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

УТВЕРЖДАЮ

Заведующий кафедрой
функционального анализа
и операторных уравнений

Каменский М.И.
подпись, расшифровка подписи
25.05.2023

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.05 Язык программирования Python

1. Код и наименование направления подготовки: 02.03.01 Математика и компьютерные науки

2. Профиль подготовки: математическое и компьютерное моделирование

3. Квалификация выпускника: бакалавр

4. Форма образования: очная

5. Кафедра, отвечающая за реализацию дисциплины: функционального анализа и операторных уравнений

6. Составители программы: Петрова Анастасия Александровна, к.ф.-м.н., преподаватель

7. Рекомендована: НМС математического факультета, протокол №0500-06 от 25.05.2023 г.

8. Учебный год: 2025-2026 **Семестр(ы): 5**

9. **Цели и задачи учебной дисциплины:**

Целями освоения учебной дисциплины являются:

- ознакомлении студентов с базовыми принципами программирования на языке Python и возможностями по использованию языка Python при решении задач профессиональной деятельности.

Задачи учебной дисциплины:

- изучение базовых конструкция языка Python;
- развитие навыков программирования на языке Python.

10. Место учебной дисциплины в структуре ООП: Место учебной дисциплины в структуре ОПОП: дисциплина относится к части, формируемой участниками образовательных отношений блока Б1.

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

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-1	Способен проводить работы по сбору, обработке и анализу научно-технической информации и результатов исследований в области математического моделирования физических и экономических процессов методами функционального анализа, а также реализовывать программно соответствующие математические алгоритмы	ПК-1.1	Обладает базовыми знаниями в области математического и компьютерного моделирования физических и экономических процессов	Знать: методы и средства языка программирования Python, основы объектной модели языка; Уметь: корректно формулировать текущие задачи курса; Владеть: практическим опытом решения вычислительных задач с помощью языка программирования Python
		ПК-1.2	Умеет использовать базовые знания в области математического моделирования физических и экономических процессов в профессиональной деятельности	Знать: основные факты курса; Уметь: применять изучаемые факты при решении задач; Владеть: навыком выбора знаний, необходимых для решения конкретной задачи.
ПК-2	Способен анализировать, систематизировать и обобщать передовой отечественный и международный опыт в области математического и компьютерного моделирования различных процессов	ПК-2.1	Владеет навыками анализа научных обзоров, публикаций, рефератов и библиографий по тематике проводимых исследований на русском и других языках	Знать: основные понятия и определения курса; Уметь: находить схожесть и отличие в понятиях; Владеть: навыками анализа литературы по тематике курса на русском и других языках
		ПК-2.2	Умеет обобщить информацию, полученную с помощью изучения библиографических материалов по тематике научных исследований в сфере математического и компьютерного моделиро-	Знать: перечень основных библиографических материалов по тематике курса; Уметь: выбрать ранее изученные факты, на которых необходимо строить решение конкретных задач; Владеть: навыком обобщения информации, полученной из разных библиографических источников.

			вания	

12. Объем дисциплины в зачетных единицах/часах в соответствии с учебным планом — 2/72

Форма промежуточной аттестации: зачёт

13. Виды учебной работы

Вид учебной работы	Трудоемкость (часы)		
	Всего	По семестрам	
		Сем.5	
		ч.	ч., в форме ПП
Аудиторные занятия	36	36	
в том числе: лекции	18	18	
практические			
лабораторные	18	18	
Самостоятельная работа	36	36	
Контроль			
Итого:	72	72	
Форма промежуточной аттестации		Зачёт	

