

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

---

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Программирование на языке Python**

Уровень образования: высшее образование – программа магистратуры

Направление подготовки: 06.04.01 Биология  
09.04.03 Прикладная информатика

Направленность (профиль): Биоинформатика

## 1. Общая характеристика дисциплины (модуля)

1.1. Цель: формирование у студентов теоретических знаний и практических навыков программирования на основных языках программирования.

1.2. Задачи: научить конструировать конкретные алгоритмы на языке высокого уровня для решения разнообразных биоинформатических задач.

1.3. Общая трудоемкость: 3 з.е.

1.4. Планируемые результаты обучения:

Формируемые компетенции (код компетенции, формулировка)	Планируемые результаты обучения по дисциплине (модулю) (индикаторы достижения компетенций)
ПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных информационных технологий, для решения профессиональных задач в области биоинформатики, биоинженерии, биотехнологии и фарминдустрии	ИПК-2.1. Знает современные алгоритмы, средства разработки и программные средства, а также принципы написания программ на различных языках программирования
	ИПК-2.2. Осуществляет анализ и выбор методов решения профессиональных задач на основе теоретических знаний в области информационных технологий
	ИПК-2.3. Разрабатывает оригинальные алгоритмы и программные средства для решения профессиональных задач в области биоинформатики, биоинженерии, биотехнологии и фарминдустрии
ПК-5. Способен определять необходимые системные и программные средства для разработки и отладки прикладного программного обеспечения в современных специализированных программных комплексах, а также реализовывать в них новые алгоритмы	ИПК-5.1. Знает математические алгоритмы и принципы определения необходимых системных и программных средств для решения профессиональных задач
	ИПК-5.2. Определяет необходимые системные и программные средства для разработки и отладки прикладного программного обеспечения в современных специализированных программных комплексах
	ИПК-5.3. Реализует новые алгоритмы в современных специализированных программных комплексах

## 2. Структура и содержание дисциплины (модуля)

2.1. Объем дисциплины (модуля) и виды учебной деятельности:

Виды учебной деятельности	1 семестр	Всего
<b>Контактная работа обучающихся с преподавателем, всего ч.</b>	76	76
Лекционные занятия, ч.	x	x
Практические (семинарские) занятия, ч.	x	x
Лабораторные занятия, ч.	72	72
Промежуточная аттестация – экзамен, ч	4	4
Промежуточная аттестация – зачет с оценкой, ч	x	x
Промежуточная аттестация – зачет, ч	x	x
<b>Самостоятельная работа обучающихся, всего ч.</b>	32	32
<b>Общая трудоемкость, ч.</b>	108	108
<b>Общая трудоемкость, з.е.</b>	3	3

2.2. Структура дисциплины (модуля) по разделам (темам) и видам учебной деятельности:

Наименования разделов (тем) дисциплины (модуля)	Лекционные занятия, ч	Практические (семинарские) занятия, ч	Лабораторные занятия, ч	Промежуточная аттестация, ч	Самостоятельная работа, ч	Всего, ч	Форма текущего контроля / промежуточной аттестации
Раздел 1. Введение. Краткая история дисциплины. Настройка окружения, установка библиотек.							письменное домашнее задание
Раздел 2. Введение в Python. Основные типы данных: числа, булевы типы данных, строки, списки, множества, словари, кортежи. Особенности типов данных и работы с ними. Операции и приоритет операций, условные операторы.			20		10	30	письменное домашнее задание
Раздел 3. Основы работы на языке программирования Python. Работа с файлами, циклы, регулярные выражения. Введение в функциональное программирование. Функции, lambda функции, функции генераторы, способы передачи параметров функциям.			20		12	32	письменное домашнее задание
Раздел 4. Объектно- ориентированное программирование создание классов наследование, множественное наследование.			12		4	16	письменное домашнее задание
Раздел 5. Математические пакеты:			20		6	26	письменное домашнее

АНОО ВО «Университет «Сириус»	Рабочая программа дисциплины (модуля) «Программирование на языке Python»	Лист 4 Листов 13
-------------------------------	---	---------------------

