

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
ПОиАИС

наименование кафедры, отвечающей за реализацию дисциплины


Артемов М.А.
подпись, расшифровка подписи

02.04.2024г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.12 Программирование на языке Python

наименование дисциплины в соответствии с учебным планом

- 1. Шифр и наименование направления подготовки:** 02.03.03
Математическое обеспечение и администрирование информационных систем
- 2. Профиль подготовки (при наличии):**
Проектирование и разработка информационных систем
- 3. Квалификация (степень) выпускника:**
бакалавр
- 4. Форма обучения:**
очная
- 5. Кафедра, отвечающая за реализацию дисциплины:**
Программного обеспечения и администрирования информационных систем
- 6. Составители программы:**
Меджидов Р.Г.
- 7. Рекомендована:** НМС факультета ПММ протокол № 5 от 22.03.2024
(наименование рекомендующей структуры, дата, номер протокола)

8. Учебный год: 2025-2026

Семестр(ы): 3

9. Цели и задачи дисциплины

Целями освоения учебной дисциплины являются:

- формирование базовых знаний языка Python,
- формирование представлений о возможностях языка Python и сфере его применения.

Задачи учебной дисциплины:

- ознакомление с синтаксисом и семантикой языка Python,
- изучение некоторых алгоритмов программирования,
- изучение современных программных средств.

10. Место дисциплины в структуре ООП:

Дисциплина входит в вариативную часть программы бакалавриата.

11. Планируемые результаты обучения по дисциплине (знания, умения, навыки), соотнесенные с планируемыми результатами освоения программы (компетенциями):

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-5	Способен проектировать, разрабатывать и верифицировать программное обеспечение информационных систем	ПК-5.1	Демонстрирует знание методов и способов формализации и алгоритмизации поставленных задач, способов анализа требований к ПО, методологии проектирования ПО	Знать: операторы языка Python; ключевые слова языка Python; принципы работы с консолью и файлом; работу с функциями в языке; принципы объектно-ориентированного программирования; понятие декоратора; принципы построения графиков. Уметь: разрабатывать приложения на языке Python. Владеть: навыками работы со средой IDLE Python; навыками программирования на языке Python; навыками тестирования и отладки решений.
		ПК-5.2	Создает программный код с использованием языков программирования, определения и манипулирования данными и оформляет его в соответствии с установленными требованиями, проверяет и отлаживает его	

12. Объем дисциплины в зачетных единицах/час. (в соответствии с учебным планом) — 2 / 72.

Форма промежуточной аттестации (зачет/экзамен) Зачет

13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоемкость	
	Всего	По семестрам
		Семестр 3
Аудиторные занятия	32	32
в том числе:	лекции	
	практические	
	лабораторные	32
Самостоятельная работа	40	40
в том числе: курсовая работа (проект)		
Форма промежуточной аттестации (экзамен – __ час.)		
Итого:	72	72

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК*
3. Лабораторные занятия			
3.1	Введение. Основные типы. Ветвления и циклы. Ввод и вывод. Коллекции.	Знакомство с языком Python. Простейшая программа на Python. Объявление переменных. Целые числа и операции над ними. Арифметические операции, инкремент. Округление. Индексы, нарезание. Строковые методы. Операции над строками. Комментарии. Ветвления. Блоки кода. Циклы for и while. Многочисленные проверки elif. Блок else в циклах. Оператор match-case. Операции сравнения. Логические операторы. Операторы членства и тождественности. Битовые операторы. Метод print. Экранирующие символы. Неформатируемые строки. Выражение форматирования строк. Методы форматирования строк. Полный синтаксис метода input. Списки, кортежи и словари. Операции, выполняемые над ними.	Онлайн-курс «Программирование на языке Python (ПОиАИС)» (https://edu.vsu.ru/course/view.php?id=12107)
3.2	Типизация и память. Особенности языка Python. Файлы. Функции.	Понятия динамической и строгой типизации. Явное приведение типов. Понятие точки входа. Представление объектов Python в памяти машины. Операции присвоения и изменение значений на месте. Присваивание последовательностей, групповое присваивание, дополненные присваивания, присваивание среза, отрицательный индекс, оператор pass, трехместное выражение, запись операций в строку, знак у нуля, кортеж из одного элемента, операторы or и and с небулевыми значениями. Режимы открытия. Чтение и запись в текстовые файлы. Объявление и вызов функций. Передача аргументов и возврат значений. Области видимости.	
3.3	Модульность, модули math и random. Построение графиков с matplotlib. Принцип ООП в Python. Декораторы. Прочие возможности языка.	Определение модуля. Создание пользовательского модуля. Подключение модуля. Работа с встроенными модулями math и random. Работа с модулями matplotlib и numpy. Построение и настройка двумерных графиков. Настройка области рисования. Графики в полярных координатах. Построение геометрических объектов. Трехмерные графики. Принципы объектно-ориентированного программирования. Особенности ООП в языке Python. Исключения. Обработка исключений. Аннотации. Определение декоратора. Создание пользовательского декоратора. Использование декораторов. Работа со встроенными декораторами. Перечисления. Длинная арифметика. Комплексные числа. Списковые включения. Оператор тождественности.	

