

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 основы программирования»

Студент		
гр. БИН-25-3	_____	В.Е. Соловьева
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) Камень давит ящерицу
- 4) Ящерица отравляет Спока
- 5) Спок ломает ножницы
- 6) Ножницы обезглавливают ящерицу
- 7) Ящерица съедает бумагу
- 8) Бумага подставляет Спока
- 9) Спок испаряет камень
- 10) 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

Оформление отчета. Отчет оформляется строго по СТО - скачать требования можно с сайта ВВГУ, либо отсюда

Не забудьте добавить страницу "Задание" с копией содержимого этого файла (с правильным оформлением списков и т.д.)

В отчете должно быть объяснено как работает ваша программа (каждое отдельное задание)

Содержание

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

1 Выполнение работы

1.1 Задание 1

В данном задании необходимо написать программу, которая находит в данном списке число 3 и заменяет его на 30. На рисунке 1 представлен код полученной программы.

```
1 kitty = [1, 50, 58, 3, 67, 4, 6, 93, 19, 49]
2 print(kitty)
3 index_of_3 = kitty.index(3)
4 kitty[index_of_3] = 30
5 print(kitty)
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Создается список kitty из 10 различных целых чисел и выводится в консоль.
- 2) В списке находится число 3 и его индекс записывается в переменную.
- 3) По индексу число 3 заменяется на 30 и измененный список выводится в консоль.

1.2 Задание 2

В данном задании необходимо превратить список чисел в список квадратов этих чисел. На рисунке 2 представлен код программы.

```
1 doggy = [2, 3, 4, 5, 6]
2 print(doggy)
3 length = len(doggy)
4 i=0
5 for i in range(length):
6     doggy[i] = doggy[i]**2
7     i+=1
8 print(doggy)
```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Создается список doggy из 5 целых чисел и выводится в консоль.
- 2) Создается переменная length и ей присваивается значение длины данного списка.
- 3) Цикл for проходится по каждому числу в списке и заменяет числа на их квадраты.
- 4) Полученный список выводится в консоль.

1.3 Задание 3

В данном задании необходимо написать программу, которая находит наибольшее из чисел списка и делит его на длину списка. На рисунке 3 представлен код программы.

```
1 huggy = [45, 4868, 29, 6699, 247, 90, 1, 2, 56, 9437, 7]
2 huggy.sort()
3 print(huggy[-1]/len(huggy))
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Создается список huggy.
- 2) Данный список сортируется по возрастанию с помощью метода sort().
- 3) В результате сортировки наибольшее число оказывается в конце.
- 4) Последнее (наибольшее) число списка делится на длину списка и результат выводится в консоль.

1.4 Задание 4

Необходимо написать программу, которая сортирует кортеж. На рисунке 4 представлен код решения.

```
1 collar = (357, 9, 18, 496)
2 collar = sorted-collar)
3 print-collar)
```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Создается кортеж collar.
- 2) Данный кортеж сортируется с помощью функции sorted() и выводится в консоль.

1.5 Задание 5

В данном задании требуется написать программу, которая в словаре товаров в магазине находит товар с минимальной и максимальной ценой. На рисунке 5 представлен код программы.

```
1 shop_items = {
2     'киви': 50,
3     'банан': 40,
4     'свекла': 60,
5     'арбуз': 100
6 }
7 min_shop_key = min(shop_items, key=shop_items.get)
8 max_shop_key = max(shop_items, key=shop_items.get)
9 print('Товар с минимальной ценой: ' + min_shop_key)
10 print('Товар с максимальной ценой: ' + max_shop_key)
```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Создается словарь.
- 2) С помощью функции `min` и метода `get` находится товар с минимальной ценой.
- 3) С помощью функции `max` и метода `get` находится товар с максимальной ценой.
- 4) Товары с минимальной и максимальной ценами выводятся в консоль.

1.6 Задание 6

В данном задании требуется написать программу, которая на основе списка произвольных элементов создает словарь, где каждый элемент списка является и ключом, и значением. На рисунке 6 представлен код программы.

```
1 jjba = ['night owl', 'ILE', 'sp7']
2 print(dict(zip(jjba, jjba)))
```

Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Создается список произвольных элементов `jjba`.
- 2) С помощью функций `zip()` и `dict()` создается словарь, где каждый элемент из списка является и ключом, и значением.

1.7 Задание 7

В данном задании необходимо написать программу, которая получает на ввод русское слово и результатом выдает перевод на английский. Имеется словарь перевода английских слов на русский. На рисунке 7 представлен код программы.

```
1 en_dict = {
2     'crucifix': 'распятие',
3     'bludgeon': 'дубинка',
4     'blister': 'волдырь',
5     'cockpit': 'кабина пилота',
6     'bootleg': 'контрабандный'
7 }
8 reversed_en_dict = {
9     value: key
10    for key, value
11    in en_dict.items()
12 }
13 ru = input('Введите слово на русском для перевода: ')
14 print('Перевод слова ' + ru + ': ' + reversed_en_dict[ru])
```

Рисунок 7 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Создается словарь перевода английских слов на русский, где ключ английского слово, значение - русского.
- 2) На основе словаря, создается словарь, где ключи и значения поменяны местами.
- 3) Пользователь вводит слово для перевода и заданное значение присваивается переменной `ru`.
- 4) Перевод слова выводится в консоль.