NumPy, Pandas, Matplotlib, Seaborn.							задание
Промежуточная аттестация			4		4		Экзамен
Итого	х	х	72	4	32	108	

### 2.3. Содержание разделов (тем) дисциплины (модуля):

Наименования разделов (тем) дисциплины (модуля)	Содержание разделов (тем) дисциплины (модуля)
Раздел 1. Введение. Краткая история дисциплины. Настройка окружения, установка библиотек.	<b>Лекции и семинары:</b> Вводная лекция про особенности языка Python. <b>Самостоятельная работа:</b> Установка интерпретатора Python и среды программирования. Изучение документации, упомянутой в лекции.
Раздел 2. Введение в Python. Основные типы данных: числа, булевы типы данных, строки, списки, множества, словари, кортежи. Особенности типов данных и работы с ними. Операции и приоритет операций, условные операторы.	<b>Лекции и семинары:</b> Особенности типов данных числа и логических типов данных. Особенности строк в Python. Особенности списков, кортежей, множеств и словарей в Python. Отличия и сходства между этими типами данных. Логические операторы: if, elif, else. Циклы: for и while. Особенности циклов генераторов. <b>Самостоятельная работа:</b> Практика работы с типами данных, взаимодействие между ними и логическими типами данных. Изучение документации упомянутых в лекции методов и выполнение домашнего задания. Практика работы с логическими операторами. Изучение документации упомянутых в лекции методов. Выполнение домашнего задания.
Раздел 3. Основы работы на языке программирования Python. Работа с файлами, циклы, регулярные выражения. Введение в функциональное программирование. Функции, lambda функции, функции генераторы, способы передачи параметров функциям.	<b>Лекции и семинары:</b> Изучение библиотеки os для работы с файловыми системами, а также функции open и close. Основы работы с файлами. Основы работы с регулярными выражениями. Обзор методов библиотеки re. Функции с переменным числом аргументов; Обычные и именованные аргументы функции; Значения аргументов функции «по умолчанию»; Лямбда-функции; Вложенные функции; Функции высших порядков <b>Самостоятельная работа:</b> Практическое освоение чтением из файла и записью в файл. Изучение документации упомянутых в лекции методов и выполнение домашнего задания. Практика работы с регулярными выражениями. Изучение документации упомянутых в лекции библиотек и методов. Выполнение домашнего задания. Практическая отработка использования функций при написании кода. Изучение документации упомянутых в лекции методов. Выполнение домашнего задания.
Раздел 4. Объектно-ориентированное программирование создание классов наследование, множественное наследование.	<b>Лекции и семинары:</b> Основные концепции ООП: инкапсуляция, наследование и полиморфизм; Структура класса: конструктор, поля и методы; создание классов; Наследование, множественное наследование; Переопределение методов. Иерархическая структура исключений. Обработка ошибок на основе исключений. Основные подходы к тестированию. Использование

АНОО ВО «Университет «Сириус»	Рабочая программа дисциплины (модуля) «Программирование на языке Python»	Лист 5 Листов 13
-------------------------------	---	---------------------

	<p>модулей unittest и pytest для написания тестов.</p> <p><b>Самостоятельная работа:</b> Написание кода с использованием ООП. Применение полученных навыков для отработки создания классов и взаимодействия с ними. Изучение документации упомянутых в лекции методов. Выполнение домашнего задания. Практикум по написанию unit-тестов при помощи модуля pytest. Изучение документации упомянутых в лекции библиотек и методов. Выполнение домашнего задания.</p>
<p>Раздел 5. Математические пакеты: NumPy, Pandas, Matplotlib, Seaborn.</p>	<p><b>Лекции и семинары:</b> Особенности работы с библиотекой numpy, векторы и матрицы. Различия между массивами numpy и массивами Python. Базовые математические функции для работы с числовыми данными в numpy. Особенности работы с библиотекой pandas. Базовые методы работы с таблицами, модификация таблиц. Способы фильтрации и сортировки таблиц. Введение в визуализацию в Python с использованием библиотеки Matplotlib. Основные типы графиков и способы работы с ними. Визуализация данных с помощью Seaborn. Взаимодействие графиков Seaborn и Matplotlib, визуализация сложных и больших объемов данных из Pandas, способы визуализации DataFrame с использованием Seaborn.</p> <p><b>Самостоятельная работа:</b> Работа с матрицами, перемножение векторов, задачи на математические операции с матрицами и векторами, задачи на работы с массивами numpy. Изучение документации упомянутых в лекции методов и выполнение домашнего задания. Самостоятельная работа: Практика работы с таблицами. Изучение документации упомянутых в лекции методов и выполнение домашнего задания. Отработка навыков по построению различных базовых графиков. Изучение документации упомянутых в лекции библиотек и методов. Практическая отработка построения графиков на данных из DataFrame с использованием Seaborn. Изучение документации упомянутых в лекции методов и выполнение домашнего задания.</p>

