

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Экономический институт
Кафедра экономической кибернетики и прикладной статистики

УТВЕРЖДАЮ

Директор

Мор Е.С.

« 28 » _____ 20 25 года



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

Бизнес-информатика-2

(наименование учебной дисциплины, практики)

38.03.05 Бизнес-информатика

(код и наименование направления подготовки (специальности))

Экономическая аналитика и бизнес-статистика

(наименование профиля подготовки (специальности, магистерской программы); при отсутствии ставится прочерк)

Разработчик (разработчики):

доцент

Велигура А.В.

ФОС рассмотрен и одобрен на заседании кафедры экономической кибернетики и прикладной статистики от « 25 » _____ 20 25 г., протокол № 25

Заведующий кафедрой экономической кибернетики и прикладной статистики

Велигура А.В.

Луганск 2025 г.

**Комплект оценочных материалов по дисциплине
«Бизнес-информатика 2»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. Выберите один правильный ответ

Какая конструкция используется для многократного выполнения кода в Python?

Варианты ответов:

- А) if
- Б) for
- В) while
- Г) else

Правильный ответ: Б.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

2. Выберите один правильный ответ

Каким образом можно преобразовать число в строку в Python?

Варианты ответов:

- А) str(x)
- Б) to_string(x)
- В) convert_to_string(x)
- Г) as_string(x)

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

3. Выберите один правильный ответ

Какая функция используется для генерации случайных чисел в Python?

Варианты ответов:

- А) random()
- Б) rand()
- В) generate_random()
- Г) rnd()

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

4. Выберите один правильный ответ

Как в Python обозначается булевый тип данных?

Варианты ответов:

- А) bool
- Б) boolean

В) true_false

Г) logic

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

5. Выберите один правильный ответ

Как можно объединить два списка в Python?

Варианты ответов:

А) append()

Б) join()

В) extend()

Г) merge()

Правильный ответ: В.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

6. Выберите один правильный ответ

Какая библиотека предоставляет возможность работы с датой и временем в Python?

Варианты ответов:

А) datetime

Б) time

В) date_time

Г) clock

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

7. Выберите один правильный ответ

Какое ключевое слово используется для выхода из цикла в Python?

Варианты ответов:

А) break

Б) exit

В) stop

Г) leave

Правильный ответ: А.

Компетенция (индикатор): ПК-3 (ПК-3.3).

8. Выберите один правильный ответ

Каким образом можно создать словарь в Python?

Варианты ответов:

А) {key: value}

Б) dict()

В) create_dict()

Г) make_dict()

Правильный ответ: Б.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

9. Выберите один правильный ответ

Какая функция используется для чтения данных из файла в Python?

Варианты ответов:

А) read()

Б) open()

В) file_read()

Г) input_file()

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

10. Выберите один правильный ответ

Какой оператор используется для объединения (конкатенации) двух строк в Python?

Варианты ответов:

А) +

Б) &

В) ||

Г) .

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

11. Выберите все правильные варианты ответов

Какие из перечисленных операторов являются арифметическими операторами в Python?

Варианты ответов:

А) +

Б) –

В) >

Г) ==

Правильный ответ: А, Б.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

12. Выберите все правильные варианты ответов

Какие из этих функций возвращают значение True, если переданный аргумент является числом?

Варианты ответов:

А) isdigit()

Б) isnumeric()

В) isspace()

Г) isalpha()

Правильный ответ: А, Б.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

13 Выберите все правильные варианты ответов

Какие из следующих операторов сравнения используются для проверки неравенства значений?

Варианты ответов:

- А) !=
- Б) <
- В) >=
- Г) <=

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

14. Выберите все правильные варианты ответов

Какие из указанных модулей входят в стандартную библиотеку Python?

Варианты ответов:

- А) math
- Б) os
- В) sys
- Г) json

Правильный ответ: А, Б, В.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

15. Выберите все правильные варианты ответов

Какие из перечисленных операций выполняются над кортежами в Python?

