

ФГБОУ ВО Университет биотехнологий

Институт цифровых технологий

Кафедра прикладной биоинформатики



Прикладное программирование на языке Python

Методические указания

по выполнению практических занятий, контрольной

и самостоятельной работы

Новосибирск 2026

УДК 004.4(07)

ББК 16.2,я7

П759

Кафедра прикладной биоинформатики

Составители: к.б.н. А.Ф.Петров, М.П.Меркушкина

Рецензент: к.тех.н., доцент Томилов И.Н.

Прикладное программирование на языке Python: методические указания по выполнению практических занятий, контрольной и самостоятельной работы / Университет биотехнологий, Институт цифровых технологий; составитель: А.Ф.Петров, М.П.Меркушкина – Новосибирск, 2026. – 23 с.

Методические указания предназначены для студентов магистратуры, обучающихся по направлению подготовки 09.04.03 Прикладная информатика, профиль Прикладная биоинформатика, и устанавливают порядок проведения практических занятий, а также самостоятельной подготовки студентов по разделам дисциплины.

Утверждены и рекомендованы к изданию учебно-методическим советом Института цифровых технологий (протокол № 5 от 20.01.2026 г.).

© Университет биотехнологий, 2026.

Содержание

Введение.....	4
1. Общие методические указания по самостоятельному изучению разделов дисциплины.....	5
2. Методические указания по самостоятельному изучению отдельных тем дисциплины и вопросы для проверки знаний.....	6
3. Выполнение контрольной работы	17
4. Промежуточная аттестация.....	22

Введение

На изучение дисциплины «Прикладное программирование на языке Python» запланировано 180 ч, в том числе 54 аудиторных и 126 ч (70%) отведено для самостоятельного освоения. Учебным планом предусмотрены экзамен и контрольная работа.

Основная цель дисциплины – овладение современными методами программирования на языке Python и их применение для решения практических задач в различных предметных областях.

В процессе изучения дисциплины решаются следующие **задачи**:

- изучение базового синтаксиса и основных конструкций языка Python;
- освоение принципов объектно-ориентированного программирования;
- разработка алгоритмов обработки данных;
- работа с файлами и базами данных;
- создание графических интерфейсов пользователя;
- применение Python для научных вычислений и анализа данных;
- основы тестирования и отладки программ.

Требования к уровню освоения учебной дисциплины. В соответствии с требованиями ФГОС ВО дисциплина «Прикладное программирование на языке Python» направлена на формирование следующих компетенций по направлению подготовки 09.04.03 Прикладная информатика (квалификация (степень) магистр):

- способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем (ОПК-5);
- способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач (ОПК-2).

1. Общие методические указания по самостоятельному изучению разделов дисциплины

Самостоятельное изучение разделов дисциплины целесообразно начинать с подбора рекомендуемой учебной и методической литературы. После изучения необходимо провести самопроверку знаний путем ответов на вопросы, предусмотренные по каждой теме.

Рекомендуемая литература

Основная

- 1 Драпак, К. А. Основы программирования на Python: учебное пособие / К. А. Драпак, М. Ю. Ветлицын, А. М. Макаров. – Волгоград: ВолгГТУ, 2024. – 160 с. – ISBN 978-5-9948-4964-4. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/486758>
- 2 Янцев, В. В. Web-программирование на Python: учебное пособие для вузов / В. В. Янцев. – 3-е изд., перераб. – Санкт-Петербург: Лань, 2024. – 180 с. – ISBN 978-5-507-48364-8. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/392993>

Дополнительная

3. Программные системы статистического анализа. Обнаружение закономерностей в данных с использованием системы и языка Python: учебное пособие / В. М. Волкова, М. А. Семёнова, Е. С. Четвертакова, С. С. Вожов. – Новосибирск: НГТУ, 2017. – 74 с. – ISBN 978-5-7782-3183-2. – Текст: электронный// Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/118287>.
4. Тарланов, А. Т. Основы языка программирования Python: учебно-методическое пособие / А. Т. Тарланов, Ш. Г. Магомедов. – Москва: РТУ МИРЭА, 2019. – 107 с. – Текст: электронный// Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/171465>
5. Сидорова, Н. П. Базы данных: практикум по проектированию реляционных баз данных: учебное пособие / Н. П. Сидорова. – Королёв: МГОТУ, 2020. – 92 с. – ISBN 978-5-4499-0799-8. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/149436>

2. Методические указания по самостоятельному изучению отдельных тем дисциплины и вопросы для проверки знаний

Раздел 1. Введение в язык программирования Python

1.1 Введение в язык программирования Python

- знакомство со средой разработки на языке Python;
- знакомство с синтаксисом языка программирования;
- преимущества и недостатки Python;
- справка, документация, ошибки, комментарии;
- сферы применения языка;
- общие правила разработки.