#### 2.4. Самостоятельная работа

Самостоятельная работа предусматривает: самостоятельное изучение теоретического материала, подготовку к ответам на семинарских занятиях, подготовку к текущему контролю и промежуточной аттестации, выполнение тестовых заданий по пройденным темам курса.

### 3. Текущий контроль и промежуточная аттестация по дисциплине (модулю) Оценочные материалы

3.1. Текущий контроль успеваемости проводится в течение семестра в следующих формах:

Наименования разделов (тем) дисциплины (модуля)	Форма текущего контроля	Оценочные материалы
Раздел 1. Введение. Краткая история дисциплины. Настройка окружения, установка библиотек.	Устный опрос Контрольные работы	Перечень вопросов для опроса, перечень заданий для контрольных работ
Раздел 2. Введение в Python. Основные типы данных: числа, булевы типы данных, строки, списки, множества, словари, кортежи. Особенности типов данных и работы с ними. Операции и приоритет операций, условные операторы.	Устный опрос Контрольные работы	Перечень вопросов для опроса, перечень заданий для контрольных работ
Раздел 3. Основы работы на языке программирования Python. Работа с файлами, циклы, регулярные выражения. Введение в функциональное программирование. Функции, lambda функции, функции генераторы, способы передачи параметров функциям.	Устный опрос Контрольные работы	Перечень вопросов для опроса, перечень заданий для контрольных работ
Раздел 4. Объектно-ориентированное программирование создание классов наследование, множественное наследование.	Устный опрос Контрольные работы	Перечень вопросов для опроса, перечень заданий для контрольных работ
Раздел 5. Математические пакеты: NumPy, Pandas, Matplotlib, Seaborn.	Устный опрос Контрольные работы	Перечень вопросов для опроса, перечень заданий для контрольных работ

3.2. Оценочные материалы для текущего контроля

Примерный перечень вопросов для устного опроса:

1. Какие IDE существуют для работы с языком программирования Python?
2. Какие отличительные особенности языка программирования Python?
3. На какие группы можно разделить типы данных в языке Python?
4. Какие бывают числовые типы данных?
5. Какие логические конструкции можно использовать в Python?

6. С какого индекса начинается индексация строк в Python?
7. Перечислите основные методы строк.
8. В чем основное различие между списками и кортежами?
9. Что такое распаковка кортежа?
10. Перечислите способы пересечения множеств.
11. Какую структуру имеют словари?
12. Перечислите способы записи информации в файл.
13. В чем отличие функций и функций-генераторов?
14. Основные парадигмы объектно-ориентированного программирования.
15. В чем отличие DataFrame от Series?

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. [2 балла] 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 2. [2 балла] OpenCV

Во всех `.cpp` и `.hpp`-файлах в текущей директории (с учетом всех поддиректорий рекурсивно) необходимо сгруппировать все инкюды заголовочных файлов OpenCV (`<opencv2/...>`) в начале файла и вставить до них строку `#pragma warning(push, 0)`, а после — `#pragma warning(pop)`.

Задача 3. [2 балла] BFS

Написать скрипт, который выводит в `stdout` все файлы текущей директории в порядке обхода в ширину.

