

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 \_\_\_\_\_ А.М. Сахарюк  
Ассистент  
преподавателя \_\_\_\_\_ 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 слова, 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:

- 1) Ножницы режут бумагу
- 2) Бумага покрывает камень
- 3) Ящерица отравляет Спока
- 4) Спок ломает ножницы
- 5) Ножницы обезглавливают ящерицу
- 6) Ящерица съедает бумагу
- 7) Бумага подставляет Спока
- 8) Спок испаряет камень
- 9) Камень разбивает ножницы

**Задание 9.** Дан список слов - например:

[”яблоко ”груша ”банан ”киви ”апельсин ”ананас”] Необходимо создать новый словарь, где:

- 1) Ключом будет первая буква слова
- 2) 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 работы .....	3
1.1 Задание 1 .....	3
1.2 Задание 2 .....	3
1.3 Задание 3 .....	4
1.4 Задание 4 .....	5
1.5 Задание 5 .....	6
1.6 Задание 6 .....	6
1.7 Задание 7 .....	7
1.8 Задание 8 .....	8
1.9 Задание 9 .....	9
1.10 Задание 10 .....	9

## 1 Выполнение работы

### 1.1 Задание 1

В данном задании был объявлен список содержащий 10 элементов и заполненный случайными числами в диапазоне от 0 до 10. Программа ищет число 3 в списке и заменяет его на 30. На рисунке 1 представлен код полученной программы.

```

1 import random
2
3 list_of_numbers = []
4
5 for i in range(10):
6     list_of_numbers.append(random.randint(0,10))
7
8 print(f'данный список: {list_of_numbers}')
9
10 for i in range(len(list_of_numbers)):
11     if list_of_numbers[i] == 3:
12         list_of_numbers[i] = 30
13
14 print(f'Лист после операций: {list_of_numbers}')

```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) импортируется стандартная библиотека random
- 2) Объявляется список list\_of\_numbers
- 3) В список добавляется 10 случайных чисел в диапазоне от 0 до 10
- 4) Через цикл перебираются значения списка
- 5) Если обнаруживается число 3, то оно заменяется на 30
- 6) Список выводится на экран

После выполнения программы в консоль выводится список после выполнения всех операций над ним.

### 1.2 Задание 2

В данном задании был объявлен список содержащий 5 элементов и заполнены случайными числами в диапазоне от 0 до 100. Программа заменяет все числа в списке на их квадраты. На рисунке 2 представлен код полученной программы.

```

1 import random
2
3 list_of_numbers = []
4
5 for i in range (5):
6     list_of_numbers.append(random.randint(0,100))
7
8 print(f'Данный список: {list_of_numbers}')
9
10 for i in range(len(list_of_numbers)):
11     list_of_numbers[i] **= 2
12
13 print(f'Лист после операций: {list_of_numbers}')

```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы

```

1 goods = {"яблоко": 100, "банан": 80, "груша": 120}
2
3 max_price = ""
4 for key, value in goods.items():
5     if(max_price == "" or goods[max_price] < goods[key]):
6         max_price = key
7
8 print(f'Максимальная цена: {max_price} - {goods[max_price]}')

```

Рисунок 3 – Листинг программы для задания 5

ы:

- 1) импортируется стандартная библиотека random
- 2) Объявляется список list\_of\_numbers
- 3) В список добавляется 5 случайных чисел в диапазоне от 0 до 100
- 4) Через цикл перебираются значения списка
- 5) каждое число возводится в квадрат и возвращается в список
- 6) Список выводится на экран

В результате выполнения программы, в консоль выводится список после выполнения всех операций над ним.

### 1.3 Задание 3

В данном задании был объявлен список содержащий случайное количество элементов от 1 до 1000 и заполненый случайными числами в диапазоне от 0 до 10000. Программа находит самое большое число и делит его на длину списка На рисунке 4 представлен код полученной программы.

```

1 import random
2
3 list_of_numbers = []
4
5 for i in range(random.randint(1,1000)):
6     list_of_numbers.append(random.randint(0,10000))
7
8 list_len = len(list_of_numbers)
9 max_int = max(list_of_numbers)
10
11 print(f'Длина данного списка: {list_len}, Наибольшее число: {max_int}')
12
13 print(f'результат: {int(max_int / list_len)})')

```

Рисунок 4 – Листинг программы для задания 3

Пояснение работы программы:

- 1) импортируется стандартная библиотека random
- 2) Объявляется список list\_of\_numbers
- 3) В список добавляется от 1 до 1000 случайных чисел в диапазоне от 0 до 10000
- 4) Через функцию len() находится длина списка
- 5) Через функцию max() находится наибольшее число в списке
- 6) На экран выводится максимальное число в списке делёное на длину списка.

