

Компонент ОПОП 09.04.01 Информатика и вычислительная техника
Направленность (профиль) Компьютерный анализ и интерпретация данных.
Data Science.
ШИФР дисциплины

Б1.О.08
шифр дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Дисциплины
(модуля)

Основы Python

Разработчик:

Золотов О.В.

ФИО

доцент

должность

канд. физ.-мат. наук

ученая степень,

звание


Утверждено на заседании кафедры

Информационных технологий

наименование кафедры

протокол № 6 от 01.02.2024

Заведующий кафедрой ИТ



подпись

Ляш О.И.

ФИО

1. Критерии и средства оценивания компетенций и индикаторов их достижения, формируемых дисциплиной (модулем)

Код и наименование компетенции	Код и наименование индикатора(ов) достижения компетенции	Результаты обучения по дисциплине (модулю)			Оценочные средства текущего контроля	Оценочные средства промежуточной аттестации
		<i>Знать</i>	<i>Уметь</i>	<i>Владеть</i>		
<p>ОПК-2.</p> <p>Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач</p>	<p>ИД-1_{опк-2} Понимает принципы и применяет методы разработки алгоритмов и программных средств</p> <p>ИД-2_{опк-2} Использует современные интеллектуальные технологии при разработке алгоритмов и программных средств</p> <p>ИД-2_{опк-3} Решает профессиональные задачи путем разработки оригинальных алгоритмов и программных средств</p>	<p>- основные принципы и методы разработки алгоритмов и программных продуктов на языке программирования высокого уровня (ЯВУ) Python 3;</p> <p>- современные методы и технологии, применяемые при разработке алгоритмов и программных средств на ЯВУ Python 3;</p> <p>- существующие в Python-сообществе лучшие</p>	<p>- применять принципы и методы разработки алгоритмов и программных продуктов на ЯВУ Python 3;</p> <p>- применять современные методы и технологии, используемые при разработке алгоритмов и программных средств на ЯВУ Python 3;</p> <p>- решать профессиональные задачи путем разработки оригинальных алгоритмов и программных средств на ЯВУ</p>	<p>- навыком применения принципов и методов разработки алгоритмов и программных продуктов на ЯВУ Python 3;</p> <p>- навыком применения современных методов и технологий, используемых при разработке алгоритмов и программных средств на ЯВУ Python 3;</p> <p>- навыком решения профессиональных задач путем разработки</p>	<p>- комплект заданий для выполнения лабораторных работ;</p> <p>- типовые задания по вариантам для выполнения расчетно-графической работы.</p>	<p>Экзаменационные билеты</p> <p>Результаты текущего контроля</p>

<p>ОПК-5.</p> <p>Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем</p>	<p>ИД-1_{опк-5} Понимает принципы построения и функционирования информационных и автоматизированных систем</p> <p>ИД-2_{опк-5} Разрабатывает и модернизирует программное обеспечение информационных и автоматизированных систем</p> <p>ИД-3_{опк-5} Разрабатывает и модернизирует аппаратное обеспечение информационных и автоматизированных систем</p>	<p>практики, применяемые при решении профессиональных задач на ЯВУ Python 3;</p> <p>- основы построения и функционирования информационных и автоматизированных систем.</p>	<p>Python 3;</p> <p>- применять в профессиональной деятельности принципы построения и функционирования информационных и автоматизированных систем на ЯВУ Python 3.</p>	<p>оригинальных алгоритмов и программных средств на ЯВУ Python 3;</p> <p>- навыком применения принципов построения и функционирования элементов информационных и автоматизированных систем.</p>		
--	--	--	--	---	--	--

2. Оценка уровня сформированности компетенций (индикаторов их достижения)