13.1. Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
		1. Лекции	
1.1	Введение в язык программирования Python	Введение. История создания языка программирования Python. Особенности и преимущества Python. Области применения Python. Структура Python-программ. Понятие интерпретатора. Выполнение программ.	-
1.2	Понятие типа данных и переменной	Классификация типов данных. Оператор присваивания. Управление память и сборщик мусора	-
1.3	Скалярные типы данных	Целые числа, вещественные числа, Операции над числами. Логический тип. NoneType	-
1.4	Коллекции	Общие операции. Строки. Списки. Словари. Кортежи. Числовые диапазоны. Множества	--
1.5	Операторы в Python	Операторы if. Множественное ветвление. Значения истинности и булевские проверки. Циклы while. Операторы break, continue, pass и конструкция else цикла. Циклы for. Циклы с подсчетом: range.	-
1.6	Функции	Операторы def. Полиморфизм в Python. Основы областей видимости в Python. Основы передачи аргументов. Возврат нескольких значений. Рекурсия. Анонимные функции: выражения lambda	-

1.7	Работа с файлами	Файловые объекты в Python. Режимы открытия файлов. Основные методы для записи данных в файл и чтения данных из файла	-
1.8	Классы и объектно-ориентированное программирование	Классы и экземпляры. Вызовы методов. Перегрузка операций. Статические методы. Наследование. Абстрактные суперклассы	-
1.9	Исключения	Роли, исполняемые исключениями. Стандартный обработчик исключений. Перехват исключений. Генерация исключений. Исключения, определяемые пользователем. Оператор try/except/finally.Оператор raise. Оператор assert.	-
1.10	Модули и пакеты	Создание модулей. Импортирование и атрибуты. Оператор import. Оператор from.	-
3. Лабораторные занятия			
3.1	Введение в язык программирования Python	Знакомство с языком программирования Python Структура Python-программ. Выполнение программ.	-
3.2	Понятие типа данных и переменной	Знакомство с основными типами данных. Ввод/вывод данных. Оператор присваивания.	-
3.3	Скалярные типы данных	Работа с переменными целого, вещественного и логического типов	-
3.4	Коллекции	Основы работы со строки, списками, словарями, кортежами, множествами. Создание числовых диапазонов	--
3.5	Операторы в Python	Решение задач с использованием оператора if, множественного ветвления циклов while и for. Использование операторов break, continue, pass и конструкции else цикла.	-
3.6	Функции	Решение задач с разбиением кода программы на отдельные функции. Решение задач с использованием рекурсии. Использование lambda-функций	-
3.7	Работа с файлами	Решение задач с использованием считывания данных из текстовых файлов и записи данных в текстовые файлы	-
3.8	Классы и объектно-ориентированное программирование	Создание классов, содержащих различные методы и перегруженные операторы, объявление экземпляров классов и демонстрация возможностей классов. Создание классов-наследников.	-
3.9	Исключения	Добавление в программы обработки исключительных ситуаций. Использование операторов try/except/finally, raise, assert.	-
3.10	Модули и пакеты	Создание модулей. Использование операторов import и from.	-

13.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение в язык программирования Python	2		1	2	5
2	Понятие типа данных и переменной	1		1	1	3
3	Скалярные типы данных	1		1	1	3
4	Коллекции	2		1	2	5
5	Операторы в Python	2		2	6	10
6	Функции	2		2	6	10
7	Работа с файлами	2		2	4	8
8	Классы и	2		4	8	14

	объектно-ориентированное программирование					
9	Исключения	2		2	2	6
10	Модули и пакеты	2		2	4	8
	Всего	18		18	36	72

14. Методические указания для обучающихся по освоению дисциплины

Преподавание дисциплины заключается в чтении лекций и проведении лабораторных занятий. На лекциях рассказывается теоретический материал, на лабораторных занятиях обучающимся предлагается ряд индивидуальных заданий, которые необходимо выполнять в течение семестров для закрепления пройденного материала и успешного освоения дисциплины. При изучении курса «Язык программирования Python» обучающимся следует внимательно слушать и конспектировать материал, излагаемый на лекционных занятиях. Для его понимания и качественного усвоения обучающимся рекомендуется следующая последовательность действий.

1. После каждой лекции студентам рекомендуется подробно разобрать прочитанный теоретический материал, разобрать коды программ, рассмотренные на лекции. Перед следующей лекцией обязательно повторить материал предыдущей лекции.

