


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

УТВЕРЖДАЮ

Заведующий кафедрой
Программного обеспечения и администрирования
информационных систем


Артемов М.А.
29.05.2023г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.12 Программирование на языке Python

1. Код и наименование направления подготовки/специальности: 02.04.03
Математическое обеспечение и администрирование информационных систем
2. Профиль подготовки/специализация: Проектирование и разработка
информационных систем
3. Квалификация выпускника: бакалавр
4. Форма обучения: очная
5. Кафедра, отвечающая за реализацию дисциплины: ПОиАИС
6. Составители программы: Меджидов Руслан Гусенович
7. Рекомендована: НМС факультета ПММ, протокол №7 от 26.05.2023

8. Учебный год: 2023-2024

Семестр(ы)/Триместр(ы): 3

9. Цели и задачи учебной дисциплины

Целями освоения учебной дисциплины являются:

- формирование базовых знаний языка Python,
- формирование культуры разработки программных продуктов на современных языках программирования высокого уровня.

Задачи учебной дисциплины:

- ознакомление с синтаксисом и семантикой языка Python,
- изучение некоторых алгоритмов программирования,
- формирование представления о языках высокого уровня.

10. Место учебной дисциплины в структуре ООП: Дисциплина входит в вариативную часть программы бакалавриата.

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-5	Способен проектировать, разрабатывать и верифицировать программное обеспечение информационных систем	ПК-5.1 ПК-5.2	Демонстрирует знание методов и способов формализации и алгоритмизации поставленных задач, способов анализа требований к ПО, методологии проектирования ПО; Создает программный код с использованием языков программирования, определения и манипулирования данными и оформляет его в соответствии с установленными требованиями, проверяет и отлаживает его	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.

12. Объем дисциплины в зачетных единицах/час. 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	Знакомство с языком Python.	Простейшая программа на Python. Объявление переменных. Целые числа и операции над ними.	Онлайн-курс «Программирование на языке Python (ПОиАИС)» (https://edu.vsu.ru/course/view.php?id=12107)
3.2	Точка входа, арифметика.	Понятие точки входа. Точка входа на языке Python. Версии языка. Арифметические операции, инкремент. Типы операндов	
3.3	Ветвления и циклы.	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.	
3.4	Типизация, строки, приведение типов.	Понятия динамической и строгой типизации. Операции над строками. Явное приведение типов.	
3.5	Нарезание, методы для работы с последовательностями.	Комментарии в Python. Понятие нарезания. Строковые методы.	
3.6	Форматированный вывод.	Метод print. Экранирующие символы. Неформатируемые строки. Выражение форматирования строк. Методы форматирования строк. Полный синтаксис метода input.	
3.7	Коллекции.	Списки, кортежи и словари. Операции, выполняемые над ними.	
3.8	Особенности языка Python.	Присваивание последовательностей, групповое присваивание, дополненное присваивание, 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.	
3.9	Файлы.	Режимы открытия. Чтение и запись в текстовые файлы.	
3.10	Функции.	Объявление и вызов функций. Передача аргументов и возврат значений. Области видимости.	
3.11	Представление в памяти машины.	Представление объектов Python в памяти машины. Операции присвоения и изменение значений на месте.	
3.12	Интерпретатор Python, PVM.	Понятия компилируемых и интерпретируемых языков. Виртуальная машина Python. Сборка мусора.	
3.13	Модульность, модули math и random	Определение модуля. Создание пользовательского модуля. Подключение модуля. Работа с встроенными модулями math и random.	
3.14	Построение графиков с matplotlib	Работа с модулями matplotlib и numpy. Построение и настройка двумерных графиков. Настройка области рисования. Графики в полярных координатах. Построение геометрических объектов. Трехмерные графики.	
3.15	Принцип ООП в Python.	Принципы объектно-ориентированного программирования. Особенности ООП в языке Python.	
3.16	Декораторы	Определение декоратора. Создание пользовательского декоратора. Использование декораторов. Работа со встроенными декораторами.	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Знакомство с языком Python.			2	1	3
2	Точка входа, арифметика.			2	1	3
3	Ветвления и циклы.			2	1	3
4	Типизация, строки, приведение типов.			2	1	3
5	Нарезание, методы для работы с последовательностями.			2	2	4
6	Форматированный вывод.			2	4	6
7	Коллекции.			2	2	4
8	Особенности языка Python.			2	2	4
9	Файлы.			2	2	4
10	Функции.			2	4	6
11	Представление в памяти машины.			2	2	4
12	Интерпретатор Python, PVM.			2	2	4
13	Модульность, модули math и random			2	4	6
14	Построение графиков с matplotlib			2	4	6
15	Принцип ООП в Python.			2	4	6
16	Декораторы			2	4	6
	Итого:			32	40	72