Показатели оценивания компетенций (индикаторов их достижения)	Шкала и критерии оценки уровня сформированности компетенций (индикаторов их достижения)			
	Ниже порогового («неудовлетворительно»)	Пороговый («удовлетворительно»)	Продвинутый («хорошо»)	Высокий («отлично»)
Полнота знаний	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущены не грубые ошибки.	Уровень знаний в объёме, соответствующем программе подготовки. Допущены некоторые погрешности.	Уровень знаний в объёме, соответствующем программе подготовки.
Наличие умений	При выполнении стандартных заданий не продемонстрированы основные умения. Имели место грубые ошибки.	Продemonстрированы основные умения. Выполнены типовые задания с не грубыми ошибками. Выполнены все задания, но не в полном объёме (отсутствуют пояснения, неполные выводы)	Продemonстрированы все основные умения. Выполнены все основные задания с некоторыми погрешностями. Выполнены все задания в полном объёме, но некоторые с недочётами.	Продemonстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Задания выполнены в полном объёме без недочётов.
Наличие навыков (владение опытом)	При выполнении стандартных заданий не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для выполнения стандартных заданий с некоторыми недочётами.	Продemonстрированы базовые навыки при выполнении стандартных заданий с некоторыми недочётами.	Продemonстрированы все основные умения. Выполнены все основные и дополнительные задания без ошибок и погрешностей. Продemonстрирован творческий подход к решению нестандартных задач.
Характеристика сформированности компетенции	Компетенции фактически не сформированы. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных) задач. ИЛИ Зачетное количество баллов не набрано согласно установленному диапазону	Сформированность компетенций соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач. ИЛИ Набрано зачетное количество баллов согласно установленному диапазону	Сформированность компетенций в целом соответствует требованиям. Имеющихся знаний, умений, навыков достаточно для решения стандартных профессиональных задач. ИЛИ Набрано зачетное количество баллов согласно установленному диапазону	Сформированность компетенций полностью соответствует требованиям. Имеющихся знаний, умений, навыков в полной мере достаточно для решения сложных, в том числе нестандартных, профессиональных задач. ИЛИ Набрано зачетное количество баллов согласно установленному диапазону

3. Критерии и шкала оценивания заданий текущего контроля

3.1 Критерии и шкала оценивания лабораторных работ

Перечень лабораторных работ, описание порядка выполнения и защиты работы, требования к результатам работы, структуре и содержанию отчета и т.п. представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

Оценка/баллы	Критерии оценивания
<i>Отлично / 3</i>	Задание выполнено полностью и правильно. Отчет по лабораторной/практической работе подготовлен качественно в соответствии с требованиями. Полнота ответов на вопросы преподавателя при защите работы.
<i>Хорошо / 2.5</i>	Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений. Все требования, предъявляемые к работе, выполнены.
<i>Удовлетворительно / 2</i>	Задания выполнены частично с ошибками. Демонстрирует средний уровень выполнения задания на лабораторную/практическую работу. Большинство требований, предъявляемых к заданию, выполнены.
<i>Неудовлетворительно / 0</i>	Задание выполнено со значительным количеством ошибок на низком уровне. Многие требования, предъявляемые к заданию, не выполнены. ИЛИ Задание не выполнено.

3.2 Формы текущего контроля успеваемости

Критерии и шкала оценивания расчетно-графической работы

Перечень заданий, рекомендации по выполнению представлены в методических материалах по освоению дисциплины (модуля) и в электронном курсе в ЭИОС МАУ.

В ФОС включен типовой вариант расчетно-графической работы:

Разработать на языке Python3 консольное программное средство. Тематика разрабатываемого программного средства определяются вариантом. Разрабатываемое приложение должно удовлетворять следующим критериям:

- должны быть реализована вся указанная в варианте функциональность;
- исходный код решения должен удовлетворять требованиям PEP-8;
- все функции и методы должны содержать документационные строки (docstrings);
- программное средство должно быть оформлено в виде Python-пакета в формате «колесо» (wheel);
- сформированный Python-пакет должен быть размещен в указателе пакетов test.pypi.org;
- программное средство должно сопровождаться тестами и примерами использования.

Варианты тем РГР:

1. Реализовать консольное приложение - телефонный справочник. Приложение должно поддерживать создание и удаление телефонной книги, создание удаление и

обновление записей телефонной книги, поиск отдельных записей и групп записей по критерию.