2. Перед лабораторным занятием обязательно повторить лекционный материал. Если при выполнении индивидуальных заданий у обучающихся возникают вопросы, их нужно обязательно задать на занятии или в присутствующий час преподавателю.

3. При подготовке к лабораторным занятиям повторить основные понятия по темам, изучить примеры. Решая задачи, предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить индивидуальные задачи.

Самостоятельная учебная деятельность студентов по дисциплине «Язык программирования Python» предполагает изучение рекомендуемой преподавателем литературы по вопросам лекционных и лабораторных занятий (приведены выше), самостоятельное освоение понятийного аппарата и решение индивидуальных заданий.

Вопросы лекционных и лабораторных занятий обсуждаются на занятиях в виде устного опроса – индивидуального и фронтального. При подготовке к лекционным и лабораторным занятиям, обучающимся важно помнить, что их задача, отвечая на основные вопросы плана занятия и дополнительные вопросы преподавателя, показать свои знания и кругозор, умение логически построить ответ, владение основными средствами языка программирования Python и иные коммуникативные навыки, умение отстаивать свою профессиональную позицию. В ходе устного опроса выявляются детали, которые по каким-то причинам оказались недостаточно осмысленными студентами в ходе учебных занятий. Тем самым опрос выполняет важнейшие обучающую, развивающую и корректирующую функции, позволяет студентам учесть недоработки и избежать их при подготовке к промежуточным аттестациям.

Все выполняемые студентами самостоятельно задания (выполнение индивидуальных заданий) подлежат последующей проверке преподавателем. Результаты текущих аттестаций учитываются преподавателем при проведении промежуточной аттестации (5 семестр – зачет).

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

а) основная литература:

№ п/п	Источник
1	**Саммерфилд, М.** . Python на практике [Электронный ресурс] / Саммерфилд М. ; Пер. с англ. Слинкин А.А. — Москва : ДМК Пресс, 2014 .— 338 с. — Книга из коллекции ДМК Пресс- Информатика .— ISBN 978-5-97060-095-5 .— <URL:http://e.lanbook.com/books/element.php?pl1_id=66480>.
2	**Шелудько, В.М.** Основы программирования на языке высокого уровня Python : учебное пособие / Шелудько В.М. — Москва : ЮФУ, 2017 .— 146 с. — Основы программирования на языке высокого уровня Python [Электронный ресурс]: учебное пособие / Шелудько В. М. - Ростов н/Д : Изд-во ЮФУ, 2017. — ISBN 5-9275-2649-9 . <URL:https://www.studentlibrary.ru/ru/book/ISBN9785927526499.html>
3	**Шелудько, В.М.** Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / Шелудько В.М. — Москва : ЮФУ, 2017.— 107 с. — Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули [Электронный ресурс] : учебное пособие / Шелудько В. М. – Ростов н/Д : Изд-во ЮФУ, 2017. — ISBN 5-9275-2648-2 .— <URL:https://www.studentlibrary.ru/book/ISBN9785927526482.html>.

б) дополнительная литература:

№ п/п	Источник
1	**Северенс, Ч.** Введение в программирование на Python : учебное пособие / Северенс Ч. — Москва : ИНТУИТ, 2016 .— с. — Введение в программирование на Python [Электронный ресурс] / Северенс Ч. - М.: Национальный Открытый Университет "ИНТУИТ", 2016. — <URL:https://www.studentlibrary.ru/book/intuit_074.html>.
2	**Лучано, Р.** . Python. К вершинам мастерства [Электронный ресурс] / Лучано Р. ; Пер. с англ. Слинкин А.А. — Москва : ДМК Пресс, 2016 .— 768 с. — Книга из коллекции ДМК Пресс- Информатика .— ISBN 978-5-97060-384-0 .— <URL:https://e.lanbook.com/book/93273>.

