

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
По дисциплине
«Информатика и программирование»

Студент

Гр. БИН-25-2

Ассистент

преподавателя

Д.В. Тимкин

М.В. Водяницкий

Задание

Выполнить задания на Python и оформить отчет по стандартам ВВГУ.

Задание 1. Дан список из 10 различных целых чисел. Необходимо найти в нем число 3 и заменить на 30.

Задание 2. Дан список из 5 целых чисел. Необходимо превратить его в список квадратов этих чисел.

Задание 3. Имеется список различных целых чисел. Программа должна найти наибольшее из чисел списка и разделить его на длину списка.

Задание 4. Имеется кортеж из нескольких произвольных элементов. Необходимо этот кортеж отсортировать. Если хотя бы один элемент не является числом, то кортеж остается неизменным.

Задание 5. Имеется словарь товаров в магазине. Необходимо найти товар с минимальной и максимальной ценой.

Задание 6. Имеется список произвольных элементов. Необходимо на основе этого списка создать словарь, где каждый элемент списка будет и ключом, и значением.

Задание 7. Имеется словарь перевода английских слов на русский, где ключ английского слово, значение - русского. Необходимо реализовать программу которая получает на ввод русское слово и результатом выдает перевод на английский.

Задание 8. Реализовать игру Камень-Ножницы-Бумага-Ящерица-Спок. Программа должна запрашивать у пользователя ввод одного из вариантов. Второй вариант случайно генерирует сама программа и возвращает победителя.

Правила игры следующие:

- Ножницы режут бумагу
- Бумага покрывает камень
- Камень давит ящерицу
- Ящерица отравляет Спока
- Спок ломает ножницы
- Ножницы обезглавливают ящерицу
- Ящерица съедает бумагу
- Бумага подставляет Спока
- Спок испаряет камень
- Камень разбивает ножницы

Задание 9. Дан список слов - например:

["яблоко", "груша", "банан", "киви", "апельсин", "ананас"]

Необходимо создать новый словарь, где:

- Ключом будет первая буква слова
- Значением - список всех слов, начинающихся с этой буквы

Пример результата:

{ 'я': ['яблоко'], 'г': ['груша'], 'б': ['банан'], 'к': ['киви'], 'а': ['апельсин', 'ананас'] }

Задание 10. Дан список кортежей, где каждый кортеж содержит имя студента и его оценки, например:

[("Анна", [5, 4, 5]), ("Иван", [3, 4, 4]), ("Мария", [5, 5, 5])]

Необходимо:

1. Создать словарь, где ключ - имя студента, значение - его средняя оценка
2. Найти студента с наибольшей средней оценкой и вывести его имя и средний балл

Пример результата:

Мария имеет наивысший средний балл: 5.0

Содержание

1	Выполнение работ	
1.1	Задание 1.....	5
1.2	Задание 2.....	5
1.3	Задание 3.....	5
1.4	Задание 4.....	6
1.5	Задание 5.....	7
1.6	Задание 6.....	8
1.7	Задание 7.....	9
1.8	Задание 8.....	9
1.9	Задание 9.....	10
1.10	Задание 10.....	11

1 Выполнение работы

1.1 Задание 1

В данном задании был создан список, содержащий десять различных целых чисел. Далее с помощью цикла производился последовательный просмотр элементов списка с целью поиска числа. При нахождении данного значения оно заменялось на число 30, после чего выводился на экран.

```
numbers1 = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
for i in range(len(numbers1)):
    if numbers1[i] == 3:
        numbers1[i] = 30
print("Задание 1:", numbers1)
```

Рисунок 1 — Листинг программы для задания 1

Пояснение работы программы:

- 1) Создается список numbers1, содержащий 10 различных целых чисел. Введённая строка преобразуется в целое число.
- 2) Запускается цикл for, проходящий по индексам списка.
- 3) На каждой итерации проверяется значение текущего элемента списка.
- 4) Выполняется проверка условия numbers1[i] == 3.
- 5) Если условие истинно, значение элемента заменяется на 30.
- 6) После завершения цикла измененный список выводится на экран с помощью print().

1.2 Задание 2

В данном задании был создан список, содержащий пять целых чисел. Далее с использованием цикла выполнялось возведение каждого элемента списка в квадрат. Полученные значения сохранялись в том же списке, после чего итоговый результат выводился на экран.

```
numbers2 = [2, 4, 6, 8, 10]
for i in range(len(numbers2)):
    numbers2[i] = numbers2[i] ** 2
print("Задание 2:", numbers2)
```

Рисунок 2 — Листинг программы для задания 2

Пояснение работы программы:

- 1) Создается список numbers2, состоящий из пяти целых чисел.
- 2) Запускается цикл for, проходящий по индексам списка.
- 3) На каждой итерации текущее значение элемента возводится в квадрат.
- 4) Полученный результат сохраняется на место исходного элемента списка.

