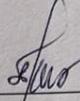


**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО
ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Дагестанский государственный университет»
Колледж

УТВЕРЖДАЮ

директор Колледжа

 Д.Ш. Пирбудагова

«14» 03 2022г.

Фонд оценочных средств
по учебной дисциплине

ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

09.02.07 Информационные системы и программирование

Махачкала -2022

Составитель:

Хангишиева Аида Хабибуллаевна – преподаватель кафедры общепрофессиональных дисциплин Колледжа ДГУ.

Фонд оценочных средств дисциплины рассмотрен и рекомендован к утверждению на заседании кафедры общепрофессиональных дисциплин Колледжа ДГУ

Протокол № 4 от « 12 » 03 2022 г.

Зав.кафедрой общепрофессиональных дисциплин Колледжа ДГУ.
к.ю.н., доцент В. Магомедов – Магомедова П. Р

Утверждена на заседании учебно-методического совета колледжа ДГУ

Ст. методист Шамсутдинова У.А. / Шамсутдинова У.А.
подпись Фамилия И.О.

ПАСПОРТ фонда оценочных средств

по дисциплине

Основы алгоритмизации и программирования

№	Контролируемые разделы, темы, модули	Код контролируемой компетенции	Наименование оценочного средства
1	Раздел 1. Основные принципы алгоритмизации и программирования	ОК 1; ОК 2; ОК 9; ОК 10.	тестирование; контрольная работа
2	Раздел 2. Язык программирования	ОК 4; ОК 9; ПК 1.1; ПК 1.2, ПК 2.4.	тестирование; контрольная работа
3	Раздел 3 Основы программирования в интегрированной среде	ОК 1; ОК 2; ОК 4; ПК 1.1; ПК 1.2.; ПК 1.3.	тестирование; контрольная работа
4	Раздел 4. Цикл в языке программирования Python	ОК 1; ОК 2; ОК 3; ОК 4; ОК 5; ОК 9; ОК 10, ПК 1.1; ПК 1.2.; ПК 1.3; ПК 1.4; ПК 1.5; ПК 2.4; ПК 2.5.	тестирование; контрольная работа
5	Раздел 5. Функции в Python	ОК 1; ОК 2; ОК 3; ОК 4; ОК 5; ОК 9; ОК 10, ПК 1.1; ПК 1.2.; ПК 1.3; ПК 1.4; ПК 1.5; ПК 2.4; ПК 2.5.	тестирование; контрольная работа
6	Раздел 6. Коллекции в Python.	ОК 1; ОК 2; ОК 3; ОК 4; ОК 5; ОК 9; ОК 10, ПК 1.1; ПК 1.2.; ПК 1.3; ПК 1.4; ПК 1.5; ПК 2.4; ПК 2.5.	тестирование; контрольная работа
7	Раздел 7. Библиотеки в Python.	ОК 1; ОК 2; ОК 3; ОК 4; ОК 5; ОК 9; ОК 10, ПК 1.1; ПК 1.2.; ПК 1.3; ПК 1.4; ПК 1.5; ПК 2.4; ПК 2.5.	тестирование; контрольная работа