1.2 Парадигмы программирования в языке Python

- функциональное программирование;
- процедурно-ориентированное программирование;
- объектно-ориентированное программирование;
- повторное использование кода;
- преимущества и недостатки выбранной парадигмы;
- принятие решения в выборе направления разработки в зависимости от поставленных задач.

Раздел 2. Обработка экспериментальных данных на языке Python

2.1 Знакомство с типами данных и операциями над ними

- стандартные типы: numbePythonons (числа), stPythonings (строки), lists (списки), dictionaPythonies (словари), tuples (кортежи), sets (множества), boolean (логический тип данных);
- классификация типов по признакам: изменяемые, неизменяемые, упорядоченные, неупорядоченные;
- исключения в Python. Конструкция tPythony - except для обработки исключений;
- приведение типов данных, операции по преобразованию типов;
- арифметические операции над числами;
- строки и принципы их форматирования;
- работа со списками и словарями;
- множества как контейнеры для других объектов.

2.2 Управляющие конструкции языка программирования, функции

- операторы сравнения;
- логические операторы;
- оператор IF, операторы IF...ELIF...ELSE, вложенные конструкции IF
- циклы (while - циклы с условием, foPython - совместный цикл), итераторы, прерывание и продолжение циклов;
- функции и рекурсия.

2.3 Модули, пакеты и файлы

- поиск пакетов и модулей;
- модули и видимость содержимого;
- импортирование модулей в Python;
- использование пакетов в Python;
- создание и доступ собственного пакета;
- файлы в Python, создание-чтение-запись-обновление, текстовые и бинарные файлы.

2.4 Работа с большими данными. Библиотека Pandas

- установка и начало работы с Pandas;
- структуры данных: DataFrame, Series, Panel;
- импорт, экспорт и проверка данных;
- сортировка записей, статистическая сводка, нарезка, фильтрация, группировка данных;
- описательная статистика в Pandas.

2.5 Визуализация в Python

- построение диаграмм рассеивания;
- построение графиков и гистограмм;
- построение столбчатых и прямоугольных диаграмм;
- matplotlib, UML.

Раздел 3. Работа с базами данных в языке Python

3.1 Работа с базами данных в Python. Многопоточность

- Python DB-API модули в зависимости от базы данных;
- соединение с базой, получение курсора;
- чтение, обновление и запись в базу данных;
- оптимизации и защита кода при работе с базой данных;
- основы и принципы написания многопоточных и многопроцессорных приложений на Python.

Текущий контроль успеваемости

Тест к Разделу №1. Введение в язык программирования Python

1. Что выведет этот код?

```
python  
print(2 ** 3 + 5 % 3)
```

Варианты:

- a) 10
- b) 9
- c) 8
- d) Ошибка

Ответ: b) 9 (возведение в степень 8 + остаток от деления 2

2. Какой тип данных неизменяем (immutable)?

Варианты:

- a) Список (list)
- b) Словарь (dict)
- c) Кортеж (tuple)
- d) Множество (set)

3. Что делает оператор // в Python?

Варианты:

- a) Возвращает остаток от деления
- b) Выполняет целочисленное деление
- c) Возводит в степень
- d) Проверяет равенство

4. Как правильно создать пустой словарь?

Варианты:

- a) {}
- b) dict()
- c) []
- d) set()

5. Что выведет код?

```
numbers = [1, 2, 3]
numbers.append([4, 5])
print(len(numbers))
```

Варианты:

- a) 3
- b) 4
- c) 5
- d) Ошибка

6. Какой оператор используется для логического "И"?

Варианты:

- a) &
- b) &&
- c) and
- d) ||

7. Что такое "срез" (slice) в Python?