1.3 Задание 3

В данном задании был создан список различных целых чисел. Далее программа последовательно находила наибольшее значение в списке путем сравнения элементов. После нахождения максимального числа оно делилось на длину списка, и результат вычислений выводился на экран.

```
numbers3 = [4, 7, 1, 9, 3]
max_number = numbers3[0]
for n in numbers3:
    if n > max_number:
        max_number = n
result = max_number / len(numbers3)
print("Задание 3:", result)
```

Рисунок 3 — Листинг программы задания 3

Пояснение работы программы:

- 1) Создается список numbers3, содержащий целые числа.
- 2) Переменная max_number инициализируется первым элементом списка.
- 3) Запускается цикл for для перебора всех элементов списка.
- 4) На каждой итерации выполняется сравнение текущего элемента с max_number.
- 5) Если текущий элемент больше, значение max_number обновляется.
- 6) После завершения цикла максимальное число делится на длину списка.
- 7) Полученный результат выводится на экран с помощью print().

1.4 Задание 4

В данном задании был создан кортеж, содержащий несколько элементов. Программа проверяла, являются ли все элементы числами. Если условие выполнялось, кортеж сортировался в порядке возрастания. Результат работы выводился на экран с помощью функции print().

```
data = (3, 1, 4, 2)
only_numbers = True
for x in data:
    if not isinstance(x, (int, float)):
        only_numbers = False

if only_numbers:
    data = tuple(sorted(data))
print("Задание 4:", data)
```

Рисунок 4 — Листинг программы задания 4

Пояснение работы программы:

- 1) Создаем data с произвольными элементами..
- 2) Инициализируется логическая переменная only_numbers значением True.

- 3) Запускается цикл `for` для проверки каждого элемента.
- 4) Если элемент не является числом (`int` или `float`), `only_numbers` присваивается `False`.
- 5) После проверки всех элементов выполняется условие `only_numbers`.
- 6) Если условие истинно, кортеж сортируется с помощью функции `sorted()` и преобразуется обратно в `tuple`.
- 7) Результат выводится на экран с помощью `print()`.

1.5 Задание 5

В данном задании был создан словарь товаров с их ценами. Программа последовательно перебирала все товары, чтобы определить товар с минимальной и максимальной стоимостью. После обработки данных результаты выводились на экран.

```
products = {
    "Хлеб": 45,
    "Молоко": 80,
    "Сыр": 350,
    "Яблоки": 120
}

min_product = list(products.keys())[0]
max_product = list(products.keys())[0]

for product in products:
    if products[product] < products[min_product]:
        min_product = product
    if products[product] > products[max_product]:
        max_product = product

print("Задание 5:")
print("Минимальная цена:", min_product, products[min_product])
print("Максимальная цена:", max_product, products[max_product])
```

Рисунок 5 — Листинг программы задания 5

Пояснение работы программы:

- 1) Создается словарь `products`, где ключ — название товара, значение — цена.
- 2) Инициализируются переменные `min_product` и `max_product` первым ключом словаря.
- 3) Запускается цикл `for` по всем ключам словаря.
- 4) На каждой итерации проверяется, меньше ли цена текущего товара цены `min_product`.
- 5) Если условие истинно, `min_product` обновляется.
- 6) Проверяется, больше ли цена текущего товара цены `max_product`.
- 7) Если условие истинно, `max_product` обновляется.
- 8) После завершения цикла выводятся на экран товары с минимальной и максимальной ценой с помощью `print()`.

1.6 Задание 6

В данном задании был создан список произвольных элементов. На основе этого списка формировался словарь, где каждый элемент списка становился одновременно ключом и значением. После завершения обработки результат выводился на экран.

```
elements = ["a", 1, True, 3.14]
new_dict = {}
for elem in elements:
    new_dict[elem] = elem
print("Задание 6:", new_dict)
```

Рисунок 6 — Листинг программы задания 6

Пояснение работы программы:

- 1) Создается список `elements` с произвольными элементами.
- 2) Инициализируется пустой словарь `new_dict`.
- 3) Запускается цикл `for` для перебора всех элементов списка.
- 4) Каждый элемент списка добавляется в словарь как ключ и как значение.
- 5) После завершения цикла сформированный словарь выводится на экран с помощью `print()`.