Критерии оценки

по дисциплине

Основы алгоритмизации и программирования

№ п/п	Наименование оценочного средства	Критерии оценивания на «неудовлетв-но»	Критерии оценивания на «удовлетв-но»	Критерии оценивания на «хорошо»	Критерии оценивания на «отлично»
1	Коллоквиум	<p>студент обнаруживает незнание или непонимание большей или наиболее существенной части содержания учебного материала;</p> <p>не способен применять знание теории к решению задач профессионального характера;</p> <p>не умеет определить собственную оценочную позицию;</p> <p>допускает грубое нарушение логики изложения материала.</p> <p>допускает принципиальные ошибки в ответе на вопросы;</p> <p>не может исправить ошибки с помощью наводящих вопросов.</p>	<p>студент в основном знает программный материал в объёме, необходимом для предстоящей работы по профессии, но ответ, отличается недостаточной полнотой и обстоятельностью изложения:</p> <p>допускает существенные ошибки и неточности в изложении теоретического материала;</p> <p>в целом усвоил основную литературу;</p> <p>обнаруживает неумение применять государственно-правовые принципы, закономерности и категории для объяснения конкретных фактов и явлений;</p> <p>требуется помощь со стороны (путем наводящих вопросов, небольших разъяснений и т.п.);</p> <p>испытывает существенные трудности при определении собственной оценочной позиции;</p> <p>наблюдается нарушение логики изложения материала.</p>	<p>студент дает ответ, отличающийся меньшей обстоятельностью и глубиной изложения:</p> <p>обнаруживает при этом твёрдое знание материала;</p> <p>допускает несущественные ошибки и неточности в изложении теоретического материала;</p> <p>исправленные после дополнительного вопроса;</p> <p>опирается при построении ответа только на обязательную литературу;</p> <p>подтверждает теоретические постулаты отдельными примерами из юридической практики;</p> <p>способен применять знание теории к решению задач профессионального характера;</p> <p>наблюдается незначительное нарушение логики изложения материала.</p>	<p>студент дает полный и правильный ответ на поставленные и дополнительные (если в таковых была необходимость) вопросы;</p> <p>обнаруживает всестороннее системное и глубокое знание материала;</p> <p>обстоятельно раскрывает соответствующие теоретические положения;</p> <p>демонстрирует знание современной учебной и научной литературы;</p> <p>владеет понятийным аппаратом;</p> <p>демонстрирует способность к анализу и сопоставлению различных подходов к решению заявленной проблематики;</p> <p>подтверждает теоретические постулаты примерами из юридической практики;</p> <p>способен творчески применять знание теории к решению профессиональных задач;</p> <p>имеет собственную оценочную позицию и умеет аргументировано и убедительно ее раскрыть;</p> <p>четко излагает материал в логической последовательности.</p>
2	Тест	0% -50% правильных ответов – оценка «неудовлетворительно»	51% - 64% правильных ответов – оценка «удовлетворительно»	65% - 84% правильных ответов – оценка «хорошо»,	85% - 100% правильных ответов – оценка «отлично»
3	Контрольная	материал раскрыт не по	Тема письменной	Тема письменной	Работа соответствует

	работа	существо, допущены грубые ошибки в изложении и содержании теоретического материала; контрольная работа выполнена не по установленному варианту. работа скопирована из Интернет без ссылки на первоисточник	работы в целом раскрыта, но при этом допущена существенная ошибка или ответ неполный, несвязный, однако содержит некоторые обоснованные выводы, которые не в полной мере раскрывают тему	работы раскрыта полностью и правильно, на основании изученных теорий; материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки	заявленной теме, целям и задачам; характерна: - полнота и конкретность ответа; - последовательность и в изложении материала; - связь теоретических положений с практикой; - высокий уровень анализа и обобщения информационного материала, полноты обзора состояния вопроса; - обоснованность выводов.
4	Реферат	Обнаруживается лишь общее представление о теме либо тема не раскрыта полностью, работа скопирована из Интернет без ссылки на первоисточник	Вопрос раскрыт частично · Реферат написан небрежно, неаккуратно, использованы необщепринятые сокращения, затрудняющие ее чтение · Допущено 3–4 фактические ошибки	Вопрос раскрыт более чем наполовину, но без ошибок. Имеются незначительные и/или единичные ошибки. Используются ссылки менее чем на половину рекомендованных по данному вопросу источников права, · Допущены 1–2 фактические ошибки	Вопрос раскрыт полностью и без ошибок, реферат написан правильным литературным языком без грамматических ошибок в юридической терминологии, умело использованы ссылки на источники права
5	Рабочая тетрадь	Не имеются ответы ни на один из поставленных вопросов по теме	Вопросы раскрыты частично и неточно. Допущены ошибки	Вопросы раскрыты, но имеются незначительные неточности в ответах.	Ответы даны точно, полностью раскрывают вопрос. Даны ссылки на нормативно-правовые акты.
6	Решение задач	допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.	работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.	работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя	работа выполнена полностью и правильно; сделаны правильные выводы.
7	Лабораторные работы	Хотя бы одно задание выполнено неверно	Задание выполнено частично и содержит существенные ошибки	Задание выполнено полностью, но решение содержит несущественные ошибки	Правильно выполнены все задания