Варианты:

- a) Метод удаления элементов
- b) Операция извлечения части списка/строки
- c) Инструмент сортировки
- d) Функция для работы с файлами

Ответ: b) Операция извлечения части списка/строки

8. Как обработать исключение?

Варианты:

- a) Использовать if/else
- b) Использовать try/except
- c) Использовать for/while
- d) Использовать assert

9. Что выведет код?

```
text = "hello"  
print(text[1:4])
```

Варианты:

- a) hel
- b) ell
- c) ello
- d) lo

10. Как создать функцию с аргументом по умолчанию?

Варианты:

- a) deffunc(x=10):
- b) deffunc(x: 10):
- c) function func(x=10)
- d) func(x default=10)

11. Что такое PEP 8?

Варианты:

- a) Сборник задач по Python
- b) Руководство по стилю кода
- c) Модуль для работы с данными
- d) Алгоритм сортировки

12. Как получить список ключей словаря my_dict?

Варианты:

- a) my_dict.keys()
- b) my_dict.items()
- c) keys(my_dict)
- d) list(my_dict)

Ответ: a) my_dict.keys()

13. Что такое "итератор"?

Варианты:

- a) Функция для создания циклов
- b) Объект, возвращающий элементы по одному

- c) Метод сортировки коллекций
- d) Тип данных для чисел

14. Какой метод добавляет элемент в конец списка?

Варианты:

- a) insert()
- b) add()
- c) push()
- d) append()

15. Что выведет код?

```
a = 10  
b = a  
b = 20  
print(a)
```

Варианты:

- a) 10
- b) 20
- c) Ошибка
- d) None

Тест к разделу №2 - Обработка экспериментальных данных на языке Python

1. Как объявить массив numpy, если этот модуль импортирован как np?*

- a) array
- б) np.array
- в) list
- г) np.list

2. Какой тип будет у элементов np.array([1, '2', 3.5])?

- a) np.int
- б) np.float
- в) U (unicode)
- г) возникнет ошибка при создании array

3. Какое максимальное число можно хранить в np.array с типом np.int16? (проверьте себя в Colab)

4. Можно ли задать n-мерный массив в numpy?

- 1) да
- 2) нет
- 3) и да и нет

5. Какая команда задаст массив размером 5*5 заполненный единицами?

- a) np.full((5,5), 1)

- б) `np.full((1,1), 5)`
- в) `np.ones((5,5))`
- г) `np.eye(5)`

6. Сколько элементов содержится в `np.zeros`

- 1). 4
- 2). 3
- 3). 12
- 4). 2

7. Какой атрибут позволяет узнать размерность `np.array`?

- а) `size`
- б) `len`
- в) `shape`
- г) `reshape`

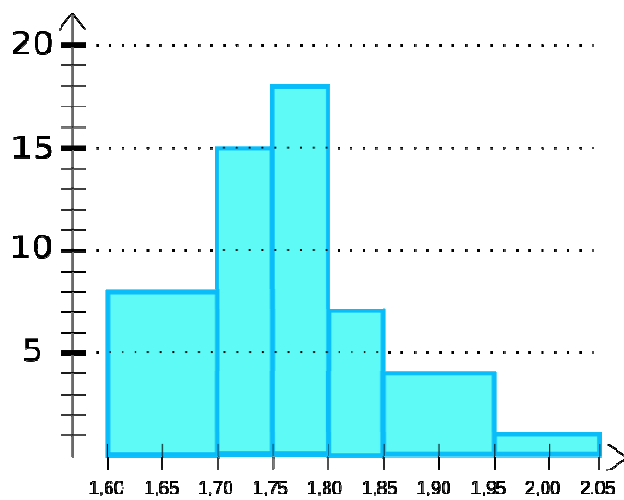
8. Какие из значений приведутся к `False`, если будут использованы в `if/while` условиях?

- а) `-1`
- б) `0`
- в) `0.0`
- г) `""` (пустая строка)
- д) `" "` (строка с пробелом)
- е) `0.1`

9. При помощи программы с циклом `for` вычислите сумму арифметической прогрессии: $1 + 3 + 5 + \dots + 123456789$.