14. Методические указания для обучающихся по освоению дисциплины: Изучить предлагаемые на электронном курсе презентационные материалы, рекомендуется вести конспект. Изучить методические пособия, прикрепленные к курсу.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

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
2	Лутц М. Изучаем Python./ М. Лутц ; пер. с. англ. Ю.Н. Артеменко. — М. ; СПб. : Диалектика, 2019. — Т. 1, 5-е изд. — 833 с.
3	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
4	www.lib.vsu.ru – ЗНБ ВГУ
5	Онлайн-курс «Программирование на языке Python (ПОиАИС)» — https://edu.vsu.ru/course/view.php?id=12107

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ 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.	Знакомство с языком Python.	ПК-5	ПК-5.2	Лабораторные работы
2.	Точка входа, арифметика.	ПК-5	ПК-5.2	
3.	Ветвления и циклы.	ПК-5	ПК-5.2	
4.	Типизация, строки, приведение типов.	ПК-5	ПК-5.2	
5.	Нарезание, методы для работы с последовательностям и.	ПК-5	ПК-5.1, ПК-5.2	
6.	Форматированный вывод.	ПК-5	ПК-5.1, ПК-5.2	
7.	Коллекции.	ПК-5	ПК-5.1, ПК-5.2	
8.	Особенности языка Python.	ПК-5	ПК-5.2	

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства	
9.	Файлы.	ПК-5	ПК-5.2		
10.	Функции.	ПК-5	ПК-5.1, ПК-5.2		
11.	Представление в памяти машины.	ПК-5	ПК-5.1		
12.	Интерпретатор Python, PVM.	ПК-5	ПК-5.2		
13.	Модульность, модули math и random	ПК-5	ПК-5.2		
14.	Построение графиков с matplotlib	ПК-5	ПК-5.1, ПК-5.2		
15.	Принцип ООП в Python.	ПК-5	ПК-5.1, ПК-5.2		
16.	Декораторы	ПК-5	ПК-5.1, ПК-5.2		
Промежуточная аттестация форма контроля – зачет					Контрольная работа

*В графе «ФОС» в обязательном порядке перечисляются оценочные средства текущей и промежуточной аттестаций.

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в формах: устного опроса; защиты лабораторных работ, выполнения контрольных работ.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. Промежуточная аттестация по итогам освоения дисциплины проводится в форме зачета.

Для получения положительной итоговой оценки необходимо выполнение всех лабораторных и контрольных работ.

20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью лабораторных и контрольных работ.

Примеры лабораторных работ:

Лабораторная работа 1.

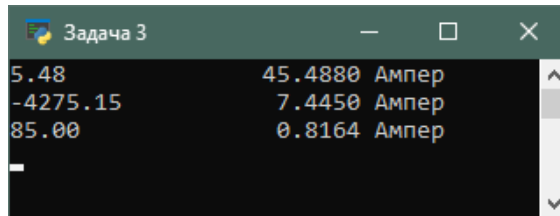
При заданных в коде значениях a , b и c , решить уравнение $\frac{x-a}{x-b} = c$. Вывести на консоль точный результат и результат, округленный до целого значения.

Лабораторная работа 2.

