

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

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

По дисциплине
«Информатика и программирование»

Студент
гр. Бин-25-2

И.А. Маклаков

Ассистент

М.В. Водяницкий

Преподавателя

Владивосток 2026

Задание

Выполнить задания и написать отчёт 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 выдаёт 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 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	5
1.6 Задание 6	5
1.7 Задание 7	6
1.8 Задание 8	7
1.9 Задание 9	8
1.10 Задание 10.....	9

1 Выполнение работы

1.1 Задание 1

В данном задании сделан код, который из списка 10 различных чисел находит значение 3 и заменяет на 30. Программа изображена на Рисунке 1.

```

1      #Change 3 to 30 in list
2      list1 = []
3      for i in range(1, 11):
4          list1.append(i)
5      print(list1)
6      l = len(list1)
7      for i in range(l):
8          if list1[i] == 3 in list1:
9              list1[i] = 30
10     print(list1)

```

Рисунок 1 - код для задания 1

Пояснение работы программы:

- 1) Создаётся пустой список
- 2) Циклом for i in range с диапазоном от 1 до 10 заполняется список различными целыми значениями
- 3) Командой print выводится полученный список в консоль
- 4) Переменная l получает в значение длину списка, а после по ней запускается цикл for I in range
- 5) Функцией if меняем значение на 30 если список от i равен 3
- 6) Выводим результат через print()

Программа способна найти в списке 3 и заменить на 30.

1.2 Задание 2

В данном задании сделан код, который из списка чисел делает новый список с квадратами этих чисел. Программа изображена на Рисунке 2.

```

1      #Made list with square of numbers
2      list1 = []
3      for i in range(1, 6):
4          list1.append(i)
5      l = len(list1)
6      print(list1)
7      for i in range(l):
8          list1[i] = list1[i] ** 2
9      print(list1)

```

Рисунок 2 - Код для задания 2

Пояснение работы программы:

- 1) Создаётся пустой список
- 2) Циклом for i in range в диапазоне от 1 до 5 заполняем список целыми числами
- 3) Переменной l даём значение длины списка
- 4) Выводим полученный список командой print()

5) Циклом for i in range проходимся по каждому элементу списка длинной 1 и заменяем его на их квадраты.

6) Выводим изменённый список командой print()

Программа корректно выводит список квадратов заданных чисел.

1.3 Задание 3

В данном задании сделан код, который находит максимальное значение списка и делит его на длину списка. Программа изображена на Рисунке 3.

```

1      #Found max argument and divide it on length
2      list1 = [1, 3, 4, 5, 7, 8, 9, 12, 15, 26, 13]
3      l = len(list1)
4      print(f'List length - {l}')
5      m = max(list1)
6      print(f'List max value - {m}')
7      res = m / l
8      print(res)

```

Рисунок 3 - Код для задания 3

Пояснение работы программы:

- 1) Создаём список и даём несколько случайных значений
- 2) Переменной l даём значение длины списка и командой print() выводим значение его длины
- 3) Переменной m максимальное значение из списка и командой print() выводим его
- 4) Создаём переменной res и даём ей значение m делённой на l, после выводим значение в консоль командой print()

Программа успешно находит максимальный аргумент и делит его на длину списка.

1.4 Задание 4

В данном задании сделан код, который сортирует кортеж, но если в нём находится хотя бы одно не целостное значение, не тип данных int, то на выход выводится тот же самый кортеж. Программа изображена на Рисунке 4.

```

1      #Sort tuple and if any argument not 'int' back to unsorted tuple
2      def f(x):
3          for z in x:
4              try:
5                  c = int(z)
6                  if c != z:
7                      return(x)
8              except:
9                  return(x)
10             z = list(x)
11             z.sort()
12             return tuple(z)
13 kort1 = (1, 9, 5, 2, 3)
14 print(f'Entered tuple:{kort1}')
15 print(f'Tuple after sorting with options: {f(kort1)}')

```

Рисунок 4 - Код для задания 4

Пояснение работы программы:

- 1) Создаётся функция f(x). В ней происходит проверка на тип данных int. Блок try-except внутри функции проверяет каждый элемент кортежа, если преобразование z в int выдаёт тоже самое значение с, то возвращается значение, если преобразование z в int выдаёт ошибку, то переходит в except и возвращает исходное значение. Переменной z даём значение кортежа преобразованного в список, сортируем и возвращаем из функции кортеж z
- 2) Создаём кортеж и вводим ему случайные значения
- 3) Командой print() выводим исходный кортеж, а вторым print() выводим кортеж с функцией f(x)

Программа корректно проверяет содержание кортежа и сортирует его при выполнении условия.