10. Вычислите сумму кубов натуральных чисел от 1 до 100. $1^3 + 2^3 + \dots + 100^3$

11. Какой способ представления данных изображен на рисунке?

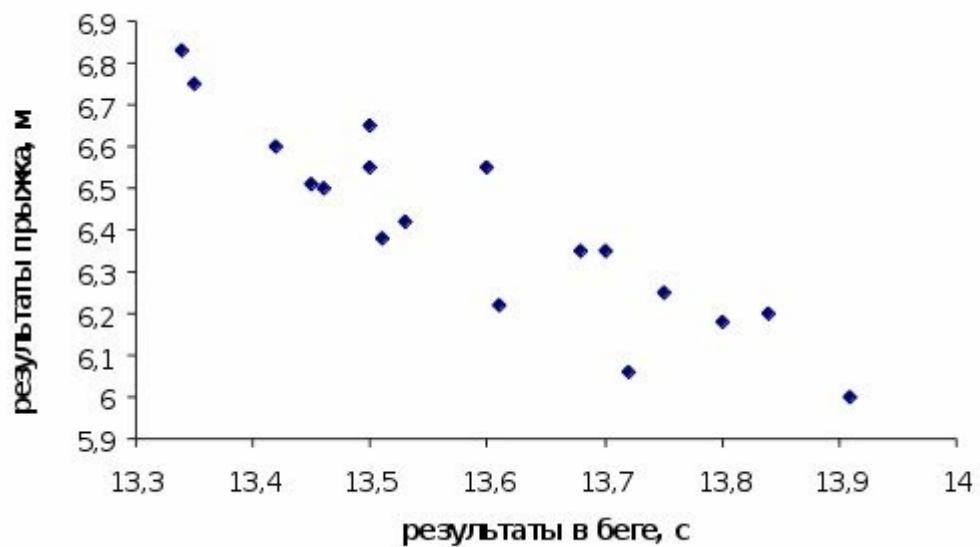


- а) Дендрограмма
- б) График функций
- в) Гистограмма
- г) Круговая диаграмма

12.Какой размерности графики можно создавать в matplotlib?

- а) 2D
- б) 3D
- в) 4D
- г) 5D

13.Какая зависимость наблюдается в данных? *



- 1). Отрицательная корреляция
- 2). Положительная корреляция

- 3). Заметной зависимости не наблюдается
- 4). Параболическая корреляция

14. Нужно ли обязательно указывать свой индекс при создании PandasSeries или DataFrame? *

15. Какие типы данных можно считать pd.read? *

- а) excel
- б) jpeg
- в) csv
- г) json
- д) mp3

16. Что будет, если сложить два массива pd.Series (пример: s1+s2), у которых не полностью совпадают индексы? *

- а) Будет ошибка
- б) Некоторые элементы будут nan
- в) Получится конкатенация массивов
- г) Новый массив pd.Series

17. Какой метод выводит первые 10 строк объекта pd.DataFrame?

- а) .first(10)
- б) .get(10)
- в) .top(10)
- г) .head(10)

18. Какой метод сбрасывает индекс у DataFrame?

- а) .drop_index()
- б) .alter_index()
- в) .reset_index()
- г) .reindex()

19. Горизонтальная ось это (колонки)? *

- а) axis 0
- б) axis 1
- в) axis 2
- г) axis 3

20. Укажите правильные ответы относительно статистик следующего ряда: 2 3 12 7 -4 8 7 2 -1 *

- а) медиана ряда равна 4
- б) среднее значение ряда равно 4
- в) в ряду несколько мод
- г) в ряду одна мода
- д) стандартное отклонение положительное

е) стандартное отклонение отрицательное

21. Как можно вывести строки таблицы с первой по третью включительно? (Нумерация с нуля и индексация таблицы стандартна, т.е. индексы - это целые числа, начинающиеся с нуля)

- а) `df.loc[1:3]`
- б) `df[1:3]`
- в) `df.iloc[1:3]`
- г) `df.iloc[1:4]`

22. Пусть есть таблица `df` с колонками 'speed', 'power' и 'type'. Какой будет вывод от команды `df.plot.scatter(x='speed', y='power')`

- а) Никакого
- б) Непрерывный график
- в) Диаграмма рассеяния
- г) Гистограмма

23. Пусть `df` является объектом класса `pd.DataFrame`. Какой из следующих методов вставит вместо пропущенных значений строку 'unknown' и вернет новую таблицу?