**Примерный перечень оценочных средств
по дисциплине «Основы алгоритмизации и программирования»**

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1	2	3	4
1	Коллоквиум	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования педагогического работника с обучающимися.	Вопросы по темам/разделам дисциплины
2	Рабочая тетрадь	Дидактический комплекс, предназначенный для самостоятельной работы обучающегося и позволяющий оценивать уровень усвоения им учебного материала.	Образец рабочей тетради
3	Решение задач	Проблемное задание, в котором обучающемуся предлагают осмыслить реальную профессионально-ориентированную ситуацию, необходимую для решения данной проблемы.	Комплект разноуровневых задач и заданий
4	Реферат	Продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее.	Темы рефератов
5	Тесты	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Комплекты тестов
6	Собеседование	Средство контроля, организованное как специальная беседа педагогического работника с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины
7	Лабораторные работы	В ходе выполнения лабораторных работ, рассматривается сложный программный комплекс модулей, разбирается архитектура, правила построения программ. Рассматривается и отрабатывается поэтапная реализация большого проекта.	Лабораторные работы
6	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект контрольных заданий по вариантам

**Комплект заданий для контрольной работы
по дисциплине «Основы алгоритмизации и программирования»**

Вариант 1

1. Понятие алгоритма. Свойства и формы записи алгоритмов
2. Операции в программировании
3. Работа с файловой системой средствами Python

Вариант 2

1. Методы и система алгоритмизации
2. Основные алгоритмические инструкции языка Python
3. Функция `readline()`, `write()`, метод `close()`, инструкция `with`

Вариант 3

1. Линейные алгоритмы. Ветвления в алгоритмах
2. Понятие идентификатор и правила их написания.
3. Методы файла в Python

Вариант 4

1. Разветвляющийся алгоритм. Простой условный оператор.
2. Основные типы данных
3. Встроенные типы последовательностей `list`, `tuple`, `range` и их методы.

Вариант 5

1. Циклические алгоритмы
2. Логический тип данных. Символьный тип данных. Операции с переменными этого типа.
3. Множества. Встроенные типы `set` и `frozenset`.

Вариант 6

1. Запись алгоритмов блок-схемами. Основные элементы блок-схем.
2. Целый и вещественный типы данных. Операции с переменными этого типа
3. Функции в Python. Основные понятия

Вариант 7

1. История развития языков программирования
2. `Print()` и форматирование вывода.
3. Методы работы со строками.

Вариант 8

1. Классификация языков программирования
2. Функции `input` и `print`. Назначение метода `format` и примеры его применения.
3. Встроенный тип `str`. Методы объекта `str`.

Вариант 9

1. Этапы решения задачи, выполняемые в процессе программирования.
2. Инструкции `if/else/elif`, логические операторы и выражения сравнения
3. Списки. Отличие от кортежей. Синтаксис объявления, срезы списков. Генерация списков.

Вариант 10

1. Методы и принципы программирования
2. Сокращенный условный оператор. Составной условный оператор.
3. Словарь. Понятия, синтаксис создания

Вариант 11

1. Особенности языка Python
2. Многозначные ветвления.
3. Библиотека Python

Вариант 12

1. Основные понятия алгоритмических языков программирования
2. Циклы while и for в Python
3. Понятие функции. Преимущества использования. Синтаксис создания функции.

Вопросы к экзамену: «Основы алгоритмизации и программирования».