* заполняется в случае использования онлайн-курса или материалов ЭК, расположенного на платформе «Электронный университет ВГУ», при реализации отдельного раздела дисциплины В других случаях в ячейки ставятся прочерки.

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Введение. Основные типы. Ветвления и циклы. Ввод и вывод. Коллекции.			12	12	24
2	Типизация и память. Особенности языка Python. Файлы. Функции.			10	12	22
3	Модульность, модули math и random. Построение графиков с matplotlib. Принцип ООП в Python. Декораторы. Прочие возможности языка.			10	16	26
	Итого:			32	40	72

14. Методические указания по освоению дисциплины: *(рекомендации по освоению дисциплины: указание наиболее сложных разделов, работа с презентационным материалом, рекомендации по организации самостоятельной работы по дисциплине, по подготовке к текущей аттестации и др.)*

Изучение материала, излагаемого на занятиях. Изучение предлагаемых на электронном курсе материалов. Выполнение практических заданий для самостоятельной работы, выполнение лабораторных работ по дисциплине. Рекомендуется вести конспект. Изучение рекомендованной литературы и методических материалов.

15. Перечень литературы, ресурсов интернет, необходимых для освоения дисциплины *(список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)*

а) основная литература:

№ п/п	Источник
1	Язык программирования Python — Часть 1 / Артемов М.А., Барановский Е.С., Меджидов Р.Г. — Воронеж : Издательский дом ВГУ, 2021 — 93 с.

б) дополнительная литература:

№ п/п	Источник
1	Лутц М. Изучаем Python./ М. Лутц ; пер. с. англ. Ю.Н. Артеменко. — М. ; СПб. : Диалектика, 2019. — Т. 1, 5-е изд. — 833 с.
2	Язык программирования Python 3 для начинающих и чайников – URL: https://pythonworld.ru/ (дата обращения: 01.05.2023).

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
1	www.lib.vsu.ru – ЗНБ ВГУ
2	Онлайн-курс «Программирование на языке Python (ПОиАИС)» — https://edu.vsu.ru/course/view.php?id=12107

* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы, онлайн-курсы, ЭУМК

16. Перечень учебно-методического обеспечения для самостоятельной работы
(учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических работ и др.)

№ п/п	Источник
1	ОС Windows
2	MS Office
3	IDLE — Python 3

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ, электронное обучение (ЭО), смешанное обучение):

При реализации дисциплины используются модульно-рейтинговая и личностно-ориентированные технологии обучения (ориентированные на индивидуальность студента, компьютерные и коммуникационные технологии).

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий, для организации самостоятельной работы обучающихся используется онлайн- курс, размещенный на платформе Электронного университета ВГУ (LMS moodle), а также другие Интернет-ресурсы, приведенные в п.15в.

18. Материально-техническое обеспечение дисциплины:

Лабораторные занятия должны проводиться в специализированной аудитории, оснащенной учебной мебелью и персональными компьютерами с доступом в сеть Интернет (компьютерные классы, студии), мультимедийным оборудованием (мультимедийный проектор, экран, средства звуковоспроизведения), Число рабочих мест в аудитории должно быть таким, чтобы обеспечивалась индивидуальная работа студента на отдельном персональном компьютере.

Для самостоятельной работы необходим компьютер с доступом к сети Интернет.

Программное обеспечение: IDLE — Python 3

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

- контрольная работа,
- лабораторные работы.

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Введение. Основные типы. Ветвления и циклы. Ввод и вывод. Коллекции.	ПК-5	ПК-5.1, ПК-5.2	Лабораторные работы
2.	Типизация и память. Особенности языка Python. Файлы. Функции.	ПК-5	ПК-5.1, ПК-5.2	
3.	Модульность, модули math и random. Построение графиков	ПК-5	ПК-5.1, ПК-5.2	

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
	с matplotlib. Принцип ООП в Python. Декораторы. Прочие возможности языка.			
Промежуточная аттестация форма контроля – зачет				Контрольная работа

19.1. Текущий контроль

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

_____ лабораторные работы _____
(наименование оценочного средства текущего контроля успеваемости)

Перечень заданий, тем рефератов, тем презентаций, докладов, требования к представлению портфолио

Примеры лабораторных работ:

Лабораторная работа 1.