- а) `pd.fillna(df, 'unknown')`
- б) `df.fillna('unknown')`
- в) `df[df == np.nan] = 'unknown'`
- г) `df.fillna('unknown', inplace=True)`

24. Для чего нужны сводные (pivottables) таблицы?

- а) Функция, которая нужна для совместимости pandas и других программ для работы с таблицами
- б) Для компактного представления агрегированных данных
- в) Для более эффективного хранения данных
- г) Для транспонирования таблиц

Тест к разделу №3 Работа с базами данных в языке Python

1. Какой модуль стандартной библиотеки Python используется для работы с SQLite?

- а) `mysql`
- б) `sqlite`
- в) `sqlite3`
- д) `db`

2. Какой метод используется для выполнения SQL-запроса, возвращающего данные?

- а) `commit()`
- б) `executemany()`
- в) `fetch()`

d) execute()

3. Что такое SQL-инъекция?

- a) Способ вставки данных в таблицу
- b) Уязвимость, позволяющая выполнить произвольный SQL-код
- c) Метод оптимизации запросов
- d) Тип соединения с БД

4. Как предотвратить SQL-инъекции при использовании параметров?

Copy

Download

```
cursor.execute("SELECT * FROM users WHERE id = ?", (user_id,))
```

- a) Использовать f-строки
- b) Конкатенировать строки
- c) Передавать параметры через кортеж
- d) Использовать executemany()

5. Для чего используется метод fetchone()?

- a) Получение всех строк результата
- b) Получение первой строки результата
- c) Получение количества строк
- d) Выполнение запроса

6. Что делает этот код?

Copy

Download

```
conn = sqlite3.connect('test.db')  
conn.commit()
```

- a) Открывает транзакцию
- b) Отменяет изменения
- c) Сохраняет изменения в БД
- d) Закрывает соединение

7. Какой пакет чаще всего используется для работы с PostgreSQL?

- a) pymysql
- b) psycopg2
- c) sqlalchemy
- d) pyodbc

8. Что такое ORM?

- a) Язык запросов к БД
- b) Технология преобразования объектов в записи БД
- c) Протокол передачи данных
- d) Система управления транзакциями

9. Какой метод SQLAlchemy используется для создания таблиц?

- a) create_database()
- b) Table.create()
- c) Base.metadata.create_all()
- d) session.make_tables()

10. Как добавить новый объект в сессию SQLAlchemy?

- a) session.insert(obj)
- b) session.new(obj)
- c) session.add(obj)
- d) session.create(obj)

11. Что делает этот код SQLAlchemy?

Copy

Download

```
session.query(User).filter(User.age > 30).all()
```

- a) Обновляет записи
- b) Удаляет записи
- c) Выбирает пользователей старше 30 лет
- d) Создает новых пользователей

12. Для чего используется connection.cursor()?

- a) Для управления транзакциями
- b) Для выполнения запросов и получения результатов
- c) Для закрытия соединения
- d) Для создания БД

13. Какой метод используется для массовой вставки данных?

- a) cursor.insert()
- b) cursor.executemany()
- c) cursor.bulk_insert()
- d) cursor.many()

14. Что такое миграции баз данных?

- a) Перенос БД на другой сервер
- b) Управление изменениями схемы БД
- c) Конвертация типов данных
- d) Резервное копирование

15. Какой инструмент используется для миграций в SQLAlchemy?

- a) SQLAlchemy-Migrate
- b) Django-Migrations
- c) Alembic
- d) Flask-Migrate

Критерии оценки:

- оценка «отлично» выставляется студенту, если тест выполнен на 80 % и выше;
- оценка «хорошо» выставляется студенту, если тест выполнен на 70 %;
- оценка «удовлетворительно» выставляется студенту, если тест выполнен на 60 %;
- оценка «неудовлетворительно» выставляется студенту, если тест выполнен на 50 %.

3. Выполнение контрольной работы

Каждому студенту randomly выдаётся 5 заданий для самостоятельного решения. Задания распределяются по принципу наименьшего совпадения номеров между двумя случайными студентами.