В коде дана строка s . Создать строку, в которой первая и последняя трети берутся из исходной строки, а вторая треть заменятся на соответствующее число символов * (звездочка). Вывести исходную строку и строку-результат.

Лабораторная работа 3.

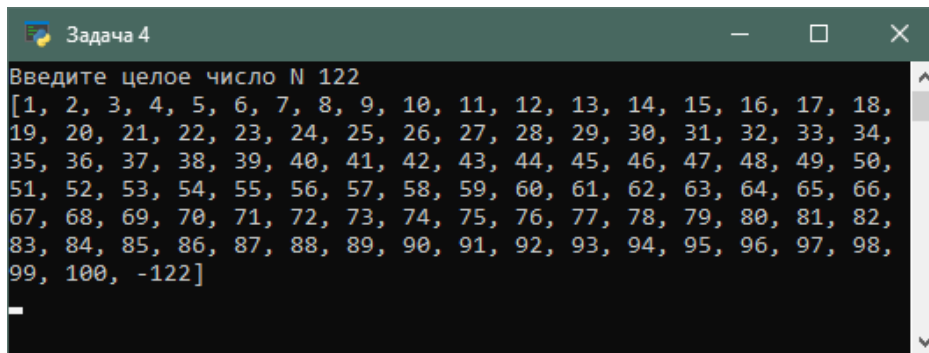
В коде даны 6 чисел с плавающей точкой. Создать строку формата, в которой эти числа будут располагаться в 2 столбца: в первом у всех чисел 2 знака после запятой и выравнивание по левому краю, во втором — 4 знака после запятой, выравнивание вправо и указана единица измерения «Ампер». Вывести данную строку.



```
Задача 3
5.48          45.4880 Ампер
-4275.15      7.4450 Ампер
85.00         0.8164 Ампер
```

Лабораторная работа 4.

Пользователь вводит целое число n . Создать список из 100 элементов. Первые n значений — натуральные числа от 1 до n . Остальные элементы списка равны нулю. После этого вставить в конец списка элемент, равный $-n$. Вывести полученный список.



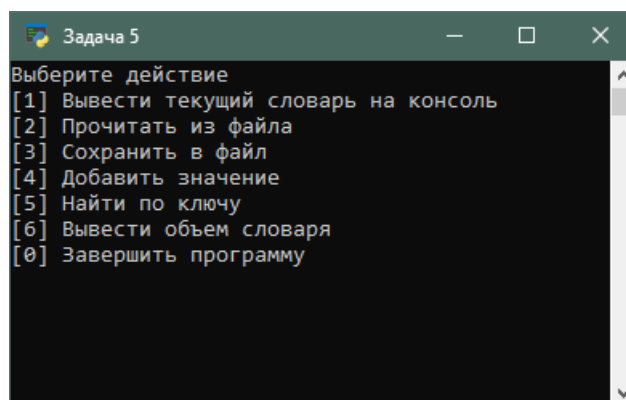
```
Задача 4
Введите целое число N 122
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66,
67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,
99, 100, -122]
```

Лабораторная работа 5.

Написать консольное приложение с меню, позволяющее:

- считать словарь из текстового файла;
- напечатать словарь на консоль;
- сохранить словарь в текстовый файл;
- добавить значение в словарь;
- поиск по ключу;
- вывести объем словаря.

Реализовать алфавитный указатель, хранящий номер страницы по ключевому слову.



```
Задача 5
Выберите действие
[1] Вывести текущий словарь на консоль
[2] Прочитать из файла
[3] Сохранить в файл
[4] Добавить значение
[5] Найти по ключу
[6] Вывести объем словаря
[0] Завершить программу
```

Лабораторная работа 6.

Написать требуемые функции и консольное приложение с меню, демонстрирующее их работу. Ввод целого числа из диапазона $[a; b]$, где a и b — параметры функции. Введенное число — ее результат.

Лабораторная работа 7.