Варианты ответов:

- А) Индексирование
- Б) Изменение элемента
- В) Добавление нового элемента
- Г) Удаление элемента

Правильный ответ: А, Г.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

16. Выберите все правильные варианты ответов

Какие из данных терминов относятся к принципам программирования?

Варианты ответов:

- А) Повторное использование кода
- Б) Модульность
- В) Параллельность
- Г) Масштабируемость

Правильный ответ: А, Б.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

17. Выберите все правильные варианты ответов

Какие из приведенных команд используются для открытия файла в Python?

Варианты ответов:

- А) open()
- Б) fopen()

В) read()

Г) write()

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

18. Выберите все правильные варианты ответов

Какие из этих функций позволяют импортировать модули в Python?

Варианты ответов:

А) import()

Б) require()

В) load()

Г) use()

Правильный ответ: А.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

19. Выберите все правильные варианты ответов

Какие из следующих типов данных являются неизменяемыми в Python?

Варианты ответов:

А) tuple

Б) string

В) int

Г) float

Правильный ответ: А, Б.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

20. Выберите все правильные варианты ответов

Какие методы списков позволяют добавлять элементы в конец списка?

Варианты ответов:

А) append()

Б) insert()

В) extend()

Г) add()

Правильный ответ: А, В.

Компетенции (индикаторы): ПК-3 (ПК-3.3).

Задания закрытого типа на установление соответствия

1. Установите соответствие между типами данных и их примерами. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Типы данных

1) int

2) list

3) dict

4) bool

Примеры значений

А) [1, "текст", 3.5]

Б) {"имя": "Анна", "возраст": 25}

В) 15

Г) True

Правильный ответ: 1-В, 2-А, 3-Б, 4-Г

Компетенции (индикаторы): ПК-3 (ПК-3.3)

2. Установите соответствие между свойствами алгоритмов и их описаниями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Свойства алгоритмов	Описания
1) Дискретность	А) Каждая команда интерпретируется однозначно.
2) Определенность	Б) Решение применимо к классу однотипных задач.
3) Конечность	В) Алгоритм завершается за конечное число шагов.
4) Массовость	Г) Алгоритм разбит на отдельные элементарные действия.

Правильный ответ: 1-Г, 2-А, 3-В, 4-Б

Компетенции (индикаторы): ПК-3 (ПК-3.3)

3. Установите соответствие между операторами и их функциями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Операторы	Функции
1) if	А) Цикл с предусловием.
2) for	Б) Условное выполнение кода.
3) while	В) Прерывание цикла.
4) break	Г) Цикл по элементам последовательности.

Правильный ответ: 1-Б, 2-Г, 3-А, 4-В

Компетенции (индикаторы): ПК-3 (ПК-3.3)

4. Установите соответствие между принципами ООП и их определениями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Принципы	Определения
1А) Инкапсуляция	А1) Возможность объектов с одинаковым интерфейсом иметь разную реализацию
2Б) Наследование	Б2) Соккрытие внутренней реализации объекта.
3В) Полиморфизм	В3) Создание новых классов на основе существующих.

Правильный ответ: 1-Б, 2-В, 3-А

Компетенции (индикаторы): ПК-3 (ПК-3.3)

5. Установите соответствие между режимами открытия файла и их описаниями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Режимы	Описания
1) "r"	А) Чтение файла в бинарном режиме.
2) "w"	Б) Перезапись файла.
3) "a"	В) Чтение файла.
4) "rb"	Г) Добавление данных в конец файла.

Правильный ответ: 1-В, 2-Б, 3-Г, 4-А

Компетенции (индикаторы): ПК-3 (ПК-3.3)

6. Установите соответствие между методами списков и их действиями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Методы	Действия
1) append()	А) Удаляет последний элемент.
2) pop()	Б) Добавляет элемент в конец.
3) sort()	В) Изменяет порядок элементов на обратный.
4) reverse()	Г) Сортирует элементы.