Задания

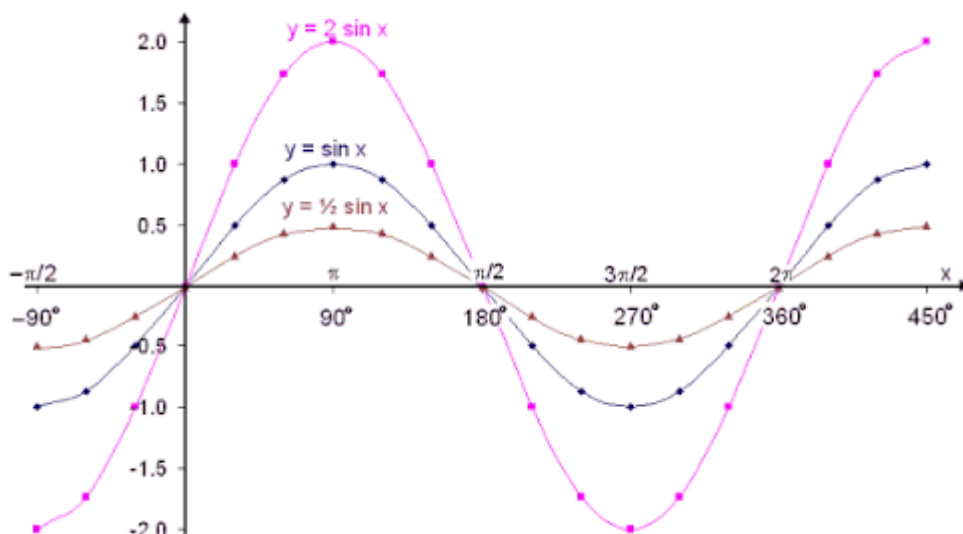
1. Постройте график зависимости stretch от distance

stretch	46	54	48	50	44	42	52
distance	148	182	173	166	109	141	16

2. Постройте гистограмму распределения снежного покрова по годам с использованием графических возможностей Python . Создайте столбец (log.snow.cover) логарифмических преобразований snow.cover.

year	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979
Snow.cover	6.5	12.0	14.9	10.0	10.7	7.9	21.9	12.5	14.5	9.2

3. Нарисуйте как можно более похожую картинку в matplotlib. Добавьте к ней описание осей, сетку, легенду, название. Сохраните картинку из Jupyter или Collab.



4. Постройте график визуализации регрессионной зависимости длины листа от ширины листа с группировкой по видам ирисов.

5. Сгенерируйте последовательность чисел с нормальным распределением при $n = 100$, $\text{mean} = 170$, $\text{sd} = 4$.

6. Сгенерируйте последовательность из 50 чисел от 0 до 1.

7. Постройте диаграмму рассеяния по сгенерированной последовательности чисел с распределением хи-квадрат при степени свободы 1.

8. Используя другой символ и/или другой цвет, постройте графики из двух фреймов данных *elastic1* и *elastic2*.

elastic1

stretch(m m)	46	54	48	50	44	42	52
distance(c m)	183	217	189	208	178	150	249

elastic2.

stretch(mm)	25	45	35	40	55	50	30	50	60
distance(cm)	71	196	127	187	249	217	114	228	291

9. Постройте линейные модели зависимости stretch(mm) от distance(cm) по данным *elastic1* и *elastic2*. Сравните полученные модели.

10. По данным датасета *adults* выведите информацию о немцах старше сорока. Рассчитайте долю тех, чей заработок превышает 50К. Округлите ответ до 2х знаков после запятой. Рассчитайте mean, median, sd, se, Q1, Q3.

11. По данным таблицы (п. 16) изучите зависимость потенциала быков по прибавке к продуктивности по удою (фунт), жиру (%) и белку(%) от года рождения и месяца рождения быка. Требуют ли они нелинейных преобразований?

12. Используя таблицу (п. 16) примените анализ главных компонент к признакам линейной оценки быков.