В результате выполнения программы, в консоль выводится результат выполнения формулы.

#### 1.4 Задание 4

В данном задании от пользователя требуется вести что либо с клавиатуры через пробел. Программа создаёт кортеж, который пытается отсортировать. Если в кортеже находится не число, то кортеж не изменяется. На рисунке 5 представлен код полученной программы.

```

1 user_input = input('введите значения для сортировки через
                     пробел: ')
2
3 user_tuple = tuple(user_input.split(' '))
4
5 try:
6     for i in user_tuple:
7         int(i)
8     user_tuple = tuple(sorted(user_tuple))
9     print(f'результат: {user_tuple}')
10 except:
11     print('Среди введённых значений есть не число, сортировка не
           производится')
12     print(f'результат: {user_tuple}')

```

Рисунок 5 – Листинг программы для задания 4

Пояснение работы программы:

- 1) У пользователя запрашивается ввод

- 2) Ввод пользователя записывается в переменную user\_input.
- 3) Объявляется user\_tuple равный разделённому по пробелам вводу пользователя
- 4) Каждое значение в кортеже пытается преобразоваться в целочисленное значение
- 5) В случае удачи - кортеж сортируется, в случае неудачи - кортеж не сортируется
- 6) Кортеж выводится на экран

В результате выполнения программы, в консоль выводится список после выполнения всех операций над ним.

## 1.5 Задание 5

В данном задании дан словарь наименований товаров и их цен. Программа должна найти товар с максимальной ценой и вывести его на экран. На рисунке 6 представлен код полученной программы.

```

1 goods = {"яблоко": 100, "банан": 80, "груша": 120}
2
3 max_price = ""
4 for key, value in goods.items():
5     if(max_price == "" or goods[max_price] < goods[key]):
6         max_price = key
7
8 print(f'Максимальная цена: {max_price} - {goods[max_price]}')

```

Рисунок 6 – Листинг программы для задания 5

Пояснение работы программы:

- 1) задаётся словарь goods со значениями "яблоко": 100, "банан": 80, "груша": 120
- 2) Объявляется переменная max\_price с типом string
- 3) Через цикл перебираются значения словаря
- 4) каждое значение проверяется на условие: если max\_price пуста, или цена товара больше чем в max\_price, то в max\_price записывается данное значение
- 5) а экран выводится максимальная цена товара

В результате выполнения программы в консоль выводится максимальная цена товара данного списка.

## 1.6 Задание 6

В данном задании от пользователя требуется ввести любые значения через пробел. Программа должна для каждого значения введённого пользователем создать вход в словарь, где значение является и ключём и значением. На рисунке 7 представлен код программы.

```

1 user_input = input('введите значения для создания словаря
2      пробел: ')
3 user_list = user_input.split(' ')
4
5 user_dict = {}
6
7 for i in user_list:
8     user_dict[i] = i
9
10 print(f'Результат: {user_dict}')

```

Рисунок 7 – Листинг программы для задания 6

Пояснение работы программы:

- 1) У пользователя запрашивается ввод
- 2) Ввод пользователя записывается в переменную user\_input.
- 3) Объявляется user\_list равный разделённому по пробелам вводу пользователя
- 4) Объявляется user\_dict
- 5) Через цикл перебираются значения ввода пользователя
- 6) В словарь вводится значение где ключ и значение - одно и то же
- 7) В консоль выводится словарь после операций

В результате выполнения программы в консоль выводится словарь после операций.

## 1.7 Задание 7

В данном задании программе требуется вывести перевод слова из словаря. На рисунке 8 представлен код программы.

```

1 translation_dict = {"Hello" : 'Привет', 'Bye' : "Пока"}
2
3 user_input = input('введите Слово на русском: ')
4
5 been_translated = False
6 for key, value in translation_dict.items():
7     if(value == user_input):
8         print(f"Перевод слова: {key}")
9         been_translated = True
10        break
11 if(not been_translated):
12     print('Такого слова нет в словаре')

```

Рисунок 8 – Листинг программы для задания 7

Пояснение работы программы:

- 1) У пользователя запрашивается ввод
- 2) Через цикл перебирается словарь для перевод
- 3) Если ввод пользователя содержится в элементе словаря, то его перевод выводится на экран

В результате выполнения программы, в консоль выводится перевод введённого пользователем слова.