1.7 Задание 7

В данном задании был создан словарь английских слов с переводом на русский язык. Программа запрашивала у пользователя русское слово и последовательно проверяла словарь, чтобы найти соответствующее английское слово. Если совпадение находилось, результат выводился на экран.

```
dictionary = {
    "cat": "кот",
    "dog": "собака",
    "apple": "яблоко",
    "book": "книга"
}

rus_word = input("Задание 7. Введите русское слово: ")
found = False

for eng in dictionary:
    if dictionary[eng] == rus_word:
        print("Перевод:", eng)
        found = True

if not found:
    print("Слово не найдено")
```

Рисунок 7 — Листинг программы задания 7

Пояснение работы программы:

- 1) Создается словарь dictionary, где ключ — английское слово, значение — перевод на русский.
- 2) Программа запрашивает ввод русского слова пользователем.
- 3) Инициализируется логическая переменная found значением False.
- 4) Запускается цикл for по всем ключам словаря.
- 5) На каждой итерации проверяется, совпадает ли значение словаря с введенным словом.
- 6) Если совпадение найдено, выводится перевод на английский язык и found присваивается True.
- 7) После завершения цикла проверяется значение found.
- 8) Если значение False, выводится сообщение о том, что слово не найдено.

1.8 Задание 8

В данном задании была реализована игра Камень-Ножницы-Бумага-Ящерица-Спок. Программа запрашивала у пользователя выбор одного из вариантов, второй вариант случайно выбирался программой. После этого выполнялась проверка правил игры и определялся победитель, результат выводился на экран.

```
user_choice = input("Задание 8. Ваш выбор: ").lower()
computer_choice = "камень"
print("Компьютер выбрал:", computer_choice)

if user_choice == computer_choice:
    print("Ничья")
elif user_choice == "бумага":
    print("Вы победили")
elif user_choice == "ножницы":
    print("Компьютер победил")
elif user_choice == "ящерица":
    print("Компьютер победил")
elif user_choice == "спок":
    print("Вы победили")
else:
    print("Неверный ввод")
```

Рисунок 8 — Листинг программы задания 8

Пояснение работы программы:

- 1) Программа запрашивает у пользователя ввод одного из вариантов игры.
- 2) Второй вариант выбирается заранее (например, computer_choice = "камень").
- 3) Сравниваются выборы пользователя и компьютера.
- 4) Если варианты совпадают, объявляется ничья.
- 5) Проверяются все правила игры для определения победителя.

- 6) В зависимости от условий выводится сообщение о победе пользователя или компьютера.
- 7) Если ввод некорректен, выводится сообщение об ошибке.

1.9 Задание 9

В данном задании был создан список слов. Программа формировала новый словарь, где ключом являлась первая буква слова, а значением — список всех слов, начинающихся с этой буквы. После завершения обработки результат выводился на экран.

```
words = ["яблоко", "груша", "банан", "киви", "апельсин", "ананас"]
word_dict = {}

for word in words:
    letter = word[0]
    if letter not in word_dict:
        word_dict[letter] = []
    word_dict[letter].append(word)

print("Задание 9:", word_dict)
```

Рисунок 9 — Листинг программы задания 9

Пояснение работы программы:

- 1) Создается список words с набором слов.
- 2) Инициализируется пустой словарь word_dict .
- 3) Запускается цикл for для перебора всех слов списка.
- 4) Для каждого слова определяется первая буква.
- 5) Проверяется, существует ли уже ключ с этой буквой в словаре.
- 6) Если ключ отсутствует, создается новый список для этой буквы.
- 7) Слово добавляется в список по соответствующему ключу.
- 8) После завершения цикла словарь выводится на экран с помощью print()

1.10 Задание 10

В данном задании был создан список с именами студентов и их оценками. Программа вычисляла средний балл каждого студента, формировала словарь с результатами и определяла студента с наивысшим средним баллом. Результат выводился на экран.

```
students = [
    ("Анна", [5, 4, 5]),
    ("Иван", [3, 4, 4]),
    ("Мария", [5, 5, 5])
]

averages = {}
for name, grades in students:
    averages[name] = sum(grades) / len(grades)

best_student = ""
best_score = 0

for name in averages:
    if averages[name] > best_score:
        best_score = averages[name]
        best_student = name

print(best_student, "имеет наивысший средний балл:", best_score)
```

Рисунок 10 — Листинг программы задания 10

Пояснение работы программы:

- 1) Создается список `students`, содержащий кортежи с именами и оценками.
- 2) Инициализируется пустой словарь `averages` для хранения средних баллов.
- 3) Запускается цикл `for` по всем кортежам списка.
- 4) Для каждого студента вычисляется среднее значение оценок и сохраняется в словарь `averages`.
- 5) Инициализируются переменные `best_student` и `best_score` для хранения лучшего результата.
- 6) Запускается цикл `for` по ключам словаря `averages`.
- 7) На каждой итерации проверяется, больше ли средний балл текущего студента `best_score`.
- 8) Если условие истинно, обновляются `best_student` и `best_score`.
- 9) После завершения цикла выводится имя студента с наивысшим средним баллом и его значение с помощью `print()`.