в) информационные электронно-образовательные ресурсы**:**

№ п/п	Источник
1	Электронно-библиотечная система "Лань" https://e.lanbook.com/
2	Электронно-библиотечная система "Консультант студента" http://www.studmedlib.ru
3	Официальный сайт разработчика. – URL: https://www.python.org

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1.	Учебный курс «Программирование на языке высокого уровня (Python)». – URL: https://www.yuripetrov.ru/edu/python/index.html
2.	Положение об организации самостоятельной работы обучающихся в Воронежском государственном университете

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

При реализации учебной дисциплины проводятся различные типы лекций: вводная лекция, лекция-информация, лекция-диалог, лекция с применение современных компьютерных технологий (лекция-презентация), а также лабораторные за-

нятия, на которых осуществляется решение задач и устные опросы по темам занятия.

В части освоения материала лекционных и лабораторных занятий, самостоятельной работы по отдельным разделам дисциплины, прохождения текущей и промежуточной аттестации может применяться электронное обучение и дистанционные образовательные технологии, в частности, электронный курс «*Язык программирования Python*» *(*URL: https://edu.vsu.ru/course/view.php?id=27358*) на* портале «Электронный университет ВГУ».

Самостоятельная работа регламентируется Положением об организации самостоятельной работы обучающихся в Воронежском государственном университете.

18. Материально-техническое обеспечение дисциплины:

Для проведения лекционных занятий используются учебные аудитории, оснащенные специализированной мебелью.

Для проведения лабораторных занятий используются компьютерные классы, оснащенные специализированной мебелью, маркерной доской, персональными компьютерами.

Необходимое программное обеспечение: Python 2/3 (Python Software Foundation License (PSFL), бесплатное и/или свободное ПО, лицензия: https://docs.python.org/3/license.html)

19. Оценочные средства для проведения текущей и промежуточной аттестаций

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.	Разделы 1-10	ПК-1, ПК-2	ПК-1.1, ПК-1.2; ПК-2.1, ПК-2.2	Индивидуальные задания 1 - 6
Промежуточная аттестация форма контроля – зачёт				КИМ №1 из п.20.2

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств: индивидуальные задания

Примеры индивидуальных заданий

Задание 1

1. За каждой партой может сидеть два учащихся. Известно количество учащихся в каждом из трех классов. Выведите наименьшее число парт, которое нужно приобрести для них.

2. Дано натуральное число n. Напечатайте все n-значные нечетные натуральные числа в порядке убывания.
3. Дано несколько чисел. Вычислите их сумму. Сначала вводите количество чисел N, затем вводится ровно N целых чисел. Какое наименьшее число переменных нужно для решения этой задачи?

Задание 2

1. Дан одномерный массив числовых значений, насчитывающий N элементов. Подсчитать количество чисел, делящихся на 3 нацело, и среднее арифметическое чисел с чётными значениями. Поставить полученные величины на первое и последнее места в массиве (увеличив массив на 2 элемента).
2. Дан одномерный массив числовых значений, насчитывающий N элементов. Добавить к элементам массива такой новый элемент, чтобы сумма элементов с положительными значениями стала бы равна модулю суммы элементов с отрицательными значениями.
3. Дан список чисел. Определите, сколько в нем встречается различных чисел.

Задание 3

1. Дана строка, состоящая из слов, разделенных ровно одним пробелом. Определите, сколько в ней слов.
2. Дана строка. Если в этой строке буква f встречается только один раз, выведите её индекс. Если она встречается два и более раз, выведите индекс её первого и последнего появления. Если буква f в данной строке не встречается, ничего не выводите.
3. Из входной строки удалить все пробелы и определить, является ли новая строка палиндромом. Если строка — палиндром, удалить из неё все повторяющиеся символы, если нет, то напечатать перевёрнутую строку.