Правильный ответ: 1-Б, 2-А, 3-Г, 4-В

Компетенции (индикаторы): ПК-3 (ПК-3.3)

7. Установите соответствие между этапами и их описаниями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Этапы	Описания
1) Постановка задачи	А) Поиск и исправление ошибок.
2) Тестирование	Б) Формализация требований.
3) Отладка	В) Проверка корректности работы программы.
4) Документирование	Г) Создание инструкций для пользователей.

Правильный ответ: 1А-Б, 2-В, 3-А, 4-Г

Компетенции (индикаторы): ПК-3 (ПК-3.3)

8. Установите соответствие между понятиями и их определениями. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Понятия	Определения
1) Функция	А) Набор связанных функций и классов.
2) Модуль	Б) Шаблон для создания объектов.
3) Библиотека	В) Отдельный файл с кодом.

4) Класс

Г) Блок кода, выполняющий определенную задачу.

Правильный ответ: 1-Г, 2-В, 3-А, 4-Б

Компетенции (индикаторы): ПК-3 (ПК-3.3)

9. Установите соответствие между операциями и их результатами. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Операции	Результаты
1) "Hello".lower()	А) True
2) "123".isdigit()	Б) "yth"
3) "text".replace("t", "T")	В) "hello"
4) Python"[1:4]	Г) "TeXt"

Правильный ответ: 1-В, 2-А, 3-Г, 4-Б

Компетенции (индикаторы): ПК-3 (ПК-3.3)

10. Установите соответствие между элементами ООП и их примерами. Каждому элементу левого столбца соответствует только один элемент правого столбца:

Элементы ООП	Примеры
1) Родительский класс	А) class Dog(Animal):
2) Дочерний класс	Б) def bark(self):
3) Атрибут	В) self.color = "черный"
4) Метод	Г) class Animal:

Правильный ответ: 1-Г, 2-А, 3-В, 4-Б

Компетенции (индикаторы): ПК-3 (ПК-3.3)

Задания закрытого типа на установление правильной последовательности

1. Установите правильную последовательность. Запишите правильную последовательность букв слева направо.

Менеджер проекта получил задачу провести анализ нового продукта перед его выводом на рынок. Он должен следовать четкому процессу бизнес-анализа.

- А) Проведение фокус-групп с клиентами.
- Б) Составление отчета по результатам анализа.
- В) Постановка целей и задач анализа.
- Г) Сбор и анализ данных.
- Д) Презентация выводов руководству.

Правильный ответ: В, Г, А, Б, Д

Компетенции (индикаторы): ПК-3 (ПК-3.3)

2. Программист приступает к созданию приложения. Установите правильную последовательность этапов разработки. Запишите правильную последовательность букв слева направо.

А) Кодирование алгоритма.

Б) Постановка задачи.

В) Анализ результатов.

Г) Создание алгоритма.

Д) Построение модели.

Правильный ответ: Б, Д, Г, А, В

Компетенции (индикаторы): ПК-3 (ПК-3.3)

3. Установите последовательность действий для обработки списка. Запишите правильную последовательность букв слева направо:

А) Удаление элемента по индексу.

Б) Инициализация списка.

В) Сортировка элементов.

Г) Добавление элемента в конец.

Д) Вывод списка на экран.

Правильный ответ: Б, Г, В, А, Д

Компетенции (индикаторы): ПК-3 (ПК-3.3)

4. Установите последовательность операций при работе с файлом. Запишите правильную последовательность букв слева направо:

А) Закрытие файла.

Б) Открытие файла в режиме записи.

В) Чтение данных из файла.

Г) Запись данных в файл.

Д) Обработка ошибок доступа.

Правильный ответ: Б, Г, А, Д

Компетенции (индикаторы): ПК-3 (ПК-3.3)

5. Установите последовательность этапов решения задачи с помощью ЭВМ. Запишите правильную последовательность букв слева направо:

А) Анализ результатов.

