


Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный
университет, НГУ)

Факультет естественных наук


_____ *подпись*
Согласовано
Декан ФЕН
Резников В.А.
«10» октября 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ-1

направление подготовки: 06.03.01 Биология

Форма обучения: очная

Разработчики:

к.б.н., доцент Лашин С.А.

З.С. Мустафин

Зав.каф. информационной биологии
Академик РАН, Колчанов Н.А.

Руководитель программы:

д.б. н., профессор Шестопалова Л..В.

Новосибирск, 2020

Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.....	3
2. Место дисциплины в структуре образовательной программы.....	3
3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося.....	4
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий.....	4
5. Перечень учебной литературы.....	5
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся..	5
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	5
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.....	6
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.....	6
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.....	6
Приложение 1 Аннотация по дисциплине	
Приложение 2 Оценочные средства по дисциплине	

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Результаты освоения образовательной программы (компетенции)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
ОПК-14 Способность и готовность вести дискуссию по социально-значимым проблемам биологии и экологии			- современными методами представления результатов биологических исследований
ПК-1 Способность эксплуатировать современную аппаратуру и оборудование для выполнения научно-исследовательских полевых и лабораторных биологических работ	- способы работы с удаленными серверами	- управлять работой приложений из командной строки	
ПК-8 Способность использовать основные технические средства поиска научно-биологической информации, универсальные пакеты прикладных компьютерных программ, создавать базы экспериментальных биологических данных, работать с биологической информацией в глобальных компьютерных сетях	- основные форматы представления биологических данных (FASTA, Genbank и другие) - основной перечень библиотек для работы с биологическими данными, их функциональность и область применения	- прорабатывать алгоритм решения задачи, выбирать методы решения. - проектировать и реализовывать программы на языке python с использованием сторонних библиотек	- современными методами программирования и некоторыми методами проектирования программных продуктов

2. Место дисциплины в структуре образовательной программы

Дисциплины (практики), изучение которых необходимо для освоения дисциплины **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ-1:**

Дисциплина «**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ-1**» опирается на следующие дисциплины данной ООП:

- Математический анализ;
- Математическая статистика и теория вероятностей.
-

Результаты освоения дисциплины «**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ-1**» используются в следующих дисциплинах данной ООП:

- Многомерный анализ биологических данных;
- При подготовке дипломной работы.

3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Трудоемкость дисциплины – 3з.е. (108 ч)

Форма промежуточной аттестации: 7 семестр – дифференцированный зачет

№	Вид деятельности	Семестр
		7
1	Лекции, ч	16
2	Практические занятия, ч	16
3	Лабораторные занятия, ч	
4	Занятия в контактной форме, ч	34
5	, из них	
5	из них аудиторных занятий, ч	32
6	в электронной форме, ч	-
7	консультаций, час.	-
8	промежуточная аттестация, ч	2
9	Самостоятельная работа, час.	74
10	Всего, ч	108

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

7 семестр
Лекции (16 ч)

Наименование темы и их содержание	Объем, час
Раздел 1: Программирование на python	
Тема 1. Типы данных в python. Условные операторы. Операторы циклов.	2
Тема 2. Пакеты и модули. Модули math, random. Списки, строки, словари. Пользовательские функции.	2
Тема 3. Работа с файлами, командной строкой. Обработка исключений.	2
Тема 4. Основы объектно-ориентированного программирования.	2
Тема 5. Модули numpy, scipy, matplotlib.	2
Тема 6. Модули pandas, seaborn.	2
Тема 7. Работа с сетью, интернет запросы и их обработка. Регулярные выражения.	2
Тема 8. Задачи биоинформатики, biopython.	2

Практические занятия (16 ч)

Содержание практического занятия	Объем, час
Решение задач по теме 1	2
Решение задач по теме 2	2
Решение задач по теме 3	2
Решение задач по теме 4	2
Решение задач по теме 5	2
Проверочная работа	2
Решение задач по теме 6	2
Решение задач по темам 7-8	2

Самостоятельная работа студентов (74 ч)