1.5 Задание 5

В данном задании сделан код, который находит товар из словаря с максимальной и минимальной ценой. Программа изображена на Рисунке 5.

```

1      #Find max and min price of fruits(in task was anything from market)
2      dict1 = {'Bananas': 3, 'Apples': 3, 'Kiwi': 5, 'Cabbage': 2, 'Carrot': 3}
3      print(dict1)
4      ly = min(dict1, key=dict1.get)
5      re = max(dict1, key=dict1.get)
6      print(f'{re} - have max price, {ly} - have min price')

```

Рисунок 5 - код для задания 5

Пояснение работы программы:

- 1) Создаём словарь с продуктами, я выбрал фрукты и овощи.
- 2) Выводим полученный словарь командой print()
- 3) Переменной ly даём минимальное значение из словаря, а переменной re максимальное.
- 4) В скобках указываем чтобы возвращал имя ключ, а искал минимальное и максимальное по значению
- 5) Командой print() выводим продукт с максимальной и минимальной ценой.

Программа успешно выводит наиболее дорогой товар из словаря с продуктами.

1.6 Задание 6

В данном задании сделан код, который из полученных аргументов создаём словарь, где значение и ключ имеют одно значение. Программа изображена на Рисунке 6.

```

1      #Get arguments and made a dict where key and value same thing
2      def trans(a, s):
3          slov = {a: s}
4          return slov
5      u =
6      li1 = []
7      ress = {}
8      print('Enter data(5 times): ')
9      while len(li1) != 5:
10         u = input()
11         li1.append(u)
12         print(li1)
13         for i in range(5):
14             res = trans(li1[i], li1[i])
15             ress.update(res)
16         print(ress)

```

Рисунок 6 - Код для задания 6

Пояснение работы программы:

- 1) Создаём функцию `trans(a, s)`. в ней создаём словарь `slov` со значением `{s это значение, а это ключ}`. Возвращаем словарь `slov`.
- 2) Создаём пустую переменную `u` с типом `str`, список `li1` и словарь `ress`.
- 3) Выводим сообщение что нужно написать значение `n` раз, в моём случае `5`
- 4) Создаём цикл который продолжается пока длина списка `li1` не будет равна нужному количеству, в моём случае `5`
- 5) Выводим список `li1`
- 6) Создаём цикл `for i in range(5)`, в нём придаём значение словарю `res` значения от функции `trans(li1[i], li1[i])`, обновляем словарь `ress` если в нём нет такого значения.
- 7) Выводим результат командой `print()`

Программа успешно создаёт словарь где ключ-значение одинаковый аргумент.

1.7 Задание 7

В данном задании сделан код, который имитирует англо-русский словарь с переводом нескольких планет. Программа изображена на Рисунке 7.

```

1      # -*- coding: cp1251 -*-
2      # Словарь
3      tr = {
4          'Jupiter': 'юпитер',
5          'Pluto': 'плутон',
6          'Mars': 'марс',
7          'Sun': 'солнце',
8          'Moon': 'луна',
9          'Saturn': 'сатурн',
10         'Venus': 'венера',
11         'Earth': 'земля',
12         'Mercury': 'меркурий',
13         'Neptune': 'нептун',
14         'Uranus': 'уран'
15     }
16     u = input('Напишите название планеты с маленькой буквы: ')
17     rev_tr = {value: key for key, value in tr.items()}
18     if u in rev_tr:
19         print(f'По английски это - {rev_tr[u]}')
20     else:
21         print('Не найдено')

```

Рисунок 7 - Код для задания 7

Пояснение работы программы:

- 1) Создаём словарь tr с данными, где значение это английское слово, ключ его перевод
- 2) Создаём переменную u, которая равна вводимому пользователем значению через input() и пишет что и как нужно ввести
- 3) Переменная rev_tr получается значение словаря tr, но теперь значение это перевод, а ключ английское слово
- 4) Блоком if-else ищем u в rev_tr и если такое есть, то выводим перевод, иначе выводится «Не найдено»

Программа успешно выполняет свой функционал и верно отображает перевод.