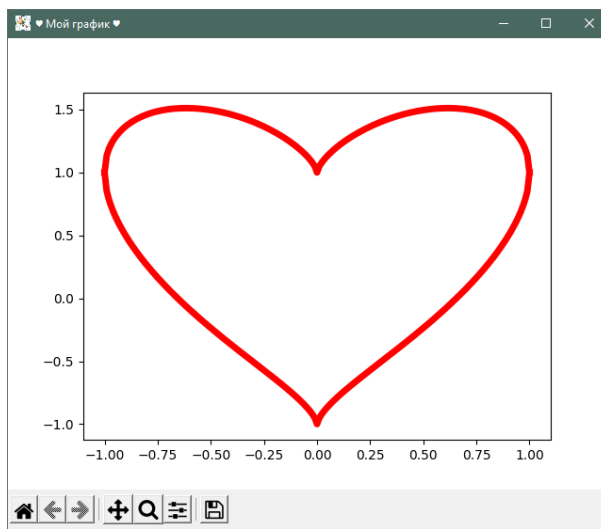
Написать консольное приложение с меню, позволяющее:

- считать словарь из текстового файла;
- напечатать словарь на консоль;
- сохранить словарь в текстовый файл;
- добавить значение в словарь;
- поиск по ключу.

Решение должно быть разбито на функции. Реализовать телефонную книгу, хранящую ФИО по номеру телефона. Добавить возможность удалять запись из словаря.

Лабораторная работа 8.

Построить график функции $y = \sqrt[3]{(x^2) \pm \sqrt{(1-x^2)}}$.



Лабораторная работа 9.

Создать класс «Артефакт» с полями «название», «эффект» и «сила эффекта». Поля должны быть закрытыми. Все поля имеют геттеры, «сила эффекта» еще и сеттер. Также определить методы `__init__` и `__str__`. Создать логирующий декоратор для функций от одного аргумента, который записывает в файл Log.txt строку вида «название функции (параметры) -> результат». Декорировать им все геттеры. Разработать консольное приложение с меню, позволяющее работать с артефактом: создать новый, вывести каждое поле, задать силу эффекта, вывести всю информацию об артефакте.

Контрольная работа

Вариант 1

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


```
def decorator(func):
    def wrapper(x, y):
        x, y = y, x
        return func(y, x)
    return wrapper
```
3. Вставить пропущенное слово «Если использовать название модуля неудобно, то для него можно создать _____, с помощью ключевого слова `as`».
4. Чем отличается статическое поле от нестатического (логически и в коде)
5. Какие из определений «декоратор в Python» не верны (выбрать одно или несколько):
 - а. Обертки вокруг функций и классов, которые изменяют работу того, к чему они применяются.

- b. Функции, которые позволяют динамически добавлять объектам новую функциональность.
- c. Функции, которые принимают другую функцию в качестве аргумента, добавляют к ней новую функциональность и возвращают функцию с измененным поведением.
- d. 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

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

Вариант 3

1. Написать код построения графиков 3 функций: $x^2 + y^2 = 1$, $y = x$ и $y = -x$ в одной координатной плоскости.
2. Что делает данный декоратор


```
def decorator(func):
    def wrapper(x, y):
        x, y = y, x
        return func(x, y)
    return wrapper
```
3. Вставить пропущенное слово «Метод sample используется, когда требуется выбрать несколько элементов из заданной последовательности. Метод возвращает список _____ элементов, которые были выбраны из указанной в параметрах последовательности».
4. Чем отличается статический метод от нестатического (логически и в коде)
5. Утверждение, не справедливое по отношению к статическим полям в Python (выбрать одно или несколько):
 - a. Статическое поле относится ко всему классу.
 - b. Статическое поле объявляются в виде переменных уровня класса.
 - c. Экземпляры класса не могут менять статические поля.
 - d. Используются, когда надо определить общие данные для всех объектов.

Вариант 4

1. Написать код построения графика $y = \pm\sqrt{1.25 - x^2 + 2x}$ с толщиной линии 4 коричневого цвета.