2. Реализовать консольное приложение – справочник праздников. В рамках одного приложения должна быть обеспечена возможность ведения разных по номенклатуре (номенклатуру определяет пользователь, несколько позиций номенклатуры должно быть реализовано заранее) праздников. Приложение должно поддерживать создание, удаление и обновление позиций номенклатуры, праздников в рамках отдельной номенклатуры, поиск отдельных записей и групп записей по критерию.
3. Реализовать консольное приложение, реализующее следующее:
 - для заданного текста расчёт
 - a) перечня уникальных слов в тексте;
 - b) частот встречаемости слов в тексте;
 - вывод
 - c) наименее часто встречающихся слов;
 - d) наиболее часто встречающихся слов;
 - e) слов из 1го квартиля;
 - f) слов из 4го квартиля;
 - g) слов из заданного перцентиля;
 - h) слов с заданным диапазоном встречаемости;
 - построение облака слов.

Входные данные должны поступать из стандартного потока ввода или из файла (согласно заданному ключу командной строки при вызове приложения). Выходные данные должны выводиться в стандартный поток вывода или в текстовый файл (согласно заданному ключу командной строки при вызове приложения).

Оценка/баллы	Критерии оценивания
<i>Отлично / 34</i>	Работа выполнена полностью, без ошибок (возможна одна неточность, описка, не являющаяся следствием непонимания материала).
<i>Хорошо / 33</i>	Работа выполнена полностью, но обоснования шагов решения недостаточны, допущена одна негрубая ошибка или два-три недочета, не влияющих на правильную последовательность рассуждений.
<i>Удовлетворительно / 31</i>	В работе допущено более одной грубой ошибки или более двух-трех недочетов, но обучающийся владеет обязательными умениями по проверяемой теме.
<i>Неудовлетворительно / 0</i>	В работе есть грубые ошибки и недочеты ИЛИ Контрольная работа не выполнена.

Критерии и шкала оценивания посещаемости занятий

Посещение занятий обучающимися определяется в процентном соотношении

Баллы	Критерии оценки
10	посещаемость 75 - 100 %
5	посещаемость 50 - 74 %
0	посещаемость менее 50 %

4. Критерии и шкала оценивания результатов обучения по дисциплине (модулю) при проведении промежуточной аттестации

Критерии и шкала оценивания результатов освоения дисциплины (модуля) с экзаменом

Для дисциплин (модулей), заканчивающихся экзаменом, результат промежуточной аттестации складывается из баллов, набранных в ходе текущего контроля и при проведении экзамена:

В ФОС включен список вопросов и заданий к экзамену и типовой вариант экзаменационного билета:

Список вопросов к экзамену:

1. Интерпретируемые языки программирования, их особенности. Виртуальная машина Python. Проблема GIL.
2. Виртуальное окружение Python (на примере venv).
3. Интерактивная среда разработки PyCharm Community, ее основные возможности, установка и настройка.
4. Особенности написания программ на Python: форматирование и оформление (PEP8).
5. Простая программа на Python3. Кодировка. Комментарии. Идентификаторы и ключевые слова. Литералы. Операторы. Разделители.
6. Понятие типа данных в Python. Переменные. Допустимые операции. Множественное присваивание / позиционное присваивание.
7. Понятие модуля. «Простые» встроенные типы Python: целые, с плавающей точкой и комплексные; булевы (логический), диапазоны (range vs xrange). Тип None.
8. Строки в Python: «простые» операции над строками, срезы.
9. Консольный ввод-вывод.
10. Управление потоком исполнения: ветвления и циклы.
11. Функции zip(), enumerate().
12. Списки, операции со списками, особенности представления в памяти. Срезы. Списковые выражения (списковые включения) и генераторы списков.
13. Кортежи. Создание кортежей при помощи генераторов. Упаковка и распаковка переменных.
14. Ассоциативные контейнеры (словари). Словарные выражения (включения) и генераторы словарей.
15. Множества. Множественные выражения и генераторы множеств.
16. Главная точка входа __main__. Оператор pass. Оператор assert.
17. Импорт пакетов, классов, функций. Определение пользовательских функций.
18. Области видимости функций и переменных (global и nonlocal). Позиционные аргументы функций. Именованные аргументы. Значения по умолчанию.
19. Пустой список как значение по умолчанию формального аргумента функции. Функции с произвольным количеством аргументов.
20. Передача функции как аргумента. Лямбда-выражения.
21. Функции sum, min, max, sorted, all, any, map, reduce, filter, apply. Декораторы.
22. Исключения. Обработка исключений.