Б) Постановка задачи.

В) Кодирование алгоритма.

Г) Создание алгоритма.

Д) Построение модели.

Правильный ответ: Б, Д, Г, В, А

Компетенции (индикаторы): ПК-3 (ПК-3.3)

6. Установите последовательность создания класса в Python. Запишите правильную последовательность букв слева направо:

А) Определение конструктора `__init__`.

Б) Объявление класса.

В) Создание экземпляра класса.

Г) Добавление методов.

Д) Вызов метода объекта.

Правильный ответ: Б, А, Г, В, Д

Компетенции (индикаторы): ПК-3 (ПК-3.3)

7. Установите последовательность цикла обработки данных. Запишите правильную последовательность букв слева направо:

А) Валидация входных данных.

Б) Ввод данных.

В) Вывод результатов.

Г) Обработка данных.

Д) Сохранение в файл.

Правильный ответ: Б, А, Г, В, Д

Компетенции (индикаторы): ПК-3 (ПК-3.3)

8. Установите последовательность работы с модулями. Запишите правильную последовательность букв слева направо:

А) Импорт модуля в основной код.

Б) Создание функций в модуле.

В) Вызов функции из модуля.

Г) Сохранение модуля в отдельный файл.

Правильный ответ: Г, Б, А, В

Компетенции (индикаторы): ПК-3 (ПК-3.3)

9. Расположите фрагменты кода для поиска максимального элемента в списке в правильном порядке. Запишите правильную последовательность букв слева направо:

А) `max_num = numbers[0]`

Б) `numbers = [15, 3, 27, 9, 12]`

В) `for num in numbers:`

Г) `if num > max_num:`

Д) `max_num = num`

Правильный ответ: Б, А, В, Г, Д

Компетенции (индикаторы): ПК-3 (ПК-3.3)

10. Расположите в правильном порядке фрагменты кода преобразования чисел в строку с разделителем. Запишите правильную последовательность букв слева направо

А) `numbers = [10, 20, 30, 40]`

Б) `str_numbers = [str(num) for num in numbers]`

В) `result = " | ".join(str_numbers)`

Г) `print(result)`

Правильный ответ: А, Б, В, Г

Компетенции (индикаторы): ПК-3 (ПК-3.3)

Задания открытого типа

Задания открытого типа на дополнение

1. *Напишите пропущенное слово (словосочетание).*

Свойство алгоритма, означающее, что он состоит из отдельных шагов, которые выполняются последовательно, называется _____.

Правильный ответ: дискретность

Компетенции (индикаторы): ПК-3 (ПК-3.3)

2. *Напишите пропущенное слово (словосочетание).*

Тип данных, который используется для хранения неупорядоченных пар «ключ-значение», называется _____.

Правильный ответ: словарь / dict

Компетенции (индикаторы): ПК-3 (ПК-3.3)

3. *Напишите пропущенное слово (словосочетание).*

Оператор _____ в Python позволяет прервать выполнение цикла до его завершения.

Правильный ответ: break

Компетенции (индикаторы): ПК-3 (ПК-3.3)

4. *Напишите пропущенное слово (словосочетание).*

Принцип ООП, при котором данные и методы их обработки объединены в одном объекте, называется _____.

Правильный ответ: инкапсуляция

Компетенции (индикаторы): ПК-3 (ПК-3.3)

5. *Напишите пропущенное слово (словосочетание).*

Режим _____ используется для открытия файла в Python, если нужно добавить данные в конец файла.

Правильный ответ: "a"

Компетенции (индикаторы): ПК-3 (ПК-3.3)

6. *Напишите пропущенное слово (словосочетание).*

Метод списка, который удаляет последний элемент и возвращает его, называется _____.

Правильный ответ: pop()

Компетенции (индикаторы): ПК-3 (ПК-3.3)

7. *Напишите пропущенное слово (словосочетание).*

Метод "text". _____ ("t", "T") заменит все вхождения символа «t» на «T».