2. Что делает данный декоратор


```
def decorator(func):
    def wrapper(x):
        return 42 * func(x)
    return func
```
3. Вставить пропущенное слово «Все объекты в Python инкапсулируют внутри себя данные и методы работы с ними, предоставляя публичные _____ для взаимодействия».
4. Что из себя представляет декоратор в языке Python (логически и в коде)
5. Для функций в Python не справедливо (выбрать одно или несколько):
 - a. Функции являются объектами.
 - b. Функции можно перегружать.
 - c. Функции можно присваивать переменным.
 - d. Функция может быть результатом функции.

Вариант 5

1. Написать код построения графика $y = \arccos(5x)$ с толщиной линии 5 синего цвета.
2. Что делает данный декоратор


```
def decorator(func):
    import random as r
    def wrapper(x):
        return func(r.randint(1,x))
    return wrapper
```
3. Вставить пропущенное слово «Атрибут может быть условно объявлен _____ с помощью нижнего подчеркивания перед именем, но настоящего скрытия на самом деле не происходит – все на уровне соглашений».
4. Как используется ключевое слово super в Python? Дать развернутый ответ.
5. Для функций в Python не справедливо (выбрать одно или несколько):
 - a. Функции можно объявлять внутри других функций.
 - b. Функции можно объявлять внутри ветки if.
 - c. Локальную функцию можно вернуть как результат.
 - d. Нельзя переопределять операторы для пользовательских типов.

Вариант 6

1. Написать код построения графика $y = \frac{x^2+6}{x^2-3x+1.25}$ с толщиной линии 6 зеленого цвета.
2. Что делает данный декоратор


```
def decorator(func):
    import random as r
    def wrapper(x):
        return func(r.randint(1,x))
    return func
```
3. Вставить пропущенное слово «Декораторы используются, чтобы изменять работу существующих функций или _____, добавлять новые возможности и обеспечивать безопасность».
4. Может ли класс-наследник не реализовывать абстрактный метод родителя? Дать развернутый ответ.
5. Для ООП Python не справедливо (выбрать одно или несколько):
 - a. Все методы в языке изначально виртуальные.
 - b. В методах можно получить и доступ к методам класса-предка.

- c. Методы обязаны использовать слово self.
- d. Язык поддерживает множественное наследование.

20.2. Промежуточная аттестация

Для оценивания результатов обучения на зачете используются следующие показатели:

- 1) теоретические знания;
- 2) практические навыки, включающие выполнение контрольных и лабораторных работ

Для оценивания результатов обучения на зачете используется: качественная шкала; «зачтено», «не зачтено».

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Свободно владеет материалом, отвечает на все вопросы; в случае незнания небольшой части материала способен выстроить собственную логическую цепочку рассуждений и получить ответ, выполнение всех лабораторных и контрольных работ.	базовый	Зачтено
Недостаточно освоен материал, отвечает не на все вопросы, частичная сдача или отсутствие лабораторных и контрольных работ.	—	Не зачтено

20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ

ПК – 5 Способен проектировать, разрабатывать и верифицировать программное обеспечение информационных систем

Вопросы с вариантами ответов (закрытые)

1. Какая из строк кода, написанного на языке Python напечатает значение True?

- a. `print(bool(0))`
- b. `print(bool(-1))`
- c. `print(bool([]))`
- d. `print(bool())`

Ответ: b

2. Какая из строк кода, написанного на языке Python напечатает значение False?

- a. `print(bool('Hi'))`
- b. `print(bool(0.0))`
- c. `print(bool('0'))`
- d. `print(bool(int('10')))`

Ответ: b

3. На языке Python дан код. Какое значение примет переменная y?