Stature							+0.80	Tall
Strength							+0.81	Strong
Body Depth							+0.68	Deep
Dairy form							+1.62	Open
Rump Angle							-0.58	High Pins
Thurl Width							+1.30	Wide
R. Legs-S View							-0.15	Posty
R. Legs-R View							+0.63	Straight
Foot Angle							+0.49	Steep
F&L Score							+0.71	High
F. Udder Att.							+2.00	Strong
R. Udder Ht.							+2.53	High
R. Udder Wid.							+3.24	Wide
Udder Cleft							+0.79	Strong
Udder Depth							+0.96	Shallow
F. Teat Place							+1.10	Close
R. Teat Place							+1.24	Close
Teat Length							-0.23	Short

Постройте диаграмму рассеяния в двухмерном пространстве исходя из страны происхождения быка. Опишите расположение США и Канады относительно друг друга.

13. Напишите функцию, возвращающую категории A, B, C, D в зависимости от значения TPI.

14. Сгенерируйте последовательность, состоящую из восьми четверок, затем семи шестерок и, наконец, девяти троек. Сохраните полученные числа в столбцах матрицы 6 на 4.

15. Создайте вектор, состоящий из одной единицы, затем двух двоек, трех троек и т. д. и заканчивающийся девятью девятками.

16. Используя данные каталогов быков AltaGenetics, ST Genetics, Semex и WWS создайте объект класса tibble, включающий столбцы с информацией о международном номере, дате рождения, полной кличке, TPI, прибавке по молоку, жиру, белку и поставщику семени.

17. По данным таблицы (п.16) создайте объект типа data.frame содержащий данные описательной статистики (mean, median, mo, lim, quantile, sqew, kurtosis, se) о геномной оценке производителей, сгруппированные по поставщикам семени.

18. Используя возможности *pandas* создайте фрейм данных

	a	b	c
0	1	2	3
1	4	5	6
2	7	8	9

Преобразуйте таблицу таким образом, чтобы заголовки столбцов: *a*, *b* и *c* стали градациями категориального фактора.

19. Создайте фрейм данных

Категория	Значение
x	0
x	1
x	2
y	0
y	0
y	2
z	0
z	3
z	3

Преобразуйте его в таблицу таким образом, чтобы *x*, *y* и *z* стали заголовками столбцов.

20. Используя табл (п. 16) создайте столбец с записями о годе рождения быка.

21. Используя данные табл (п. 16) создайте объект класса *data.frame*, включающий в себя только строки не содержащие «NA».

22. Определите выбросы по прибавке к молоку, жиру, белку и TPI по таблице из п.16.

23. Напишите функцию, которая вычисляет скользящее среднее 2-го порядка значений в заданном векторе. Применить к данным *iris*.

24. Напишите функцию, которая возвращает значение согласно формуле

$$\frac{\log I_1 - \log I_2}{I_3 - I_4}$$

где I_1 - I_4 соответствующие столбцы таблицы *iris*.

25. Напишите функцию, возвращающую выбросы согласно фильтру Хампеля.

26. Проведите расчеты факториала через рекурсии на произвольном наборе данных, вызовите факториальную функцию, применяя поток.

Критерии оценки:

- оценка «отлично» выставляется студенту, если в контрольной работе правильно выполнено 80 % заданий и выше;
- оценка «хорошо» выставляется студенту, если в контрольной работе правильно выполнено 70 % заданий;
- оценка «удовлетворительно» выставляется студенту, если в контрольной работе правильно выполнено 60 % заданий;
- оценка «неудовлетворительно» выставляется студенту, если в контрольной работе правильно выполнено 50 % заданий%.

4. Промежуточная аттестация

Для аттестации студентов по дисциплине используется традиционная система контроля и оценки успеваемости обучающихся.

Промежуточный контроль проводится с целью установления уровня освоения материала по самостоятельным разделам в виде контрольных работ и выполнения заданий на семинарских занятиях.

Итоговый контроль – оценка уровня освоения дисциплины по окончании её изучения в форме экзамена в устной форме.