1. Понятие алгоритма. Свойства и формы записи алгоритмов
2. Методы и система алгоритмизации
3. Линейные алгоритмы. Ветвления в алгоритмах
4. Разветвляющийся алгоритм. Простой условный оператор.
5. Циклические алгоритмы
6. Запись алгоритмов блок-схемами. Основные элементы блок-схем.
7. Основные понятия алгоритмических языков программирования
8. История развития языков программирования
9. Классификация языков программирования
10. Этапы решения задачи, выполняемые в процессе программирования.
11. Методы и принципы программирования
12. Особенности языка Python
13. Операции в программировании
14. Основные алгоритмические инструкции языка Python
15. Понятие идентификатор и правила их написания.
16. Основные типы данных
17. Логический тип данных. Символьный тип данных. Операции с переменными этого типа.
18. Целый и вещественный типы данных. Операции с переменными этого типа
19. Циклы while и for в Python
20. Print() и форматирование вывода.
21. Функции input и print. Назначение метода format и примеры его применения.
22. Инструкции if/else/elif, логические операторы и выражения сравнения
23. Сокращенный условный оператор. Составной условный оператор.
24. Многозначные ветвления.
25. Работа с файловой системой средствами Python
26. Функция readline(), write(), метод close(), инструкция with
27. Методы файла в Python
28. Встроенные типы последовательностей list, tuple, range и их методы.
29. Множества. Встроенные типы set и frozenset.
30. Функции в Python. Основные понятия
31. Понятие функции. Преимущества использования. Синтаксис создания функции.
32. Методы работы со строками.
33. Встроенный тип str. Методы объекта str.
34. Списки. Отличие от кортежей. Синтаксис объявления, срезы списков. Генерация списков.

35. Словарь. Понятия, синтаксис создания

36. Библиотека Python

**Комплект тестов (тестовых заданий)
по дисциплине**

«Основы алгоритмизации и программирования»

Тесты

1. Сколько всего базовых структур алгоритмов вы знаете?

- a) 3
- b) 4
- c) 2
- d) 1

2. Как называется определенная последовательность действий, которую нужно выполнить для решения конкретной задачи:

- a) алгоритм
- b) программа
- c) система команд исполнителя
- d) переменная

3. Как называется алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз:

- a) линейный алгоритм
- b) одиночный алгоритм
- c) не повторяющийся алгоритм
- d) циклический алгоритм

4. Какому критерию свойств алгоритмов относится решение целого класса однотипных задач?

- a) массовость
- b) конечность
- c) однозначность
- d) правильность

5. Свойство алгоритма, заключающиеся в том, что каждое действие и алгоритм в целом должны иметь возможность завершения, называется

- a) конечность
- b) дискретность
- c) детерминированность
- d) массовость

6. Свойство алгоритма, заключающиеся в том, что алгоритм должен состоять из конкретных действий, следующих в определенном порядке, называется

- a) дискретность
- b) детерминированность
- c) конечность
- d) массовость

7. Свойство алгоритма, заключающиеся в отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях, называется

- a) результативность
- b) детерминированность
- c) конечность

- d) массовость
8. Свойство алгоритма, заключающиеся в том, что один и тот же алгоритм можно использовать с разными исходными данными, называется
- a) массовость
 - b) дискретность
 - c) детерминированность
 - d) результативность
9. Свойство алгоритма, заключающиеся в том, что любое действие должно быть строго и недвусмысленно определено в каждом случае, называется
- a) детерминированность
 - b) дискретность
 - c) конечность
 - d) массовость
10. Алгоритм, записанный на «понятном» компьютеру языке программирования, называется
- a) программой
 - b) исполнителем алгоритмов
 - c) листингом
 - d) протоколом алгоритма
11. Языком программирования называется
- a) совокупность средств и правил представления алгоритма в виде, пригодном для выполнения вычислительной машиной
 - b) совокупность средств и правил перевода текста с естественного языка на формальный.
 - c) совокупность средств и правил перевода текста с формального языка на естественный.
 - d) язык, понятный программистам
 - e) язык, понятный компьютеру
12. Система программирования – это:
- a) программное обеспечение компьютера, предназначенное для разработки, отладки и исполнения программ, записанных на определённом языке программирования
 - b) устройство для создания компьютерных программ
 - c) специальная программа, предназначенная для создания компьютерных программ
 - d) операционная система компьютера
13. Тип переменной во время выполнения скрипта определяется по следующим правилам:
- a) тип переменной изменяется при присваивании, но не может изменяться в зависимости от контекста использования этой переменной
 - b) тип переменной явно указывается при определении переменной и не изменяется в процессе выполнения скрипта
 - c) тип переменной явно указывается при определении переменной и изменяется только при приведении этой переменной к другому типу
 - d) тип переменной определяется типом первого значения, которое было ей присвоено, и далее не изменяется
 - e) тип переменной изменяется при приведении этой переменной к другому типу, а также может изменяться в зависимости от контекста использования этой переменной
14. Инструкция for в языке Python:

- a) перебирает любую последовательность в порядке следования элементов
- b) всегда перебирает элементы арифметической прогрессии, причем шаг и условия выхода задаются пользователем
- c) всегда перебирает элементы арифметической прогрессии, причем шаг и условия выхода фиксированные
- d) предоставляет полный выбор перебираемой последовательности, итератора и условий выхода пользователю
- e) перебирает любую последовательность в порядке, заданным пользователем

15. Встроенный метод списка `index()` используется для:

- a) получения индекса первого элемента списка с указанным значением
- b) получения индекса последнего элемента списка с указанным значением
- c) получения списка индексов элементов списка с указанным значением
- d) получения значения элемента списка по индексу

16. Встроенный метод списка `remove()` используется для:

- a) удаления первого элемента списка с указанным значением
- b) удаления элемента списка с указанным индексом
- c) удаления последнего элемента списка с указанным значением
- d) удаления всех элементов списка с указанным значением

17. В языке Python строковые литералы задаются:

- a) кавычками, одинарными кавычками и этими символами, повторенными трижды (`"""`, `'''`)
- b) одинарными кавычками (`'`)
- c) кавычками и кавычками, повторенными трижды (`"""`)
- d) одинарными кавычками и одинарными кавычками, повторенными трижды (`'''`)

18. Что будет в результате работы программы:

```
a=input()
```

```
b=input()
```

```
c=a+b
```

```
print (c)
```

Входные данные a=21 b=33

- a) 2133
- b) 54
- c) 21
- d) 33

19. Компонент интегрированной среды разработки приложений, который позволяет превратить листинг программы в исполняемый код, называется:

- a) компилятор
- b) текстовый редактор
- c) средства сборки
- d) отладчик

20. Какого типа значение получится в результате вычисления следующего выражения: `(r'\u0589')`

- a) `str` (строка)
- b) `unicode` (Unicode-строка)
- c) `tuple` (кортеж)
- d) это синтаксическая ошибка

21. Среди приведенных ниже фрагментов укажите все варианты кода, при подстановке которого вместо знаков подчеркивания результатом выполнения скрипта станет строка 8:

```
def foo (a=2, b=3, *c):
```

```
    res=a+b
```

```
    for i in c:
```

```
        res=res+i
```

```
    print (res)
```

a) `foo(4,b=4)`

b) `foo(2,3,3)`

c) `foo(a=3,b=3,2)`

d) скрипт не будет выполнен, т.к. код содержит ошибки

22. Какие парадигмы и стили программирования поддерживает Python ?

a) модульный

b) функциональное программирование

c) программирование в ограничениях

d) логистическое программирование

e) структурный стиль

23. Какие встроенные функции возвращают в качестве значения словарь?

a) `locals()`

b) `vars()`

c) `dir()`

d) `enumerate()`

e) `reload()`

f) `isinstance()`

24. Какие встроенные функции служат для создания объектов определенного типа (являются конструкторами классов или функциями-фабриками)?

a) `dict()`

b) `tuple()`

c) `chr()`

d) `isinstance()`

e) `locals()`

f) `vars()`

25. Какими способами можно получить доступ к функции `fun()` из модуля `fns`?

a) `from fns import fun; fun()`

b) `import fns; fns.fun()`

c) `import fns.fun; fns.fun()`

d) `import fns.fun as fun; fun()`

e) `import fns.fun; fun()`

f) `import fns.fun; fun()`

26. Как определить функцию в Python?

a) с помощью оператора `def`

b) с помощью `lambda`-выражения

c) с помощью оператора `import`

d) заданием списка строк исходного кода

e) с помощью функции `map()`

f) с помощью функции `iter()`

27. Дана функция-генератор:

```
def ggg(s):  
    for i in range(0, len(s)-1, 2):  
        yield s[i], s[i+1]
```

27. Какие из перечисленных ниже способов ее использования для вывода значений правильные?