`x = 2`

`y = x << 1`

- a. 4
- b. 1
- c. 21
- d. 2

Ответ: а

4. Каким будет значение переменной y после выполнения кода на языке Python?

```
x = 2
```

```
y = x | 4
```

- a. 6
- b. 2
- c. 4
- d. 0.5

Ответ: а

5. Чему будет равна переменная x после выполнения кода на языке Python?

```
x = 15 ^ 7
```

- a. 8
- b. 7
- c. 170859375
- d. 2

Ответ: а

6. Какое значение примет переменная x после выполнения следующего кода на языке Python?

```
x = 14 >> 1
```

- a. 7
- b. 14
- c. 1
- d. 28

Ответ: а

7. На языке Python дан код. Что будет напечатано в результате его работы?

```
y = '{} + {} = {}'.format(1, 2)
```

```
print(y)
```

- a. 1 + 1 = 2
- b. 1 + 2 = 2
- c. 1 + 2 =
- d. Исполнение кода вызовет ошибку

Ответ: d

8. На языке Python представлен код. Что будет напечатано в результате его выполнения?

```
x = '{0} + {0} = {1}'.format(1, sum = 2)
```

```
print(x)
```

- a. Исполнение кода вызовет ошибку
- b. 1 + 1 = 2
- c. {1} + {1} = {2}
- d. 1 + sum = 2

Ответ: а

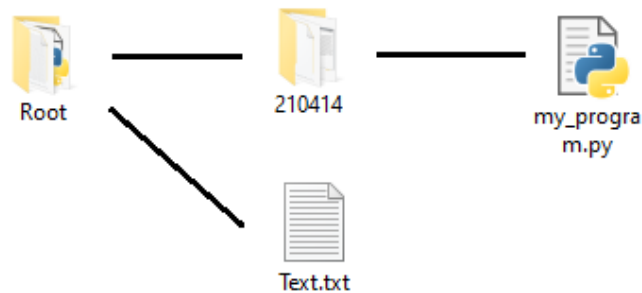
9. На языке Python дан код. Что будет напечатано в результате его работы?

```
print('{0:.> 7}'.format(1.23))
```

- a. .. 1.23
- b. False
- c. ...1.23
- d. ..+1.23

Ответ: a

10. Дан следующий каталог. Каков должен быть относительный путь к файлу Text.txt из сценария my_program.py?



- a. r'../Text.txt'
- b. r'210414/Text.txt'
- c. r'Root/210414/Text.txt'
- d. r'Text.txt'

Ответ: a

11. Что произойдет в результате работы кода, написанного на языке Python?

```
name = r'..\Files\Text.txt'  
file = open(name, 'rw')
```

- a. Возникнет ошибка
- b. Файл откроется для чтения и записи, но изначально пустым
- c. Файл откроется для чтения и записи, данные не сотрутся
- d. Файл откроется только на чтение

Ответ: a

12. Какой из режимов (язык Python) откроет файл для записи в конец?

- a. w
- b. r
- c. x
- d. a

Ответ: d

13. Какой из режимов (язык Python) откроет файл для чтения?

- a. r
- b. w
- c. wr
- d. a

Ответ: a

14. Какой из режимов (язык Python) работы с файлом недопустим?

- a. xt
- b. ab
- c. bt
- d. bx

Ответ: c

15. На языке Python дан код. Что будет напечатано в результате его работы?

```
x, y, z = 'GMC'  
print(x, y, z)
```

- a. G M C
- b. GMC GMC GMC
- c. Исполнение кода вызовет ошибку
- d. GMC

Ответ: a

16. На языке Python представлен код. Что будет напечатано в результате его выполнения?

```
x, y, z = 'Python'  
print(x, y, z)
```

- a. Python
- b. Python Python Python
- c. Py th on
- d. Исполнение кода вызовет ошибку

Ответ: d

17. На языке Python дан код. Что будет напечатано в результате его работы?

```
*x, y, z = 'GMC'  
print(x, y, z)
```

- a. ['G'] M C
- b. GMC
- c. Исполнение кода вызовет ошибку
- d. [G, M, C]

Ответ: a

18. На языке Python представлен код. Что будет напечатано в результате его выполнения?

```
w, *x, *y, z = 'Python'  
print(w, x, y, z)
```

- a. Исполнение кода вызовет ошибку
- b. Python *Python *Python Python
- c. P [yt] [ho] n
- d. P ['y', 't'] ['h', 'o'] n