Задание 4

1. Из входной строки определить 3 наиболее часто встречаемых символа в ней. Пробелы нужно игнорировать (не учитывать при подсчете). Для выведения результатов вычислений требуется написать функцию top3(str). Итог работы функции top3 —строка следующего вида: символ – количество раз, символ – количество раз... например ‘a-5, m-3, h-1’.
2. Создайте список чисел. Используя lambda-функции, создайте список чисел, каждое из которых умножается на 3.
3. Дана последовательность чисел, завершающаяся числом 0. Найдите сумму всех этих чисел, не используя цикл.

Задание 5

1. Во входном файле записано два целых числа (через пробел), каждое в отдельной строке. Выведите в выходной файл их сумму.
2. В выходной файл выведите все строки наибольшей длины из входного файла, не меняя их порядок.
3. В файле могут быть записаны десятичные цифры и все, что угодно.

Числом назовем последовательность цифр, идущих подряд (т.е. число всегда неотрицательно).Вычислите сумму всех чисел, записанных в файле. В данной задаче удобно считывать данные посимвольно.

Задание 6

Реализовать класс Dot, принимающий координаты точки; класс Line, принимающий 2 координаты. Перегрузить оператор сложения в Dot для получения экземпляра Line. Реализовать 2 варианта создания Line: Dot+Dot/две пары координат.

На основе имеющихся классов необходимо создать класс Triangle, образуемый с помощью трех точек (x, y). Перегрузить оператор + для Dot, чтобы сумма трех точек приводила к экземпляру Triangle, а также сумма Dot + Line => Triangle.

Описание технологии проведения

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета.

Текущий контроль предназначен для проверки хода и качества формирования компетенций, стимулирования учебной работы обучаемых и совершенствования методики освоения новых знаний. Он обеспечивается контролем выполнения обучающимися индивидуальных заданий.

При текущем контроле уровень освоения учебной дисциплины и степень сформированности компетенции определяются оценками «зачтено» и «не зачтено». Индивидуальные выдаются обучающимся для внеаудиторной самостоятельной работы. Срок выполнения одного здания – две недели. Если при выполнении индивидуальных заданий у обучающихся возникают вопросы, их нужно обязательно задать на занятии или в присутствующий час преподавателю. По результатам выполнения индивидуальных заданий осуществляется устный опрос обучающихся.

Требования к выполнению заданий

Для оценивания результатов обучения на контрольной работе используются следующие показатели:

1) знание учебного материала и владение понятийным аппаратом;

2) умение связывать теорию с практикой;

3) умение применять полученные знания в практическом задании.

20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: КИМ №1.

КИМ №1

Необходимо реализовать класс Vect, унаследованный от list, со следующими изменениями:

1. При создании экземпляра класса должно проверяться, что
   а. Все элементы последовательности одного типа (isinstance)
   б. Этот тип поддерживает сложение и умножение на число

2. В противном случае возникает ошибка:
raise Exception('TypeError!')

3. Класс поддерживает поэлементное умножение на число: vect * 5

4. Класс поддерживает поэлементное сложение с неперичислимыми типами данных(вместо конкатенации): vect + 5

5. Класс поддерживает запись в файл vect.write('filename') и чтение из файла vect.read('filename')

6. Класс поддерживает метод vect.sort, производящий сортировку и переписывающий экземпляр класса в упорядоченном виде

7. Класс поддерживает метод vect.stat(), выводящий каждый уникальный элемент экземпляра класса и через дефис "-" количество вхождений значения в экземпляр класса

8. Класс поддерживает метод vect.clear(), удаляющий все дубликаты из экземпляра класса и перезаписывая его.

9. Класс поддерживает метод сложения с другими экземплярами класса Vect (конкатенация)

10. Класс поддерживает статический метод vect.check(), проверяющий пригодность данных для инициализации экземпляра класса Vect на их основе.

Описание технологии проведения

Промежуточная аттестация предназначена для определения уровня освоения всего объема учебной дисциплины. Промежуточная аттестация по дисциплине «Язык программирования Python» проводится в форме зачета.