Правильный ответ: replace

Компетенции (индикаторы): ПК-3 (ПК-3.3)

8. *Напишите пропущенное слово (словосочетание).*

Строка `def` в Python используется для _____ функции.

Правильный ответ: объявления/определения

Компетенции (индикаторы): ПК-3 (ПК-3.3)

9. *Напишите пропущенное слово (словосочетание).*

Файл с расширением `.py`, содержащий набор функций и классов, называется _____.

Правильный ответ: модуль

Компетенции (индикаторы): ПК-3 (ПК-3.3)

10. *Напишите пропущенное слово (словосочетание)*

Метод _____ в Python вызывается автоматически при создании экземпляра класса.

Правильный ответ: `__init__`

Компетенции (индикаторы): ПК-3 (ПК-3.3)

Задания открытого типа с кратким свободным ответом

1. *Напишите ответ на вопрос.* В какой строке допущена синтаксическая ошибка?

```
1 sum = 0
2 numbers = [1, 2, 3, 4]
3 for num in numbers:
4     sum += num
5     print("Сумма:", sum)
```

Правильный ответ: Строка 5/5.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

2. *Напишите ответ на вопрос.* Каким будет результат выполнения программы?

```
1 text = "Hello, World!"
2 new_text = text.replace("o", "1")
3 print(new_text)
```

Правильный ответ: Hell1, W1rld!

Компетенции (индикаторы): ПК-3 (ПК-3.3)

3. *Напишите ответ на вопрос.* Что выведет программа?

```
1 my_list = [1, 2, 3]
2 my_list.append([4, 5])
3 print(len(my_list))
```

Правильный ответ: 4

Компетенции (индикаторы): ПК-3 (ПК-3.3)

4. *Напишите ответ на вопрос.* Что будет выведено?

```

1 x = 10
2 if x > 5 and x < 15:
3     print("A")
4 elif x == 10:
5     print("B")
6 else:
7     print("C")

```

Правильный ответ: A

Компетенции (индикаторы): ПК-3 (ПК-3.3)

5. *Напишите ответ на вопрос. Почему функция возвращает None?*

```

1 def add(a, b):
2     result = a + b
3
4 print(add(2, 3))

```

Правильный ответ: Отсутствует return result

Компетенции (индикаторы): ПК-3 (ПК-3.3)

6. *Напишите ответ на вопрос. Какое исключение возникнет при выполнении?*

```

1 try:
2     print(10 / 0)
3 except ValueError:
4     print("Ошибка!")

```

Правильный ответ: ZeroDivisionError

Компетенции (индикаторы): ПК-3 (ПК-3.3)

7. *Напишите ответ на вопрос. Какая строка кода отсутствует, что может привести к утечке ресурсов?*

```

1 file = open("data.txt", "r")
2 content = file.read()

```

Правильный ответ: file.close()

Компетенции (индикаторы): ПК-3 (ПК-3.3)

8. *Напишите ответ на вопрос. В чем ошибка?*

```

1 import maths
2 print(maths.sqrt(16))

```

Правильный ответ: Неправильное имя модуля: maths вместо math.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

9. *Напишите ответ на вопрос. Какая ошибка возникнет при выполнении программы?*

```

1 data = {"name": "Alice", "age": 25}
2 print(data["city"])

```

Правильный ответ: KeyError: 'city'

Компетенции (индикаторы): ПК-3 (ПК-3.3)

10. *Напишите ответ на вопрос. Что произойдет при выполнении программы?*

```

1 def factorial(n):
2     return n * factorial(n-1)
3
4 print(factorial(3)) |

```

Правильный ответ: Программа будет выполняться бесконечно / Рекурсия без условия останова вызовет ошибку максимальной глубины

Компетенции (индикаторы): ПК-3 (ПК-3.3)