Файлы, находящиеся на одной глубине, должны выводиться через перевод строки в лексикографическом порядке; уровни глубины должны разделяться пустыми строками, допускается пустая строка в конце вывода.

Задача 4. [2 балла] Украшаем `find`

Написать скрипт, который оформляет вывод команды `find` в древовидную структуру.

Пример входной директории

```
$ find
```

```
./answers
./answers/Tree.hs
./answers/bfs.sh
./hw
./hw/Makefile
./hw/unix_hw2.tex
```

Пример запуска решения

```
$ ./4.sh
.
|-answers
|---Tree.hs
|---bfs.sh
|-hw
|---Makefile
|---unix_hw2.tex
```

Задача 5. [2 балла] Стек

Написать скрипт, реализующий стек (см. пример).

Пример работы скрипта

```
$ ./stack.sh push "Line 1" && echo "OK"
OK
$ ./stack.sh push "Line 2" && echo "OK"
OK
$ ./stack.sh pop && echo "OK"
Line 2
OK
$ ./stack.sh pop && echo "OK"
Line 1
OK
$ ./stack.sh pop || echo "FAIL"
FAIL
```

Ограничения

Если потребуются временные файлы, размещать их можно только в скрытой папке `.au_sw_stack/` в домашней директории текущего пользователя (создавать ее нужно самостоятельно изнутри скрипта).

Ограничения

$\$1 \geq 2$ .

Домашнее задание №2 выдается студентам в одном варианте. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Напишите скрипт, который получает содержимое MBR вашего жесткого диска. После этого по содержимому MBR, скрипт получает количество разделов и выводит это число на экран. Для простоты можно считать, что раздел не задан, если соответствующая ему часть MBR заполнена нулями.

По содержимому MBR выведите на экран информацию о типе присутствующих разделов.

[https://ru.wikipedia.org/wiki/Главная\\_загрузочная\\_запись](https://ru.wikipedia.org/wiki/Главная_загрузочная_запись).

Домашнее задание №3 выдается студентам в одном варианте. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

Создайте метакласс WithFinals, а также опишите декоратор final, который будучи приписан к функции не позволяет переопределять ее в дочерних классах. Т.е. при попытке создать класс с переопределенной функцией, которая у родителя обозначена как final, должно генерироваться исключение. Не забудьте обработать ситуации, когда есть несколько родителей. Подумайте про MRO!

В непонятных ситуациях руководствуйтесь принципом, что допустимая ситуация - та, в которой в наследниках, при вызове метода, вызывается правильная реализация (см. пример г) и е).

```
class A(metaclass = WithFinals):
```

```
    @final
```

```
    def foo(self):
```

```
        pass
```

```
class X(metaclass = WithFinals):
```

```
    @final
```

```
    def foo(self):
```

```
        pass
```

```
-----  
a) class B(A):
```

```
    def foo(self):
```

```
        pass
```

```
-----  
б) class B(A):
```

```
    pass
```

```
class C(B):
```

```
    def foo(self):
```

```
        pass
```

```
-----  
в) class C:
```

```
        def foo(self):
            pass
class B(C, A):
    pass
```

```
-----
г) class C:
    def foo(self):
        pass
class B(A, C):
    pass
```

```
-----
д) class B(A, X):
    pass
```

```
-----
е) class Y:
    def foo(self):
        pass
class X(Y, metaclass = WithFinals):
    @final
    def foo(self):
        pass
class A(X)
```

Домашнее задание №4 выдается студентам в одном варианте. Срок выполнения домашнего задания - 2 недели. Форма представления обучающимися домашнего задания - представленные в письменном виде решения задач.

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 использовать `print`), числа строки результирующей матрицы должны быть разделены ненулевым количеством пробелов или табуляций, и каждая строка матрицы должна выводиться на новой строке.