Промежуточная аттестация, как правило, осуществляется в конце семестра. В ходе проведения зачёта обучающемуся выдается КИМ с практической задачей и предлагается написать соответствующую программу на языке программирования Python . В ходе выполнения заданий разрешено пользоваться справочными материалами, а также электронными носителями, имеющимися у обучающихся. Запрещено использование любых средст связи. Ограничение по времени - 90 минут. При проведении зачета учитываются результаты текущих аттестаций. Если студент имеет оценки «зачтено» по всем индивидуальным заданиям (задания 1 – 6 из п. 20.1), то он освобождается от решения задачи на зачёте.

Требования к выполнению заданий, шкалы и критерии оценивания

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Обучающийся верно выполнил 5 и более пунктов зачётного задания	*Базовый уровень*	*Зачтено*
Выполнено менее 5 пунктов зачётного задания	-	*Не зачтено*

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

1) закрытые задания (тестовые):

1. Ниже приведён код программы на языке Python. Что будет найдено в результате её выполнения?

```
n=int(input())
k=0
while (n>0):
   k+=n%10
   n//=10
print(k)
```

а) Сумма первых 10 натуральных чисел
б) Сумма цифр числа, введённого пользователем
в) Первая цифра числа, введённого пользователем
г) Остаток от деления на 10 числа, введённого пользователем

Ответ: б)
Комментарий.

```
n=int(input()) //Ввод целого числа пользователем
k=0
while (n>0): //До тех пор пока введённое числе не станет равно 0 (то
есть пока в нём не закончатся цифры)
   k+=n%10 //Складываем последовательно цифры числа (операция n%10 –
отделение последней цифры от числа)
   n//=10 //в результате этой операции на каждом проходе цикла «удаля-
ем» из текущего числа последнюю цифру
print(k)
```

2. Ниже приведён код программы на языке Python. Что будет найдено в результате её выполнения?

```
a=int(input())
f=1
while (a!=1):
    f*=a
    a-=1
print(f)
```

а) Значение факториала числа, введённого пользователем
б) Последняя цифра числа, введённого пользователем
в) Количество цифр в числе, введённом пользователем
г) Произведение цифр числа, введённого пользователем

Ответ: а)
Комментарий. В приведённом цикле while находится произведение a*(a-1)*(a-2)*…*2, то есть значение факториала числа a, которое введёт пользователь.

3. Ниже приведён код программы на языке Python. Что будет выведено на экран в результате её выполнения, если пользователь введёт число 3?

```
n=int(input())
s = ''
```

```
for i in range(1,n+1):
    s+=str(i)
    print(s)
```

а) 123

б) 1
   2
   3

В) 1
   12
   123

Г)1234

Ответ: в)
Комментарий. Цикл for перебирает целые числа 1, 2, 3. На каждой итерации текущее число приписывается в конец строки s, после чего производится вывод текущей строки s.

4. Ниже приведён фрагмент кода программы на языке Python.

```
n=0
for i in range (len(A)-1):
if (A[i]==0 and A[i+1]==0):
        n+=1
        break
if (n==0):
    print('Таких пар нет')
else:
    print('Такая пара есть')
```

Для какого из массивов в результате выполнения этой программы будет выведено сообщение «Такая пара есть»?
А) A=[23,35,60,88,92,55,0,11,93,84,31,95,9,74,23]
Б) A=[23,35,60,88,92,55,0,11,93,0,31,95,9,74,23]
В) A=[23,35,60,88,92,55,0,0,93,84,31,95,9,74,23]
Г) A=[23,23,60,60,92,92]

Ответ: в)
Комментарий. В приведённом фрагменте кода определяется, имеются ли в массиве два подряд идущих нуля. Два подряд идущихз нуля встречаются только в массиве под буквой в).

5. Ниже приведён фрагмент кода программы на языке Python. Что произойдёт в ходе его выполнения?

```
with open("text.txt","r") as f:
```

```
    s=f.readline()
```

а) Из файла с именем text.txt будет считана строка текста
б) В файл с именем text.txt будет записана строка текста
в) Из файла с именем text.txt будет считан один символ текста
г) Это некорректный фрагмент кода, будет выведено сообщение об ошибке

Ответ: а)
Комментарий: функция readline() служит для считывания из файла строки текста. Файл открыт в режиме для чтения (“r”), так что чтение из файла будет произведено корректно.