- a) `for k in ggg([1, 2, 3, 4]): print k`
- b) `print list(ggg("asdf"))`
- c) `print iter(ggg("asdf"))`
- d) `print ggg("asdf")`
- e) `print gg("asdf")`
- f) `print list("as")`

28. Какие варианты использования значений генераторного выражения `gg` имеют смысл?

```
gg = ((s[i], s[i+1]) for i in range(0, len(s)-1, 2))
```

- a) `print list(gg)`
- b) `d = dict(gg)`
- c) `print gg`
- d) `print len(gg)`
- e) `print len`
- f) `print (gg)`

29. Как можно найти минимальный элемент матрицы `a`?

- a) `minimum.reduce(minimum.reduce(a))`
- b) `a[argmin(a)]`
- c) `min(a)`
- d) `minimum(a)`
- e) `max(a)`

30. С помощью какой универсальной функции (возможно, в сочетании с функцией `sum()`) можно посчитать количество отрицательных чисел в массиве?

- a) `less()`
- b) `greater()`
- c) `count()`
- d) `sign()`
- e) `tuple`
- f) `executemany()`

31. Охарактеризуйте методы `keys()`, `items()`, `values()`. Что они возвращают, какова специфика результирующих объектов?

Методы `keys()`, `items()`, `values()` еще называют объектами `views` (объекты просмотра словаря). Они позволяют просматривать все элементы словаря и имеют некоторые особенности:

Ответ:

1. `keys()` – возвращает список ключей в исходном порядке, все элементы уникальны, ведут себя как множество (`set`): могут объединяться, пересекаться и т.п., являются итерабельными, их нельзя менять напрямую и вызывать по индексу

2. `values()` – контейнер со значениями словаря, в котором элементы могут повторяться (т.е. не ведет себя как множество). В остальном, свойства аналогичны методу `keys()`

3. `items()` – список элементов словаря в виде кортежей, где первым идет ключ, а вторым объектом – значение. Если значения словаря неизменяемые, то контейнер `items()` будет обладать свойствами сета. Остальные особенности совпадают с `keys()`.

32. Перечислите все типы данных, которые потенциально можно преобразовать в целое число напрямую через функцию `int()`.

Ответ:

К целым числам можно привести:

- целые числа
- числа с плавающей точкой
- булевы значения
- десятичные дроби
- дробные числа
- строки

33. Какими способами можно открыть текстовый файл (в формате `.txt`) в Python (без использования сторонних библиотек)?

Ответ:

Существует 2 основных варианта открытия текстового документа:

При помощи функции `open()` – дополнительно требуется закрыть файл после работы с ним, иначе он останется в памяти; С использованием контекстного менеджера `with` – закрывать документ не требуется, так как это произойдет автоматически.

34. Сопоставьте определения:

1. Способность объекта скрывать внутреннее устройство своих свойств и методов, называется

2. Создание новых классов на основе существующих, с использованием их структурных и поведенческих характеристик, называется

3. Компонент интегрированной среды разработки приложений, который позволяет превратить листинг программы в исполняемый код, называется:

- a) Компилятор
- b) Наследование
- c) Инкапсуляция

Ответ:

1-с

2-б

3-а

Задания открытого типа

35. Как используют функцию `print()` для записи файлов?

36. Чем отличаются методы `append()` и `extend()`?

36. Дано 2 скрипта. В первом – возникает ошибка, во втором – ошибки нет.