### 1.8 Задание 8

В данном задании нужно написать игру Камень-Ножницы-Бумага-Ящерица-Спок. От пользователя требуется выбрать свой ход. Программа выводит выбор бота и то, кто победил. На рисунке 9 представлен код программы.

```

1 import random
2
3 Winner_dict = {
4     'Ножницы': ['Бумага', 'режут бумагу', 'Ящерица', 'обезглавливают ящерицу'],
5     'Бумага': ['Камень', 'покрывает камень', 'Спок', 'подставляет Спока'],
6     'Камень': ['Ножницы', 'разбивает ножницы', 'Ящерица', 'давит ящерицу'],
7     'Ящерица': ['Спок', 'отравляет Спока', 'Бумага', 'съедает бумагу'],
8     'Спок': ['Ножницы', 'ломает ножницы', 'Камень', 'испаряет камень']
9 }
10
11 print('Доступные варианты: Камень Ножницы Бумага Ящерица Спок')
12 user_input = input('Введите свой вариант: ')
13
14 bot_choise = list(Winner_dict.keys())[random.randint(0,4)]
15 print(f'Бот выбрал: {bot_choise}')
16
17 if(bot_choise in Winner_dict[user_input]):
18     win_phrase_index = Winner_dict[user_input].index(
19         bot_choise) + 1
20     print(f'{user_input} {Winner_dict[user_input][
21         win_phrase_index]}')
22     print("Вы победили!")
23 elif(user_input in Winner_dict[bot_choise]):
24     win_phrase_index = Winner_dict[bot_choise].index(
25         user_input) + 1
26     print(f'{bot_choise} {Winner_dict[bot_choise][
27         win_phrase_index]}')
28     print("Вы проиграли!")
29 else:
30     print('Ничья')

```

Рисунок 9 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Задаётся словарь Со следующими значениями: Выбор хода, список - 1 противоположный ход, сообщение о победе, 2 противоположный ход, сообщение о победе.
- 2) От пользователя запрашивается ввод.
- 3) Задаётся случайный выбор бота
- 4) Проверяется кто победил и выводиться на экран сообщение о победе.

В результате выполнения программы, по правилам игры Камень-Ножницы-Бумага-Ящерица-Спок, у пользователя запрашивается ввод, и генерируется результат, кто победил.

## 1.9 Задание 9

В данном задании требуется получить список элементов, найти первые буквы элементы и вывести на экран все элементы списка начинающиеся с этой буквы. На рисунке 10 представлен код программы.

```

1 words = ["яблоко", "груша", "банан", "киви", "апельсин", "
2           ананас"]
3 result = {}
4
5 for word in words:
6     first_letter = word[0]
7     if first_letter in result:
8         result[first_letter].append(word)
9     else:
10        result[first_letter] = [word]
11
12 print(f'Результат: {result}')

```

Рисунок 10 – Листинг программы для задания 9

Пояснение работы программы:

- 1) Задаётся список слов.
- 2) Задаётся словарь result.
- 3) Каждый элемент списка перебирается, и в случае обнаружения новой буквы, она добавляется, как ключ в словарь.
- 4) Если обнаруживается слово на то же букву, то оно добавляется в список по ключу словаря.
- 5) На экран выводиться готовый словарь.

В результате выполнения программы в консоль выводится готовый словарь.

## 1.10 Задание 10

В данном задании требуется получить словарь с именами студентов и списком их оценок. Требуется перебрать список и найти средний балл студентов, затем вывести студента с высшим баллом. На рисунке 11 представлен код программы.

```
1 students = [("Анна", [5, 4, 5]), ("Иван", [3, 4, 4]), ("  
    Мария", [5, 5, 5])]  
2  
3 average_grades = {}  
4 for name, grades in students:  
5     average = sum(grades) / len(grades)  
6     average_grades[name] = average  
7  
8 print(f"Словарь средних оценок: {average_grades}")  
9  
10 best_student = max(average_grades, key=average_grades.get)  
11 best_grade = average_grades[best_student]  
12  
13 print(f"{best_student} имеет наивысший средний балл: {  
    best_grade}")
```

Рисунок 11 – Листинг программы для задания 10

Пояснение работы программы:

- 1) Задаётся кортеж `students`, С именами студентов и списком их оценок.
- 2) Задаётся словарь `average_grades`.
- 3) Каждый элемент кортежа перебирается и записывается в словарь в порядке: имя студента, средняя оценка.
- 4) Вычисляется студент с наивысшим средним баллом, и записывается в переменную
- 5) На экран выводится имя лучшего студента и его средний балл.

В результате выполнения данной программы выводится в консоль имя лучшего студента и его средний балл.