2) открытые задания:

1. Ниже приведён код программы на языке Python. Что будет выведено на экран в результате её выполнения?

```
A=[1, 3, 7, 8, 9, 25, 12]
k=0
for i in A:
    if(i%3==0):
        k+=1
print(k)
```

Ответ: 3
Комментарий. Это программа для подсчёта количества элементов массива, кратных трём. В приведённом в коде массиве А таких элемента 3: 3, 9, 12.

2. Ниже приведён код программы на языке Python. Что будет выведено на экран в результате её выполнения?

```
A=[1, 3, 7, 8, 9, 25, 12]
s=0
k1=0
for i in A:
    if(i%2==0):
        s+=i
        k1+=1
s/=k1
print(s)
```

Ответ: 10
Комментарий. Это программа для подсчёта среднего арифметического чётных элементов массива. В приведённом в коде массиве А содержится два таких элемента: 8 и 12. Таким образом, среднее арифметическое равно: (8+12)/2=10.

3. В файле с именем 1.txt записаны два числа: 3, 5 (каждое в отдельной строке). Ниже приведён код программы на языке Python. Что будет записано в файле 2.txt после её выполнения?

```
with open ("1.txt","r") as fin:
    a=int(fin.readline())
    b=int(fin.readline())
    with open("2.txt", "w") as fout:
        fout.write(str(a+b))
```

Ответ: 8

Комментарий. Програма считывает числа, записанные в файле 1.txt в пременные a и b. А затем записывает в файл 2.txt сумму их значений. 3+5=8.

4. Ниже приведён код программы на языке Python. Что будет выведено на экран в результате её выполнения?

```
class Soda():
def __init__(self,string=''):
        self.__a=string
def show_my_drink(self):
if self.__a:
            print("Газировка и "+self.__a)
 else:
            print("Обычная газировка")
x=Soda()
x.show_my_drink()
```

Ответ: Обычная газировка

Комментарий. В программе создан класс Soda для определения типа газированной воды, принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду).
В этом классе реализован метод show_my_drink(), выводящий на печать «Газировка и {ДОБАВКА}» в случае наличия добавки,
а иначе отображающий фразу: «Обычная газировка». Поскольку для объекта x метод show_my_drink() вызывается без аргумента, то будет выведена фраза «Обычная газировка».

5. Ниже приведён код программы на языке Python. Что будет выведено на экран в результате её выполнения?

```
class Soda():
def __init__(self,string=''):
        self.__a=string
def show_my_drink(self):
if self.__a:
            print("Газировка и "+self.__a)
 else:
            print("Обычная газировка")
y=Soda('лимонный сироп')
y.show_my_drink()
```

Ответ: Газировка и лимонный сироп

Комментарий. В программе создан класс Soda для определения типа газированной воды, принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду).
В этом классе реализован метод show_my_drink(), выводящий на печать «Газировка и {ДОБАВКА}» в случае наличия добавки,
а иначе отображающий фразу: «Обычная газировка». Поскольку для объекта у метод show_my_drink() вызывается с аргументом `'лимонный сироп'` , то будет выведена фраза «Газировка и лимонный сироп».

Критерии и шкалы оценивания заданий ФОС:

1) Задания закрытого типа (выбор одного варианта ответа, верно/неверно):

- 1 балл – указан верный ответ;
- 0 баллов – указан неверный ответ.

2) Задания закрытого типа (множественный выбор):

- 2 балла – указаны все верные ответы;
- 0 баллов — указан хотя бы один неверный ответ.

3) Задания закрытого типа (на соответствие):

- 2 балла – все соответствия определены верно;
- 0 баллов – хотя бы одно сопоставление определено неверно.

4) Задания открытого типа (короткий текст):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

5) Задания открытого типа (число):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).