Перечень занятий на СРС	Объем, час
Подготовка к практическим занятиям.	56
Подготовка к проверочной работе.	4
Подготовка к дифференцированному зачету.	14

5. Перечень учебной литературы

5.1 Основная литература

1. Н.А. Прохоренок, В.А. Дронов, Python 3 Самое необходимое // БХВ-Петербург, 2016, Библиотека НГУ, всего: 1
2. М. Лутц, Изучаем python // Символ, 2009, Библиотека НГУ, всего: 1

5.2 Дополнительная литература

3. Д. Силен, Ф. Мейсман, М. Али, Основы Data Science и Big Data // Питер, 2018, Библиотека НГУ, всего: 1

6. Перечень учебно-методических материалов по самостоятельной работе обучающихся

4. Электронно-лекционный курс <Введение в информационную биологию> (составители Колчанов Н.А., Лашин С.А.) http://kib.nsu.ru/open_upload/Lashin-IT-and-programming-language.pdf
5. Курс «Информационные технологии и языки программирования» (составитель Лашин С.А.) http://kib.nsu.ru/open_upload/UMK_Lashin_IT_and_programming_lang.pdf

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет.

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС, электронную почту.

7.1 Современные профессиональные базы данных:

- Не используются

7.2. Информационные справочные системы

- Не используются

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1 Перечень программного обеспечения
MicrosoftWindows (лицензированное ПО)
Anaconda(свободно распространяемое ПО)
Biopython(свободно распространяемая библиотека)

8.2 Информационные справочные системы
Не используются

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для реализации дисциплины **Информационные технологии и языки программирования-1** используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации;
2. Помещения для самостоятельной работы обучающихся;

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду НГУ.

Для проведения занятий лекционного типа предлагаются следующие наборы демонстрационного оборудования и учебно-наглядных пособий:

- комплект лекций-презентаций по темам дисциплины;

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине

Перечень результатов обучения по дисциплине **Информационные технологии и языки программирования-1** и индикаторов их достижения представлен в виде знаний, умений и владений в разделе 1.

10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль успеваемости:

Текущий контроль успеваемости осуществляется путем контроля выполнения

письменной проверочной работы. Работа проводится во время практического занятия, под контролем преподавателя. Время на выполнение работы: 4 часа. Студенты могут пользоваться материалами лекций, дополнительной литературой и компьютерами.

Форма проведения: решение задач.

Примерный перечень вопросов для подготовки к письменной проверочной работе:

1. Условные операторы. Операторы циклов.
2. Списки, строки, словари. Пользовательские функции.
3. Работа с файлами. Основы ООП.
4. Модуль matplotlib.

Промежуточная аттестация:

Промежуточная аттестация осуществляется в форме дифференцированного зачета в виде решения задач.

Оценка "удовлетворительно" выставляется, если студент решил все задачи, относящиеся к категории простых. Задачи из категории простых проверяют знание и умение работать с базовыми типами данных и структурами языка программирования python;

оценка "хорошо" выставляется, если студент решил все задачи, относящиеся к категории задач средней сложности. Задачи средней сложности проверяют умение решать задачи, состоящие из нескольких модулей, каждый из которых проверяет различные аспекты работы с языком программирования python;

оценка "отлично" выставляется, если студент решил все задачи, относящиеся к категории сложных. Сложные задачи проверяют умение решать задачи, требующие написания собственного математического алгоритма, работы со сторонними библиотеками, анализ документации этих библиотек.

Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине Информационные технологии и языки программирования-1

Таблица 10.1

Код компетенции	Результат обучения по дисциплине	Оценочное средство
ОПК-14	Владение современными методами представления результатов биологических исследований	Дифференцированный зачет
ПК-1	Знание способов работы с удаленными серверами	Дифференцированный зачет
	Умение управлять работой приложений из командной строки	Дифференцированный зачет
ПК-8	Знание основных форматов представления биологических данных (FASTA, Genbank и другие)	Дифференцированный зачет
	Знание основных библиотек для работы с биологическими данными, их функциональности и области применения	Дифференцированный зачет
	Умение прорабатывать алгоритм решения задачи, выбирать методы решения.	Проверочная работа
	Умение проектировать и реализовывать программы на языке python с использованием сторонних библиотек	Проверочная работа
	Владение современными методами программирования и некоторыми методами проектирования программных продуктов	Дифференцированный зачет