**Пример ввода:**

```
3
1 0 0
```

0 1 0

0 0 1

2 0 0

0 2 0

0 0 2

Пример вывода:

2 0 0

0 2 0

0 0 2

3.3. Formой промежуточной аттестации является экзамен.

3.4. Оценочные материалы для промежуточной аттестации

Перечень вопросов для подготовки к экзамену:

1. Какие функция принятия запроса от пользователя существуют в Python? Какие можно использовать для вывода информации в командную строку?

2. Вам дана случайная строка. Верните строку, в которой все символы будут отсортированы в алфавитном порядке.

3. Перечислите все методы списков, которые позволяют изменять их содержимое. Приведите примеры и особенности.

4. 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).

5. 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.

6. 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.

7. Опишите все способы получения данных из словарей и приведите примеры. В чём особенность каждого из них?

8. Напишите обработчик исключений, позволяющий избегать аварийного завершения программы при распаковке кортежей.

9. 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.

10. 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.

11. 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.

12. Что такое наследование классов? Чем отличается родительский класс от наследника? Напишите код в качестве примера реализации наследования.

#### **4. Учебно-методическое и информационное обеспечение дисциплины (модуля)**

##### 4.1. Перечень основной литературы:

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с.

##### 4.2. Перечень дополнительной литературы:

1. Операционные системы, среды и оболочки: Учебное пособие / Т.Л. Партыка, И.И. Попов. - 5-е изд., перераб. и доп. - М.: Форум: НИЦ ИНФРА-М, 2013. - 560 с.

2. Python Algorithms: Mastering Basic Algorithms in the Python Language; Second Edition / Magnus Lie Hetland. Apress. 2014

3. Черпаков, И. В. Основы программирования: учебник и практикум для прикладного бакалавриата / И. В. Черпаков. — М.: Издательство Юрайт, 2018. — 219 с.

4. Федоров, Д. Ю. Программирование на языке высокого уровня python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М.: Издательство Юрайт, 2018. — 126 с.

##### 4.3. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю):

В рамках самостоятельной работы обучающиеся используют интернет-ресурсы:

<https://collection.asdlib.org/>

## 5. Материально-техническое и программное обеспечение дисциплины

### 5.1. Материально-техническое обеспечение:

<i>Вид аудитории</i>	<i>Технические средства и оборудование</i>
<i>Учебная аудитория для проведения промежуточной аттестации</i>	Альфа 5.1 - учебная аудитория для проведения учебных занятий, предусмотренных программой магистратуры. Доска магнитно-маркерная поворотная BoardSYS Twist 100x160 ПО-15Ф 1 шт. Флипчарт 70*100 на роликах 1 шт. Стол-кафедра 1 шт. Стол аудиторный 1 шт. Столы-трансформеры Summa GA ученические 25 шт. Стулья на колесах ученические 25 шт. Ноутбук HP 1 шт. Интерактивная панель NexTouch Nextpanel 86” 1 шт. Радиосистема Arthur Forty U-9700C PSC (UHF) в комплекте. Акустическая система Behringer B215D 2 шт. Веб-камера 4К с технологией искусственного интеллекта JazzTel JT-Vintage-4K 1 шт. Комплект электронных презентаций.
<i>Учебная аудитория для проведения практических занятий – Компьютерный класс</i>	Дельта 2.3 – учебная аудитория для проведения практических занятий (компьютерный класс). Доска магнитно-маркерная поворотная BoardSYS Twist 100x160 ПО-15Ф 1 шт. Флипчарт 70*100 на роликах 1 шт. Стол преподавателя аудиторный 1 шт. Столы и стулья ученические 45 шт. Компьютеры Lenovo ThinkCentre M920s SFF в комплекте с мониторами IIYAMA 27” и периферией – 45 шт. Интерактивная панель NexTouch Nextpanel 86” 1 шт. Радиосистема Arthur Forty U-9700C PSC (UHF) в комплекте. Акустическая система Behringer B215D 2 шт. Веб-камера 4К с технологией искусственного интеллекта JazzTel JT-Vintage-4K 1 шт. Комплект электронных презентаций.

5.2. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе российского производства:

- пакет библиотек для Python (Anaconda)
- инструмент для сборки Haskell (Stack), компилятор C++ (clang).