1.8 Задание 8

В данном задании нужно написать программу, которая реализовывает игру Камень-Ножницы-Бумага-Ящерица-Спок. Программа запрашивает ввод одного из вариантов. Вторым вариантом случайно генерирует сама программа. На рисунке 8 представлен код программы.

```

1 import random
2 game = ['камень', 'ножницы', 'бумага', 'ящерица', 'спок']
3 user_input = input('Ввод (камень/ножницы/бумага/ящерица/спок): ')
4 random_item = random.choice(game)
5 print('Пользователь: ' + user_input + ', ' + 'Программа: ' + random_item)
6 if user_input == 'ножницы' and random_item == 'бумага': print('Ножницы режут бумагу! Пользователь победил!')
7 if random_item == 'ножницы' and user_input == 'бумага': print('Ножницы режут бумагу! Программа победила!')
8 if user_input == 'бумага' and random_item == 'камень': print('Бумага покрывает камень! Пользователь победил!')
9 if random_item == 'бумага' and user_input == 'камень': print('Бумага покрывает камень! Программа победила!')
10 if user_input == 'камень' and random_item == 'ящерица': print('Камень давит ящерицу! Пользователь победил!')
11 if random_item == 'камень' and user_input == 'ящерица': print('Камень давит ящерицу! Программа победила!')
12 if user_input == 'ящерица' and random_item == 'спок': print('Ящерица отравляет Спика! Пользователь победил!')
13 if random_item == 'ящерица' and user_input == 'спок': print('Ящерица отравляет Спика! Программа победила!')
14 if user_input == 'спок' and random_item == 'ножницы': print('Спок ломает ножницы! Пользователь победил!')
15 if random_item == 'спок' and user_input == 'ножницы': print('Спок ломает ножницы! Программа победила!')
16 if user_input == 'ножницы' and random_item == 'ящерица': print('Ножницы обезглавливают ящерицу! Пользователь победил!')
17 if random_item == 'ножницы' and user_input == 'ящерица': print('Ножницы обезглавливают ящерицу! Программа победила!')
18 if user_input == 'ящерица' and random_item == 'бумага': print('Ящерица съедает бумагу! Пользователь победил!')
19 if random_item == 'ящерица' and user_input == 'бумага': print('Ящерица съедает бумагу! Программа победила!')
20 if user_input == 'бумага' and random_item == 'спок': print('Бумага подставляет Спика! Пользователь победил!')
21 if random_item == 'бумага' and user_input == 'спок': print('Бумага подставляет Спика! Программа победила!')
22 if user_input == 'спок' and random_item == 'камень': print('Спок испаряет камень! Пользователь победил!')
23 if random_item == 'спок' and user_input == 'камень': print('Спок испаряет камень! Программа победила!')
24 if user_input == 'камень' and random_item == 'ножницы': print('Камень разбивает ножницы! Пользователь победил!')
25 if random_item == 'камень' and user_input == 'ножницы': print('Камень разбивает ножницы! Программа победила!')
26 if random_item == 'камень' and user_input == 'камень': print('Ничья!')
27 if random_item == 'ножницы' and user_input == 'ножницы': print('Ничья!')
28 if random_item == 'бумага' and user_input == 'бумага': print('Ничья!')
29 if random_item == 'ящерица' and user_input == 'ящерица': print('Ничья!')
30 if random_item == 'спок' and user_input == 'спок': print('Ничья!')

```

Рисунок 8 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Импортируется библиотека `random`.
- 2) Создается список с элементами игры `game`.
- 3) Запрашивается ввод пользователя и введенный вариант присваивается переменной.
- 4) С помощью функции `random.choice()` случайный вариант присваивается переменной.
- 5) Выводится вариант пользователя и случайный вариант программы.
- 6) Перебираются все возможные комбинации и победитель выводится в консоль.

1.9 Задание 9

В данном задании необходимо написать программу, которая из списка создает новый словарь, где:

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

На рисунке 9 представлен код программы.

```

1 words = ['раскалывать', 'ящерица', 'мангус', 'бегемот', 'рысь']
2 def create_dict(words):
3     frost = {}
4     for word in words:
5         first_letter = word[0]
6         if first_letter in frost: frost[first_letter].append
          (word)
7         else: frost[first_letter] = [word]
8     return frost
9 print(create_dict(words))

```

Рисунок 9 – Листинг программы для задания 9

Пояснение работы программы:

- 1) Создается список слов words.
- 2) Создается функция.
- 3) Создается пустой словарь frost.
- 4) С помощью цикла for программа проходится по каждому слову в данном списке.
- 5) В цикле создается переменная, которой присваивается значение первой буквы слова.
- 6) С помощью условного оператора if проверяется, является ли данная первая буква ключом в словаре frost.
- 7) Если данная первая буква является ключом в словаре frost, то слово, которому принадлежит эта первая буква добавляется как значение к этому ключу.
- 8) Если данная первая буква не является ключом в словаре frost, то она записывается туда в качестве ключа, а слово, которому она принадлежит добавляется к этому ключу как значение.
- 9) Выводится созданный словарь.

1.10 Задание 10

В данном задании необходимо написать программу, которая создает словарь, где ключ - имя студента, значение - его средняя оценка. А также находит студента с наибольшей средней оценкой и выводит его имя и средний балл. На рисунке 10 представлен код программы.

```
1 students = [  
2     ('Алиса', [2,2,2]),  
3     ('Даша', [5,5,5]),  
4     ('Денис', [3,2,2])  
5 ]  
6 labubu = {  
7     student: sum(grades) / len(grades)  
8     for student, grades in students  
9 }  
10 max_avg = max(labubu, key=labubu.get)  
11 print(max_avg + ' имеет наивысший средний балл: ' + str(labubu  
    .get(max_avg)))
```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

- 1) Создается список кортежей студентов и их оценок students.
- 2) Создается словарь labubu студентов и их средних оценок.
- 3) С помощью функции max и метода get находится студент с наивысшим средним баллом.
- 4) Имя студента с наивысшим средним баллом и его балл выводятся в консоль.