23. Файловый ввод-вывод. Обращение к стандартным потокам ввода-вывода как к файловым потокам.
24. Менеджеры контекста.
25. Работа с датой и временем.
26. Коллекции Python.
27. Типы bytes, bytearray, memoryview.
28. Общая характеристика пакетов sys, os, copy, math, cmath, random.
29. Определение собственных классов. Поля и методы классов, их область видимости.
30. Управление доступом к полям и методам класса. Свойства.
31. Классы. Наследование. Суперклассы и подклассы.
32. Классы. Множественное наследование.
33. Статические классы и статические методы.
34. Основные возможности pytest. Организация тестового кода.
35. Пакеты в Python3. Форматы 'egg' («яйцо») и 'wheel' («колесо»).
36. Основные возможности (сценарии использования) setuptools, build.
37. Стандартная структура и назначение файлов и папок пакета.
38. Документирование зависимостей пакетов.
39. Включение произвольных бинарных файлов в пакет.
40. Вложенные пакеты. Тестирование пакета на корректность.
41. Публичные репозитории пакетов PyPI и Test.PyPI. Загрузка и размещение пакетов в PyPI и Test.PyPI.

Список заданий

1. Напишите функцию, принимающую произвольное количество аргументов (известное только на этапе вызова функции) и выводящая их на печать (каждый аргумент с новой строки)
2. Напишите функцию, осуществляющую подсчет частот появления слов в файле.
3. Напишите класс, реализующий операции сложения, вычитания и умножения множеств. Под умножением понимается прямое декартово произведение.
4. Реализуйте декоратор для функции, не принимающей аргументов
5. Реализуйте класс, осуществляющий проверку корректности присваиваемых значений полям класса. Для проверки корректности используйте механизм свойств.
6. Пользователь задает две последовательности пар значений вида строка-число. Найдите объединение, пересечение и разность указанных последовательностей. (реализуйте программу)
7. Пользователь вводит с консоли в бесконечном цикле значения. Определите, является ли введенное значение строкой, целым числом или числом с плавающей точкой. (реализуйте программу)

Типовой вариант экзаменационного билета:

Федеральное государственное автономное образовательное учреждение
высшего образования
«МУРМАНСКИЙ АРКТИЧЕСКИЙ УНИВЕРСИТЕТ»
кафедра Информационных технологий
направление обучение 09.04.01 «Информатика и вычислительная техника»

Основы Python

БИЛЕТ № 2

Вопрос 1

Списки, операции со списками, особенности представления в памяти. Срезы. Списковые выражения (списковые включения) и генераторы списков.

Вопрос 2

Декораторы функций

Вопрос 3.

Задача.

Оценка	Критерии оценки ответа на экзамене
<i>Отлично</i>	Обучающийся глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, не затрудняется с ответом при видоизменении вопроса. Владеет специальной терминологией, демонстрирует общую эрудицию в предметной области, использует при ответе ссылки на материал специализированных источников, в том числе на Интернет-ресурсы.
<i>Хорошо</i>	Обучающийся твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, владеет специальной терминологией на достаточном уровне; могут возникнуть затруднения при ответе на уточняющие вопросы по рассматриваемой теме; в целом демонстрирует общую эрудицию в предметной области.
<i>Удовлетворительно</i>	Обучающийся имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, плохо владеет специальной терминологией, допускает существенные ошибки при ответе, недостаточно ориентируется в источниках специализированных знаний.
<i>Неудовлетворительно</i>	Обучающийся не знает значительной части программного материала, допускает существенные ошибки, нарушения логической последовательности в изложении программного материала, не владеет специальной терминологией, не ориентируется в источниках специализированных знаний. Нет ответа на поставленный вопрос.