Ответ: a

19. На языке Python дан код. Что будет напечатано в результате его работы?

```
x = y = 5
```

```
y = 100
print(x, y)
```

- a. 5 100
- b. 5 5
- c. 100 100
- d. 100 5

Ответ: a

20. На языке Python представлен код. Что будет напечатано в результате его выполнения?

```
x = y = [5]
y[0] = 100
print(x, y)
```

- a. [100] [100]
- b. [100] [5]
- c. [5] [100]
- d. [5] [5]

Ответ: a

Вопросы с кратким текстовым ответом (открытые)

1. На языке Python дан код. Какое значение примет переменная x?

```
x = int(2/3)
```

Ответ: 0

2. Каким будет значение переменной x после выполнения кода на языке Python?

```
x = round(2/3)
```

Ответ: 1

3. Чему будет равна переменная x после выполнения кода на языке Python?

```
x = round(1/2)
```

Ответ: 0

4. На языке Python дан код. Какое значение примет переменная x?

```
x = round(3/2)
```

Ответ: 2

5. Каким будет значение переменной x после выполнения кода на языке Python?

```
x = round(5/2)
```

Ответ: 2

6. Чему будет равна переменная x после выполнения кода на языке Python?

```
x = 17 % 17
```

Ответ: 0

7. Какое значение примет переменная x после выполнения следующего кода на языке Python?

```
x = 3 % 20
```

Ответ: 3

8. Фрагмент кода представлен на языке Python. Какое значение примет переменная x после его выполнения?

```
x = 3 % -10
```

Ответ: -7

9. На языке Python дан код. Какое значение примет переменная x?

```
x = 13 % 12
```

Ответ: 1

10. Каким будет значение переменной x после выполнения кода на языке Python?

```
x = 2 ** 10
```

Ответ: 1024

11. На языке Python дан код. Что будет напечатано в результате его работы?

```
x = True
```

```
y = False
```

```
if x:
```

```
    if y:
```

```
        print(1)
```

```
        print(2)
```

```
else:
```

```
    print(3)
```

Ответ: 2

12. На языке Python представлен код. Что будет напечатано в результате его выполнения?

```
x = False
```

```
y = True
```

```
if x:
```

```
    if y:
```

```
        print(1)
```

```
        print(2)
```

```
else:
```

```
    print(3)
```

Ответ: 3

13. На языке Python дан код. Что будет напечатано в результате его работы?

```
a = 4815162342
```

```
i = 0
```

```
while a:
```

```
    i += a % 10
```

```
    a //= 10
```

```
print(i)
```

Ответ: 36

14. На языке Python дан код. Какое значение примет переменная x?

```
x = 10 * 5 // +12
```

Ответ: 4

15. Каким будет значение переменной x после выполнения кода на языке Python (кавычки строкам указывать не нужно)?

```
s = 'Боже мой - подумал он. Неужели я глуп?'
```

```
x = s[1]
```

Ответ: о

16. Каким будет значение переменной x после выполнения кода на языке Python (кавычки строкам указывать не нужно)?

```
s = 'Но никто не должен об этом знать.'
```

```
x = s[:2]
```

Ответ: Но

17. Чему будет равна переменная x после выполнения кода на языке Python (кавычки строкам указывать не нужно)?

```
s = 'Ну, как ваше мнение?'
```

```
x = s[:5]
```

Ответ: Наше

18. Какое значение примет переменная x после выполнения следующего кода на языке Python?

```
s = 'Ткачи потребовали еще денег, шелка и золота'
```

```
x = s.find('ы')
```

Ответ: -1

19. Фрагмент кода представлен на языке Python. Какое значение примет переменная x после его выполнения?

```
s = "Do you believe in magic in a young girl's heart?"
```

```
x = s.index(' ')
```

Ответ: 2

20. На языке Python дан код. Какое значение примет переменная x?

```
a = True
```

```
b = False
```

```
x = not a or b
```

Ответ: False