Описание шкалы оценивания:

Критерии оценивания устного ответа на экзаменационные вопросы:

«5» (отлично) – дан полный, развёрнутый ответ на поставленный вопрос, показана совокупность осознанных знаний об объекте, доказательно раскрыты основные положения темы; в ответе прослеживается чёткая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Знание об объекте демонстрируется на фоне понимания его в системе данной науки, и междисциплинарных связей. Ответ изложен литературным языком в терминах науки, показана способность быстро реагировать на уточняющие вопросы;

«4» (хорошо) – дан полный, развёрнутый ответ на поставленный вопрос, показано умение выделить существенные и несущественные признаки, причинно-следственные связи. Ответ чётко структурирован, логичен, изложен в терминах науки, Однако допущены незначительные ошибки ли недочёты, исправленные студентом с помощью «наводящих» вопросов;

«3» (удовлетворительно) – дан неполный ответ, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, теорий явлений, в следствии непонимания студентом их существенных и несущественных признаков и связей. В ответе отсутствуют выводы. Умение раскрыть конкретные проявления обобщённых знаний не показано. Речевое оформление требует поправок, коррекций.

«2» (неудовлетворительно) – студент демонстрирует незнание теоретических основ предмета, не умеет выделять аргументированные выводы и приводить примеры, показывает слабое владение монологической речью, не владеет терминологией, проявляет отсутствие логичности и последовательности изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на занятия.

Вопросы для экзамена

1. Особенности и причины популярности Python, его преимущества и недостатки.
2. Особенности работы с объектами типа list.
3. Типы и структуры данных в Python, команды для работы с данными.
4. Технологии импорта данных в среду Python из различных источников.

5. Одномерные данные и одномерный анализ. Преимущества и недостатки. Примеры.
6. Многомерные данные и многомерный анализ. Преимущества и недостатки. Примеры.
7. Векторы. Основные операции, команды и функции для работы с векторами.
8. Создание таблицы в Python, варианты доступа к строкам, столбцам, отдельным значениям.
9. Методы расчёта описательной статистики в Python.
10. Генерация последовательностей с заданными параметрами.
11. Расчёт критериев нормальности (хи-квадрат, Колмогорова-Смирнова, Шапиро-Уилка, Андерсона-Дарлингa).
12. Расчет стандартного отклонения встроенная функция и код на языке Python.
13. Алгоритмы вычисления медианы, первого и третьего квартиля. Готовые решения на Python.
14. Циклы. Выполнение при наступлении условия. Функции, написанные пользователем.
15. Графические возможности среды Python. Круговые диаграммы. Гистограммы. Диаграммы оценки функции плотности. Диаграммы размахов. Точечные диаграммы. Веерные диаграммы.
16. Графическое представление данных в Python. Виды графиков и диаграмм. Столбчатые диаграммы.
17. Диаграмма типа боксплот. Описание, использование, варианты отображения, анализ по диаграмме.
27. Настройка графических параметров. Символы и линии, цвета, текст, размеры диаграмм и полей. Объединение диаграмм. Параметры осей и условных обозначений. Опорные линии. Легенда. Аннотации.
28. Простые диаграммы, составные и диаграммы с группировкой. Диаграммы для средних значений. Спинограммы.
29. Пропущенные данные и выбросы. Способы исключения выбросов и пропущенных данных из анализа.
- Графические возможности Python.
30. Работа с данными типа «Strings» в Python.
31. Работа с числовыми данными в Python.
32. Создание переменных в R. Переименование и перекодировка переменных. Пропущенные значения. Исключение пропущенных значений из анализа. Преобразование типов.
33. Сортировка и объединение наборов данных в Python. Добавление столбцов и строк. Разделение наборов данных на составляющие. Выбор и исключение переменных. Случайные выборки.
34. Понятие пакета. Загрузка и установка пакета. Получение информации о пакете.
35. Библиотека Pandas.
36. Многопоточность в Python.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФГБОУ ВО Университет биотехнологий

Институт цифровых технологий

Кафедра прикладной биоинформатики

Контрольная работа

по дисциплине: «**Прикладное программирование на языке Python**»

Тема: «...»

Выполнил(а): студент(ка) ... курса

группы ...

ФИО

Проверил: степень,

ФИО

Новосибирск 202...

Составители:

Петров Алексей Федорович

Меркушкина Мария Петровна

....

Прикладное программирование на языке Python

Методические указания по выполнению
практических занятий, контрольной и самостоятельной работы

Компьютерная верстка Меркушкина М.П.

ФГБОУ ВО Университет биотехнологий
630039, г. Новосибирск, ул. Добролюбова, 160