Создать консольное приложение с меню. Задача не должна завершаться после одного действия — вместо этого каждый раз должна возвращаться в меню до выбора пункта «завершить». Осуществить форматированный вывод. Использовать выражение форматирования строк. База ГАИ хранит марку и цвет авто по номерному знаку с помощью словаря. Она поддерживает операции: добавления и удаления автомобиля, вывода данных по «номерам», вывод всего словаря (в столбик записи вида «номер-марка-цвет»).

Лабораторная работа 2.

Задача должна быть выполнена на языке Python с использованием библиотек matplotlib, numpy, math и, при необходимости, random. Условие задачи и автор должны быть указаны в комментариях. Название окна должно быть осмысленным и не должно оставаться по умолчанию. Область рисования должна быть настроена так, чтобы графики отображались корректно.

$$1) r = \sin \frac{111\varphi}{122}$$

2) Формулы заданы параметрически:

$$x = a \cdot \cos(t) \quad y = a \cdot \sin(2 \cdot t) / 2$$

и

$$x = b \cdot \cos(t) \quad y = b \cdot \sin(2 \cdot t) / 2$$

где a и b — случайные вещественные числа из отрезка $[1, 5]$

Лабораторная работа 3.

Создать консольное приложение с меню, которое будет демонстрировать все действия, которые можно выполнить над указанным классом. Задача не должна завершаться после одного действия — вместо этого каждый раз должна возвращаться в меню до выбора пункта «завершить». Пользователь (преподаватель) должен понимать что ему нужно делать из консоли, а не из условия задачи, кода программы или ваших устных подсказок. Во всех задачах должна быть предусмотрена возможность ввода и вывода из файла и с консоли. Задачи должны быть разбиты на функции. Классы вынести в отдельный модуль. Поля сделать закрытыми и с геттерами; сеттеры дописать по мере необходимости. У класса должны быть определены инициализатор и метод приведения к строке.

Класс «Truck» с полями «марка», «мощность двигателя», «коробка передач» (механика/автомат), «число осей». Декоратор приводит строку-результат функции к верхнему регистру. Декорировать все методы и функции, возвращающие строки. Добавить логический метод, который проверяет указанной ли марки тягач.

Требования к выполнению заданий (или шкалы и критерии оценивания)

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Лабораторная работа выполнена корректно, не содержит необработанных исключений, поставленная цель достигнута, алгоритм решения оптимальный. Свободно отвечает на все вопросы; в случае незнания небольшой части материала способен выстроить собственную логическую цепочку рассуждений и получить ответ.	базовый	Принято
Лабораторная работа содержит ошибки, имеются ситуации, приводящие к падению задачи, выбран неподходящий алгоритм решения, цель задачи не достигнута или достигнута частично. Не отвечает или отвечает не на все вопросы.	—	Не принято

19.2. Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

_____ контрольная работа _____
(наименование оценочного средства промежуточной аттестации)

Перечень практических заданий, тем презентаций, докладов, требования к представлению портфолио, вопросов к экзамену (зачету) и порядок формирования КИМ

Контрольная работа

Вариант 1

- Написать код построения графика $x^2 - y^2 = 1$ пунктирной линией.
- Что делает данный декоратор


```
def decorator(func):
    def wrapper(x, y):
        x, y = y, x
        return func(y, x)
    return wrapper
```
- Вставить пропущенное слово «Если использовать название модуля неудобно, то для него можно создать _____, с помощью ключевого слова as».
- Чем отличается статическое поле от нестатического (логически и в коде)
- Какие из определений «декоратор в Python» не верны (выбрать одно или несколько):
 - Обертки вокруг функций и классов, которые изменяют работу того, к чему они применяются.
 - Функции, которые позволяют динамически добавлять объектам новую функциональность.
 - Функции, которые принимают другую функцию в качестве аргумента, добавляют к ней новую функциональность и возвращают функцию с измененным поведением.
 - Функции, которые в качестве параметра получают функцию и в качестве результата также возвращают функцию, позволяя модифицировать выполняемую функцию, значения ее параметров и ее результат без изменения исходного кода этой функции.

Вариант 2

- Написать код построения графиков 2 функций: $y = (x^2 - 10)/x$ и $y = (10 - x^2)/x$ в одной координатной плоскости.
- Что делает данный декоратор


```
def decorator(func):
```

```
def wrapper(x):  
    x *= 10  
    return func(x)  
return func
```

3. Вставить пропущенное слово «Подключить отдельные _____ модуля можно с помощью инструкции from».
4. Чем отличается абстрактный класс от неабстрактного (логически и в коде)
5. Не используя декораторы, в Python можно работать с (выбрать одно или несколько):
 - a. Статическими полями.
 - b. Статическими методами.
 - c. Геттерами.
 - d. Сеттерами.

Требования к выполнению заданий (или шкалы и критерии оценивания)

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Представлены верные ответы на все вопросы.	базовый	Принято
Не на все вопросы дан правильный ответ или ответ на вопрос отсутствует.	—	Не принято