Поясните почему.

Скрипт 1.

```
a = None
```

```
if len(a) > 0 and a is not None:
```

```
    print('OK')
```

Скрипт 2.

```
a = None
```

if a is not None and len(a) > 0:

```
    print('OK')
```

38. Напишите функцию `sum_range(start, end)`, которая суммирует все целые числа от значения «start» до величины «end» включительно. Если пользователь задаст первое число большее чем второе, просто поменяйте их местами.

39. Антонина в целях экономии памяти компьютера вместо последовательности чисел от 1 до миллиона в виде списка решила создать ленивый итератор `numbers` с помощью функции `range()`. Затем в двух разных местах своего скрипта она делала проверку того, есть ли число 2000 в этой коллекции.

В первом случае код вернул `True`, а во втором `False`

Объясните ученице ее ошибку.

Код – IDE

```
--
```

```
numbers = iter(range(1, 1000001))
```

```
print(2000 in numbers)
```

```
print(2000 in numbers)
```

Результат выполнения

```
True
```

```
False
```

40. Что произойдет со списком `lst1` в первом и втором случаях? Поясните результат.

Случай 1 – IDE

```
lst1 = [1, 2, 3, 14, 33, 1, 9]
```

```
lst2 = [1, 2, 3, 14, 33, 1, 9]
```

```
lst2.append(789)
```

Случай 2 – IDE

```
lst1 = [1, 2, 3, 14, 33, 1, 9]
```

```
lst2 = lst1
```

```
lst2.append(789)
```

41. Сколько элементов будет содержать список `L` (то есть, чему будет равно `len(L)`) после выполнения следующего кода:

```
L = []
```

```
L.extend([1,2,3])
```

```
L.insert(1, "abc")
```

```
del L[0][0]
```

42. Что будет в результате выполнения программы (чему равно `a` и `b`):

```
a=5
```

```
b=a+2
```

```
a=(a+2)*(b-3)
```

```
b=b+1
```

43. Николай написал функцию `is_alive(health)`, которая проверяет здоровье персонажа в игре.

Если оно равно или меньше нуля, то функция возвращает `False`, в противном случае `True`.

К сожалению, функция не работает, так как ученик допустил в ней ряд ошибок.

Исправьте их.

```
def is_alive(health):
```

```
    if:
```

```
health < 0
False
else:
    return true
```

44. Определите структуру, которую имеет представленный алгоритм:

```
ввод значения x;
ЕСЛИ  $x \geq 0$ , ТОГДА у присвоить x в квадрате, ИНАЧЕ у присвоить x в кубе;
вывод значения у.
```

45. Определите структуру, которую имеет представленный алгоритм:

```
ввод значения x;
ПОКА  $x \leq 100$  умножать x на 2;
вывод значения у.
```

46. Что такое регулярное выражение?

47. Какого типа значение получится в результате вычисления следующего выражения:
(r'\u0432')

48. Что будет получено в результате вычисления следующего выражения:
 $(0 < [1, 4][1] < 3) + 1$

49. Что будет получено в результате вычисления следующего выражения:
 $0 < [1, 4], [1] < 3$

50. Что будет выведено следующей программой:

```
a = 1
b = 2
a, b = b, a
print a, b, a
```

51. Что будет выведено следующей программой:

```
a = "A"
b = "B"
b = b + a
print a + b
```

52. Что будет выведено следующей программой:

```
a = "AB"
b = "BC"
print "%sa, b" % a, b
```

53. Что будет выведено в следующем примере:

```
>>> a = [1, 2, 3]
>>> a[1] = a[1:2]
>>> print a[1]
```

54. Что выведет следующая программа

```
S = 0
for i in range(1, 10):
    if i % 2 == 0:
        S = S + i
print S
```

55. Что выведет следующая программа:

```
S = 0
for i in range(1, 10, 2):
```

```
if i % 2 == 0:  
    S = S + i
```

```
print S
```

56. Что выведет следующая программа:

```
S = 0
```

```
for i in range(10, 2, -1):
```

```
    if i % 2 == 0:  
        S = S + i
```

```
print S
```