Задания открытого типа с развернутым ответом

1. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Имеется программа, которая запрашивает у пользователя количество товаров и цену за единицу, а затем выводит общую стоимость заказа с учетом скидки 10%, если сумма заказа превышает 1000 рублей:

```

1 def calculate_order_total(items, unit_price):
2     total_cost = items * unit_price
3     discount = [0.1](0.1) if total_cost > 1000 else 0
4     final_cost = total_cost - (total_cost * discount)
5     return round(final_cost, 2)
6
7 items = int(input("Введите количество товаров: "))
8 unit_price = float(input("Введите цену за единицу товара: "))
9
10 total_cost = calculate_order_total(items, unit_price)
11 print(f"Общая стоимость заказа: {total_cost}")

```

Вопрос:

Как работает конструкция if ... else в данном коде? Почему мы используем округление результата до двух знаков после запятой?

Время выполнения 20 минут.

Ожидаемый результат:

Конструкция if ... else проверяет условие и назначает скидку, если общая стоимость заказа превышает 1000 рублей. Округление до двух знаков после запятой выполняется для удобства отображения суммы, так как деньги обычно указываются с точностью до копеек.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

2. *Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.*

Имеется программа, которая запрашивает у пользователя температуру в градусах Цельсия и преобразует ее в градусы Фаренгейта:

```

1 def convert_celsius_to_fahrenheit(celsius):
2     fahrenheit = (celsius * 1.80) + 32
3     return round(fahrenheit, 2)
4
5 celsius_temp = float(input("Введите температуру в градусах Цельсия: "))
6 fahrenheit_temp = convert_celsius_to_fahrenheit(celsius_temp)
7 print(f"Температура в градусах Фаренгейта: {fahrenheit_temp}")

```

Вопрос:

Как работает формула преобразования температуры из Цельсия в Фаренгейт?

Почему мы используем округление до двух знаков после запятой?

Время выполнения 20 минут.

Ожидаемый результат:

Формула преобразования основана на соотношении между градусами Цельсия и Фаренгейта: $F = C \times 1.80 + 32$. Округление до двух знаков после запятой делается для точности представления результата, так как температура часто измеряется с точностью до десятых долей градуса.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

3. Прочитайте текст задания. Продумайте логику и полную ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая принимает предложение от пользователя и выводит среднюю длину слов в этом предложении:

```

1 def average_word_length(sentence):
2     words = sentence.split()
3     total_chars = sum(len(word) for word in words)
4     avg_len = total_chars / len(words)
5     return round(avg_len, 2)
6
7 sentence = input("Введите предложение: ")
8 avg_len = average_word_length(sentence)
9 print(f"Средняя длина слов в предложении: {avg_len}")

```

Вопрос:

Как работает метод .split() в данном коде? Зачем нам нужно суммировать длину всех слов и делить на их количество?

Время выполнения 20 минут.

Ожидаемый результат:

Метод .split() разбивает строку на отдельные слова по пробелам. Сумма длин всех слов делится на их количество, чтобы получить среднее арифметическое. Это позволяет узнать, насколько длинные слова в среднем содержатся в предложении.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

4. Прочитайте текст задания. Продумайте логику и полную ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая выводит таблицу умножения для любого введенного пользователем числа от 1 до 10:

```
1 def multiplication_table(number):
2     for i in range(1, 11):
3         print(f"{number} × {i} = {number * i}")
4
5 number = int(input("Введите число от 1 до 10: "))
6 multiplication_table(number)
```

Вопрос:

Как работает цикл for в данном коде? Почему мы начинаем цикл с 1 и заканчиваем на 10?

Время выполнения 20 минут.