Оценка, полученная на экзамене, переводится в баллы («5» - 20 баллов, «4» - 15 баллов, «3» - 10 баллов) и суммируется с баллами, набранными в ходе текущего контроля.

Итоговая оценка по дисциплине (модулю)	Суммарные баллы по дисциплине (модулю), в том числе	Критерии оценивания
<i>Отлично</i>	91 - 100	Выполнены все контрольные точки текущего контроля на высоком уровне. Экзамен сдан
<i>Хорошо</i>	81-90	Выполнены все контрольные точки текущего контроля. Экзамен сдан
<i>Удовлетворительно</i>	70- 80	Контрольные точки выполнены в неполном объеме. Экзамен сдан
<i>Неудовлетворительно</i>	69 и менее	Контрольные точки не выполнены или не сдан экзамен

5. Задания диагностической работы для оценки результатов обучения по дисциплине (модулю) в рамках внутренней и внешней независимой оценки качества образования

ФОС содержит задания для оценивания знаний, умений и навыков, демонстрирующих уровень сформированности компетенций и индикаторов их достижения в процессе освоения дисциплины (модуля).

Комплект заданий разработан таким образом, чтобы осуществить процедуру оценки каждой компетенции, формируемых дисциплиной (модулем), у обучающегося в письменной форме.

Содержание комплекта заданий включает *тестовые задания*.

Комплект заданий диагностической работы

ОПК-2. Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	
1	<p>Следующий код</p> <pre>def demo_fun(): ... demo_fun()</pre> <p>является</p> <p>А. псевдокодом (из-за наличия многоточия интерпретатор не сможет выполнить код) Б. некорректным Python3-кодом В. корректным Python3-кодом, использующим объект Ellipsis Г. нет верного ответа</p> <p>Ответ: В. корректным Python3-кодом, использующим объект Ellipsis</p>
2	<p>Следующий код</p> <pre>t = (1,)</pre> <p>А. присваивает переменной t значение 1 (целочисленную единицу) Б. присваивает переменной t кортеж, содержащий один элемент, равный единице В. присваивает переменной t множество, содержащее один элемент Г. нет верного ответа</p> <p>Ответ:</p>

3	<p>Б. присваивает переменной <code>t</code> кортеж, содержащий один элемент, равный единице</p> <p>Дан фрагмент Python3-кода</p> <pre>var1, = temp</pre> <p>Известно, что в переменной <code>temp</code> хранится объект типа <code>list</code> (список). Этот код выполнится без ошибок, если</p> <p>A. <code>len(temp) == 0</code> Б. <code>len(temp) == 1</code> В. <code>len(temp) > 0</code> Г. <code>len(temp) > 1</code></p> <p>Ответ: Б. <code>len(temp) == 1</code></p>
4	<p>Дан следующий код</p> <pre>def foo(a, b): print(f"foo is called with {a} and {b}") foo(1, 2) foo(b=2, a=1)</pre> <p>A. первый вызов функции <code>foo</code> является корректным, а второй – нет Б. второй вызов функции <code>foo</code> является корректным, а первый – нет В. оба вызова функции <code>foo</code> являются корректными Г. нет верного ответа</p> <p>Ответ: В. оба вызова функции <code>foo</code> являются корректными</p>
5	<p>Дан следующий код</p> <pre>def bar(*, a, b): print(f"bar is called with {a} and {b}") bar(1, 2) bar(b=2, a=1)</pre> <p>A. первый вызов функции – <code>bar(1, 2)</code> – является корректным Б. первый вызов функции – <code>bar(1, 2)</code> – является некорректным В. второй вызов функции – <code>bar(b=2, a=1)</code> – является некорректным Г. оба вызова функции <code>bar</code> являются корректными</p> <p>Ответ: Б. первый вызов функции – <code>bar(1, 2)</code> – является некорректным</p>

6

Дан следующий код

```
def bar():  
    print("Empty bar")  
  
def bar(a):  
    print(f"some bar with {a}")  
  
bar()
```