57. Сколько элементов будет содержать список L (то есть, чему будет равно len(L)) после выполнения следующего кода:

```
L = []
```

```
L.append([1,2,3])
```

```
L.insert(1, "abc")
```

```
del L[0][0]
```

58. Что выведет следующая программа:

```
S = 0
```

```
for i in range(1, 10):
```

```
    for j in range(1, 3):
```

```
        if i == j:
```

```
            S = S + i + j
```

```
break
```

```
print S
```

59. Сколько элементов будет содержать словарь D (то есть, чему будет равно len(D)) после выполнения следующего кода:

```
D = { }
```

```
D[1], D[2], D[3] = "ABV"
```

```
D[0], D[1] = "AB"
```

60. Что будет выведено следующей программой:

```
A = B = 1
```

```
a = "A" * 5
```

```
b = "B" * 5
```

```
a = a + b * 5
```

```
print a, b
```

Основная литература

1 Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2021. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. —

URL: <https://urait.ru/bcode/473347>

2 Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2021. — 137 с. — (Высшее образование). — ISBN 978-5-534-07834-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. —

URL: <https://urait.ru/bcode/471125>

3 Чернышев, С. А. Основы программирования на Python : учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва : Издательство Юрайт, 2021. — 286 с. — (Профессиональное образование). — ISBN 978-5-534-15160-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/487638>

б) дополнительная литература:

1 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с. — (Профессиональное образование). — ISBN 978-5-534-12829-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/487079>

2 Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472502>

3 Васильев А.Н. Python на примерах [Электронный ресурс]: практический курс по программированию/ Васильев А.Н.— Электрон. текстовые данные — Санкт-Петербург: Наука и Техника, 2017. — 432 с.— Режим доступа: <http://www.iprbookshop.ru/73043.html>. — ЭБС «IPRbooks»

4 Сузи Р.А. Язык программирования Python [Электронный ресурс]: учебное пособие/ Сузи Р.А.— Электрон. текстовые данные — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 с.— Режим доступа: <http://www.iprbookshop.ru/97589.html>. — ЭБС «IPRbooks»

5 Шелудько В.М. Основы программирования на языке высокого уровня Python [Электронный ресурс]: учебное пособие/ Шелудько В.М.— Электрон. текстовые данные — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. — 146 с.— Режим доступа: <http://www.iprbookshop.ru/87461.html>.— ЭБС «IPRbooks»

Перечень ресурсов информационно-телекоммуникационной сети

«Интернет», необходимых для освоения дисциплины.

1. 1. Портал доступа к электронным образовательным ресурсам ДГУ [Электронный ресурс] Режим доступа URL: <http://dgu.ru>

2. Электронно-библиотечная система IPRbooks [Электронный ресурс] Режим доступа URL: <http://www.iprbookshop.ru>

3. Электронно-библиотечная система «Университетская библиотека он-лайн» [Электронный ресурс] Режим доступа URL: <http://biblioclub.ru>

4. Электронно-библиотечная система издательства «Инфра» [Электронный ресурс] Режим доступа URL: <http://znaniyum.com>

5. IT-портал [Электронный ресурс] Режим доступа URL: <http://citforum.ru>

6. Портал Национального открытого университета «Интуит» [Электронный ресурс] Режим доступа URL: <http://www.intuit.ru>

7. Техническая документация фирмы Microsoft [Электронный ресурс] Режим доступа URL: <http://technet.microsoft.com/ru-ru/sysinternals>

8. Moodle [Электронный ресурс]: система виртуального обучения: [база данных] / Даг. гос. ун-т. – Махачкала, г. – Доступ из сети ДГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. – Режим доступа URL: <http://moodle.dgu.ru/>

9. Учебный курс по программированию «Учите Питон» [Электронный ресурс] — Режим доступа: <http://pythontutor.ru>

10. Платформа учебных курсов Stepik [Электронный ресурс] — Режим доступа URL: <https://stepik.org/catalog?q=Python>