Ожидаемый результат:

Цикл for последовательно перебирает числа от 1 до 10, подставляя каждое из них в выражение умножения. Начальная и конечная границы цикла определяют диапазон чисел, для которых будет выведена таблица умножения.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

5. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая проверяет, удовлетворяет ли пароль минимальным требованиям сложности: длина не менее 8 символов, наличие заглавных и строчных букв, а также хотя бы одной цифры.

```
1 def check_password_complexity(password):
2     has_upper = any(char.isupper() for char in password)
3     has_lower = any(char.islower() for char in password)
4     has_digit = any(char.isdigit() for char in password)
5     length_ok = len(password) >= 8
6
7     if length_ok and has_upper and has_lower and has_digit:
8         return True
9     else:
10        return False
11
12 password = input("Введите пароль: ")
13 is_complex = check_password_complexity(password)
14 print(f"Пароль {' ' if is_complex else 'не '}соответствует требованиям сложности.")
```

Вопрос:

Как работают функции any() и isupper(), islower(), isdigit() в данном коде? Почему проверка длины обязательна перед другими условиями?

Время выполнения 20 минут.

Ожидаемый результат:

Функция any() проверяет, присутствует ли хотя бы одно истинное значение в предоставленной ей последовательности. Функции isupper(), islower(), isdigit() проверяют наличие соответствующих символов в пароле. Проверка длины обязательна первой, потому что она устанавливает основной критерий сложности – минимальная длина пароля.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

6. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая конвертирует сумму денег из одной валюты в другую, используя заранее известные курсы валют:

```
1  exchange_rates = {
2      "USD": 1.0,      # Доллар США
3      "EUR": 0.9,      # Евро
4      "RUB": 70.0,     # Российский рубль
5      "GBP": 0.8,      # Британский фунт
6  }
7
8  def currency_converter(amount, from_currency, to_currency):
9      if from_currency == to_currency:
10         return amount
11         return amount * exchange_rates[to_currency] / exchange_rates[from_currency]
12
13  from_currency = input("Введите исходную валюту (USD, EUR, RUB, GBP): ").upper()
14  amount = float(input("Введите сумму: "))
15  to_currency = input("Введите целевую валюту (USD, EUR, RUB, GBP): ").upper()
16
17  converted_amount = currency_converter(amount, from_currency, to_currency)
18  print(f"{amount} {from_currency} равно {converted_amount} {to_currency}.")
```

Вопрос:

Какие валюты поддерживаются в этой программе? Как можно расширить этот список?

Время выполнения 20 минут.

Ожидаемый результат:

Программа поддерживает четыре валюты: доллар США (USD), евро (EUR), российский рубль (RUB) и британский фунт (GBP). Чтобы расширить список, достаточно добавить новые пары курсов валют в словарь `exchange_rates`.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

7. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая генерирует случайные пароли длиной 12 символов, содержащие заглавные и строчные буквы, цифры:

```

1  import random
2
3  def generate_random_password():
4      letters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
5      digits = "0123456789"
6      special_characters = "!@#$%^&*()"
7
8      all_characters = letters + digits + special_characters
9      password = []
10
11     for _ in range(12):
12         character_type = random.randint(1, 3) # Случайный выбор типа символа
13         if character_type == 1:
14             password.append(random.choice(letters))
15         elif character_type == 2:
16             password.append(random.choice(digits))
17         else:
18             password.append(random.choice(special_characters))
19
20     random.shuffle(password) # Перемешиваем порядок символов
21     return "".join(password)
22
23 password = generate_random_password()
24 print(f"Сгенерированный пароль: {password}")

```

Вопрос:

Как работает генератор случайных чисел в данном коде? Почему мы перемешиваем символы в конце?

Время выполнения 20 минут.

Ожидаемый результат:

Функция `random.randint(1, 3)` выбирает тип символа случайно: либо буква, либо цифра, либо специальный символ. Перемешивание символов в конце гарантирует, что пароль будет выглядеть более хаотично и трудно предсказуемо.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

8. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая принимает текст от пользователя и выводит статистику: количество слов, уникальных слов, символов без учета пробелов и процент уникальности слов:

```

1 def text_statistics(text):
2     words = text.split() # Разбиваем текст на слова
3     unique_words = set(words) # Преобразуем список слов в множество для удаления дублей
4     symbol_count = len(text) - len(' '.join(words)) + len(words) # Подсчет символов без пробелов
5     word_count = len(words)
6     unique_word_ratio = (len(unique_words) / len(words)) * 100 if word_count > 0 else 0
7
8     stats = {
9         "word_count": word_count,
10        "unique_word_count": len(unique_words),
11        "symbol_count_without_spaces": symbol_count,
12        "unique_word_percentage": unique_word_ratio
13    }
14
15    return stats
16
17 text = input("Введите текст: ")
18 stats = text_statistics(text)
19 print(f"Количество слов: {stats['word_count']}\n"
20       f"Уникальных слов: {stats['unique_word_count']}\n"
21       f"Символов без пробелов: {stats['symbol_count_without_spaces']}\n"
22       f"% Уникальных слов: {stats['unique_word_percentage']:.2f}%")

```

Вопрос:

Почему мы используем `set(words)` для получения уникальных слов? Чем отличается операция `split()` от `join()`?

Время выполнения 20 минут.

Ожидаемый результат:

Операция `set(words)` автоматически удаляет дубликаты, превращая список слов во множество. Метод `split()` разбивает строку на слова по пробелам, а `join()` объединяет элементы списка обратно в строку, позволяя легко подсчитать количество символов без пробелов.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

9. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая находит и удаляет дубликаты из списка чисел, оставляя только уникальные значения.

```

1 def remove_duplicates(numbers):
2     result = [] # Новый список для уникальных значений
3     for number in numbers:
4         if number not in result: # Добавляем число в новый список, если его там нет
5             result.append(number)
6     return result
7
8 numbers = list(map(int, input("Введите числа через пробел: ").split()))
9 unique_numbers = remove_duplicates(numbers)
10 print(f"Уникальные числа: {' '.join(map(str, unique_numbers))}")

```

Вопрос:

Почему мы можем безопасно добавлять числа в пустой список, проверяя их отсутствие? Как улучшить производительность этого алгоритма?

Время выполнения 20 минут.

Ожидаемый результат:

Этот алгоритм работает эффективно благодаря тому, что каждый раз проверяет, содержится ли число в текущем списке уникальных значений.

Улучшить производительность можно, предварительно отсортировав список, что позволит ускорить проверку на наличие дубликатов.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

10. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите развернутый и обоснованный ответ.

Имеется программа, которая находит самое частое слово в тексте, игнорируя стоп-слова (например, предлоги, союзы):

```
1 stopwords = {"the", "and", "a", "in", "is", "it"}
2
3 def most_frequent_word(text):
4     words = text.split()          # Разбиваем текст на слова
5     word_counts = {}
6
7     for word in words:
8         if word not in stopwords: # Игнорируем стоп-слова
9             if word in word_counts:
10                word_counts[word] += 1
11            else:
12                word_counts[word] = 1
13
14     max_count = max(word_counts.values())
15     frequent_words = [word for word, count in word_counts.items() if count == max_count]
16
17     return frequent_words[0] if frequent_words else ""
18
19 text = input("Введите текст: ")
20 most_common_word = most_frequent_word(text)
21 print(f"Наиболее частое слово: {most_common_word}")
```

Вопрос:

Почему мы создаем отдельный словарь для подсчета слов? Как можно оптимизировать этот алгоритм?

Время выполнения 20 минут.

Ожидаемый результат:

Словарь `word_counts` позволяет удобно отслеживать количество появлений каждого слова. Оптимизация возможна за счет предварительного удаления стоп-слов перед подсчетом частот, что уменьшит объем обрабатываемых данных.

Критерии оценивания: правильный анализ кода.

Компетенции (индикаторы): ПК-3 (ПК-3.3)

Экспертное заключение

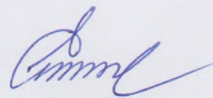
Представленный комплект оценочных материалов по дисциплине (практике) «Бизнес-информатика-2» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 38.03.05 Бизнес-информатика.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии экономического института



Е.Н. Шаповалова

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)