строка. Верните строку, в которой все символы будут отсортированы в алфавитном порядке.
9. Перечислите все методы списков, которые позволяют изменять их содержимое. Приведите примеры и особенности.
10. Вам даны 2 списка. Задача расшифровать предложение. Задача найти такие символы, которые есть в обоих списках, получить их в том порядке, в котором они должны следовать и перевести их из кодировки ASCII в буквенную кодировку (для этого используйте функцию `chr`).

11. Напишите функцию, которая принимает и сортирует буквы в строке в зависимости от регистра. Также данная функция должна посчитать, сколько гласных и сколько согласных букв в передаваемой ей строке.
12. Напишите программу, которая принимает от пользователя целое число n и возвращает целое число, представляющее количество перетасовок Фаро, необходимых для восстановления исходного порядка колоды из n карт. Предположим, что n — чётное число от 2 до 2000.
13. Опишите все способы получения данных из словарей и приведите примеры. В чём особенность каждого из них?
14. Напишите обработчик исключений, позволяющий избегать аварийного завершения программы при распаковке кортежей.
15. В чём разница между абсолютным и относительным путем до файла? Перечислите методы, которые позволяют получить информацию о путях, каталогах и их содержимом из библиотеки `os`.
16. Создайте `DataFrame` который будет содержать 3 столбца и 4 строки. Название строк и столбцов должны быть заданы, а не получены автогенерацией самого `Pandas`. `DataFrame` должен содержать числа.
17. Какие библиотеки могут быть использованы для визуализации данных в Python? Выберите любую из них и выполните следующие условия: - сгенерируйте 2-мерный список с числами и постройте для этого списка обычный линейный график и подпишите оси.
18. Что такое наследование классов? Чем отличается родительский класс от наследника? Напишите код в качестве примера реализации наследования

7.3. Методические рекомендации

Обучение по дисциплине предполагает изучение курса на аудиторных занятиях (лекции и практические занятия) и в ходе самостоятельной работы студентов. Обучающимся необходимо ознакомиться с содержанием рабочей программы дисциплины, с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, методическими разработками по данной дисциплине.

Обучение по дисциплине проводится последовательно путем чтения лекций с углублением и закреплением полученных знаний в ходе самостоятельной работы с последующим переводом знаний в умения в ходе практических занятий. На лекциях излагаются лишь основные, имеющие принципиальное значение и наиболее трудные для понимания и усвоения вопросы. Теоретические знания, полученные обучающимися на лекциях и при самостоятельном изучении курса по литературным источникам, закрепляются на практических занятиях. Семинарские занятия дисциплины предполагают их проведение в различных формах с целью выявления полученных знаний, умений, навыков и компетенций с проведением контрольных мероприятий. С целью обеспечения успешного обучения обучающийся должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса.

Получение углубленных знаний по изучаемой дисциплине достигается за счет дополнительных часов к аудиторной работе самостоятельной работы студентов. Выделяемые часы целесообразно использовать для знакомства с дополнительной научной литературой по проблематике дисциплины, анализа научных концепций и современных подходов к осмыслению рассматриваемых проблем. К самостоятельному виду работы студентов относится работа в библиотеках, в электронных поисковых системах и т.п. по сбору материалов, необходимых для проведения практических занятий или выполнения конкретных заданий преподавателя по изучаемым темам. Обучающиеся могут установить электронный диалог с преподавателем, выполнять посредством него контрольные задания.