В результате его выполнения

- А. будет выведено сообщение «Empty bar», т.к. Python3 поддерживает перегрузку функций
- Б. при вызове функции bar() интерпретатор выдаст сообщение об ошибке (т.к. требуется 1 пропущенный аргумент)
- В. Python3 не поддерживает перегрузку функций, поэтому интерпретатор сгенерирует сообщение об ошибке при обнаружении второго определения функции bar
- Г. нет верного ответа

Ответ:

- Б. при вызове функции bar() интерпретатор выдаст сообщение об ошибке (т.к. требуется 1 пропущенный аргумент)

7

Дан следующий код

```
def my_sum(a, b):  
    print(a + b)  
  
my_sum(1, 2)  
my_sum('1', '2')
```

В результате его выполнения

- А. в стандартный поток вывода будут напечатаны значения «3» и «12» (без кавычек, на разных строках)
- Б. в стандартный поток вывода будет напечатано значение «3», а при втором вызове интерпретатор выдаст сообщение об ошибке (несоответствии типа передаваемых аргументов формальным параметрам функции)
- В. в стандартном потоке вывода будет напечатано «312» (без кавычек)
- Г. нет верного ответа

Ответ:

- А. в стандартный поток вывода будут напечатаны значения «3» и «12» (без кавычек, на разных строках)

8

Дан следующий код

```
print(list(range(1, 10)))
```

В результате его выполнения в стандартный поток вывода будет напечатано

- А. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
- Б. [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- В. [1, 2, 3, 4, 5, 6, 7, 8, 9]
- Г. нет верного ответа

Ответ:

- В. [1, 2, 3, 4, 5, 6, 7, 8, 9]

9

Дан следующий код

```
lst1 = [1, 2, 3]
lst2 = lst1
lst3 = lst1[:]

lst3[1] = 'x'
print(lst1)

lst2[1] = 'y'
print(lst1)
```

В результате его выполнения в стандартный поток вывода будет напечатано

- А.
[1, 2, 3]
[1, 'y', 3]
- Б.
[1, 'x', 3]
[1, 'y', 3]

- В.
[1, 'x', 3]
[1, 'x', 3]

Г. нет верного ответа

Ответ:

- А.
[1, 2, 3]
[1, 'y', 3]

10	<p><i>Дан следующий код</i></p> <pre data-bbox="341 271 919 566">class demo_class: def __init__(self): self.a = 'a' demo0 = demo_class() demo1 = demo_class() demo1.b = 'b'</pre> <p><i>В результате его выполнения</i></p> <p>А. интерпретатор Python3 выдаст сообщение об ошибке, так как в классе demo_class нет поля b</p> <p>Б. в экземпляр класса demo1 будет динамически добавлено поле b</p> <p>В. как в demo1, так и в demo0 будет динамически добавлено поле b</p> <p>Г. нет верного ответа</p> <p>Ответ: Б. в экземпляр класса demo1 будет динамически добавлено поле b</p>
<p>ОПК-5. Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем</p>	
1	<p><i>Обычно файлы с программами на языке Python набраны в кодировке</i></p> <p>А. cp1251</p> <p>Б. cp866</p> <p>В. UTF-8</p> <p>Г. UTF-16</p> <p>Ответ: В. UTF-8</p>
2	<p><i>Файл demo.py содержит следующий Python3 - код.</i></p> <pre data-bbox="341 1417 1110 1621">def напечатать(значение): print(значение) напечатать(значение=10)</pre> <p><i>При его выполнении</i></p> <p>А. интерпретатор выдаст ошибку, так как в именах функции и ее параметра используются нелатинские символы</p> <p>Б. интерпретатор выполнит код (напечатает в стандартный поток вывода значение «10»(без кавычек)), при этом этот код НЕ соответствует требованиям PEP-8</p> <p>В. интерпретатор выполнит код (напечатает в стандартный поток вывода значение «10» (без кавычек)), при этом этот код соответствует требованиям PEP-8</p> <p>Г. нет верного ответа</p> <p>Ответ: Б. интерпретатор выполнит код (напечатает в стандартный поток вывода значение</p>