Таблица 10.2

Критерии оценивания результатов обучения	Шкала оценивания
<p><u>Дифференцированный зачет:</u> Студент решает все задачи, относящиеся к категории сложных. Способен представить решение и объяснить все его этапы. Сложные задачи проверяют умение решать задачи, требующие написания собственного математического алгоритма, работы со сторонними библиотеками, анализ документации этих библиотек.</p> <p><u>Проверочная работа:</u> Решено 3 из 3 задач.</p>	<i>Отлично</i>
<p><u>Дифференцированный зачет:</u> Студент решает все задачи, относящиеся к категории задач средней сложности. Способен представить решение и объяснить все его этапы. Задачи средней сложности проверяют умение решать задачи, состоящие из нескольких модулей, каждый из которых проверяет различные аспекты работы с языком программирования python.</p> <p><u>Проверочная работа:</u> Решено 2 из 3 задач.</p>	<i>Хорошо</i>
<p><u>Дифференцированный зачет:</u> Студент решает все задачи, относящиеся к категории простых. Задачи из категории простых проверяют знание и умение работать с базовыми типами данных и структурами языка программирования python.</p> <p><u>Проверочная работа:</u> Решена 1 из 3 задач.</p>	<i>Удовлетворительно</i>
<p><u>Дифференцированный зачет:</u> Студент не может решить представленные задачи из категории простых.</p> <p><u>Проверочная работа:</u> Не решено ни одной задачи.</p>	<i>Неудовлетворительно</i>

Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения

Типовые задания на проверочную работу:

№1. Дан файл, состоящий из строк (нуклеотидных последовательностей). Для всех строк сгенерировать scatterplot (функция `pyplot.scatter`), где у каждой точки координата x – длина последовательности, координата y – её GC-состав.

Точки последовательностей с наибольшим и наименьшим GC составом выделить среди остальных (цветом и размером).

№2. Сгенерировать нуклеотидную последовательность длины 100. Определить, какой кодон является наиболее представленным во всех трёх рамках считывания. Проверить, может ли быть сгенерированная последовательность транслирована (т.е. в ней есть фрагмент от старт до стоп кодона в одной из рамок). Вывести в файл в первых трех строчках самый часто встречаемый кодон и его частоту (одна строка на одну рамку считывания), а в четвертой строке текст, сообщающий о том, может ли последовательность быть транслирована.

Типовые задания на дифференцированный зачет:

- Решить следующие задачи:

№1. Создать класс для хранения нуклеотидной последовательности. Класс должен иметь следующее:

1) Поля `text` и `length` для хранения самой последовательности и ее длины соответственно.

2) Конструктор, принимающий на вход последовательность и ее длину и возвращающий объект класса. Необходимо провести проверку, не содержит ли входная последовательность символов, отличных от символов нуклеотидной последовательности. Если содержит – сообщить об этом пользователю и остановить выполнение программы.

3) Метод, производящий трансляцию нуклеотидной последовательности и возвращающий аминокислотную последовательность.

4) Метод, который строит график `pyplot.pie` для нуклеотидов последовательности.

5) Метод, возвращающий последовательность с “-” цепи для нуклеотидной последовательности (т.е. обратную комплементарную последовательность).

Задачу необходимо сделать без применения `biopython`

№2.

1) Из файла `sequence.gb` (карточка `genbank`, предоставляется преподавателем) извлечь все последовательности с минус цепи (помечены в карточке как `gene complement`).

2) Найти среди них последовательность с наибольшим GC составом

3) Определить в ней наиболее часто встречающийся кодон.

Задачу удобно решать с помощью `biopython`.

Оценочные материалы по промежуточной аттестации (приложение 2), предназначенные для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, хранятся на кафедре-разработчике РПД в печатном и электронном виде.