	«10»(без кавычек)), при этом этот код НЕ соответствует требованиям PEP-8
3	<p><i>Python3-файл с расширением «.py» является</i></p> <p>А. модулем Б. пакетом В. проектом Г. нет верного ответа</p> <p>Ответ: А. модулем</p>
4	<p><i>Файл demo.py содержит следующий Python3 - код.</i></p> <pre>def Δ(φ): print(f'φ is {φ}')</pre> <p><i>Δ(3.14)</i></p> <p><i>При его выполнении</i></p> <p>А. интерпретатор выдаст ошибку, так как в именах функции и ее параметра используются нелатинские символы Б. интерпретатор выполнит код (напечатает в стандартный поток вывода строку «φ is 3.14» (без кавычек)), при этом этот код НЕ соответствует требованиям PEP-8 В. интерпретатор выполнит код (напечатает в стандартный поток вывода строку «φ is 3.14» (без кавычек)), при этом этот код соответствует требованиям PEP-8 Г. нет верного ответа</p> <p>Ответ: Б. интерпретатор выполнит код (напечатает в стандартный поток вывода строку «φ is 3.14» (без кавычек)), при этом этот код НЕ соответствует требованиям PEP-8</p>
5	<p><i>Если папка (директория) содержит файл с именем __init__.py, то Python будет интерпретировать ее как</i></p> <p>А. модуль Б. пакет В. проект Г. нет верного ответа</p> <p>Ответ: Б. пакет</p>
6	<p><i>Python файлы с расширением «.pyc» являются</i></p> <p>А. файлами исходных кодов Python-программ Б. файлами с байт-кодом Python загруженных модулей или программ В. исполняемыми бинарными файлами используемой операционной системы Г. нет верного ответа</p> <p>Ответ:</p>

Б. файлами с байт-кодом Python загруженных модулей или программ	
7	<p>Дан следующий Python3-код</p> <pre>s = "abcdef" s[1] = 'x'</pre> <p>При его выполнении</p> <p>А. интерпретатор выдаст ошибку Б. в строке s символ 'a' будет заменен на символ 'x' В. в строке s символ 'b' будет заменен на символ 'x' Г. нет верного ответа</p> <p>Ответ: А. интерпретатор выдаст ошибку</p>
8	<p>Следующий Python3-код</p> <pre>values = [val for val in range(10)]</pre> <p>использует</p> <p>А. списковые выражения (списковые включения) Б. генераторы списков В. циклы Г. нет верного ответа</p> <p>Ответ: А. списковые выражения (списковые включения)</p>
9	<p>Следующий Python3-код</p> <pre>values = (val for val in range(10))</pre> <p>использует</p> <p>А. списковые выражения (списковые включения) Б. генераторы списков В. циклы Г. нет верного ответа</p> <p>Ответ: Б. генераторы списков</p>

10	<p><i>Дан следующий Python3-код</i></p> <pre style="background-color: #f0f0f0; padding: 10px;">def foo(lst=[], value=None): lst.append(value) return lst print(foo(value=1)) print(foo(value=1)) print(foo(value=1))</pre> <p>В результате его выполнения в стандартный поток вывода будет выведено</p> <p>А. [] [] []</p> <p>Б. [1] [1] [1]</p> <p>В. [1] [1, 1] [1, 1, 1]</p> <p>Г. нет верного ответа</p> <p>Ответ: В. [1] [1, 1] [1, 1, 1]</p>
----	---

При оценивании тестовых заданий диагностической карты используются следующие критерии и шкала оценивания тестирования

Оценка/баллы	Критерии оценки
<i>Отлично</i>	90-100 % правильных ответов
<i>Хорошо</i>	70-89 % правильных ответов
<i>Удовлетворительно</i>	50-69 % правильных ответов
<i>Неудовлетворительно</i>	49% и меньше правильных ответов

Полученные баллы (за каждую компетенцию) равны целому (округленному до целого) проценту правильных ответов. Итоговый балл за выполнение диагностической работы равен сумме баллов, полученных за каждую компетенцию, умноженной на $1 / K$, где K – количество компетенций, проверяемых в диагностической работе.