1.8 Задание 8

В данном задании сделан код, который воспроизводит игру камень-ножницы-бумага-ящерица-спок в консоли. Программа изображена на Рисунке 8.

```

1  #Камень-Ножницы-Бумага-Ящерица-Спок
2  import random
3  print('''Game: Rock-Scissors-Paper-Lizard-Spock
4  Rules:
5
6  Scissors cut paper
7  Paper covers the stone
8  The stone crushes the lizard
9  the lizard poisons Spock
10 Spock breaks the scissors
11 Scissors decapitate a lizard
12 The lizard eats paper
13 Paper sets up Spock
14 Spock vaporizes the stone
15 The stone breaks the scissors
16
17 To close program enter "quit"
18 I will remind it for you every 10 games'''')
19 item = ('stone', 'scissors', 'paper', 'lizard', 'spock')
20 rules = {
21     'stone': ['scissors', 'lizard'],
22     'scissors': ['paper', 'lizard'],
23     'paper': ['stone', 'spock'],
24     'lizard': ['spock', 'paper'],
25     'spock': ['stone', 'scissors']
26 }
27 game = 0
28 while True:
29     game += 1
30     if game % 10 == 0:
31         print('''
32             Remind, to exit enter "quit"
33             ''')
34     try:
35         print('''
36             1 - Stone
37             2 - Scissors
38             3 - Paper
39             4 - Lizard
40             5 - Spock
41             ''')
42         u= input('Enter number: ')
43         if u == "quit":
44             print('Closing game... ')
45             exit()
46         elif ',' in u or '.' in u:
47             print('Not use any float')
48             exit()
49         elif int(u) < 1 or int(u) > 5:
50             print('Error: Out of options')
51             exit()
52         else:
53             u = int(u)
54             us = item[u-1]
55             print(f'Selected: {us}')
56     except ValueError:
57         print('Error: Not number')
58         exit()
59     pc = random.choice(item)
60     print(f'{us} vs {pc}')
61     if us == pc:
62         print('Draw!')
63     elif pc in rules[us]:
64         print('User wins!')
65     else:
66         print('PC wins!')
```

Рисунок 8 - Код для задания 8

Пояснение работы программы:

- 1) Импортируем библиотеку random для дальнейшего использования в случайных выборах компьютера.
- 2) Выводим название, правила игры и как выйти из программы командой print()
- 3) Создаём список(item) с теми кого можно выбрать и словарь(rules) с тем кто кого побеждает
- 4) Создаём цикл который будет всегда активен пока не выйдет ошибка или не будет введена пользователем команда quit
- 5) Создаём переменную game со значением 0, это нужно чтобы каждую 10-ую игру выходило напоминание о том как выйти из программы
- 6) В блоке try-except выполняем саму игру, выводим командой print() выбор, через переменную i и принимаем ввод пользователя. Если i равно quit, то выполняется выход из программы, когда i < 1 или i > 6 , то выводится ошибка что ней такой функции, если ввести число с плавающей точкой, то выводится соответствующая ошибка, иначе создаём переменную us которая получает в значение наш выбор переведённый из числа в слово, из списка item
- 7) Командой print выводим наш выбор
- 8) Создаём переменную pc и даём ей случайное значение из списка item
- 9) Прописываем условия, если us равно pc, то выводится «Ничья!», если переменная pc в ключе словаря rules по значение us, то выводится «Пользователь победил!», иначе «Компьютер победил!»

Программа верно отображает доступные опции и корректно выполняет свой функционал.

1.9 Задание 9

В данном задании сделан код, который создаёт словарь, где ключом является первая буква, а значение это слова с этой буквы. Программа изображена на Рисунке 9.

```

1      #Dict, where a key - first symbol and value - words that writing from that symbol
2      a = ["apple", "pear", "banana", "kiwi", "orange", "ananas"]
3      print(a)
4      l = len(a)
5      res = {}
6      for word in a:
7          f_s = word[0]
8          if f_s not in res:
9              res[f_s] = [word]
10         else:
11             res[f_s].append(word)
12     print(res)

```

Рисунок 9 - Код для задания 9

Пояснение работы программы:

- 1) Создаётся словарь с разными словами
- 2) Командой print() выводится словарь
- 3) Переменной l даём значение длины словаря
- 4) Создаём пустой словарь res
- 5) Создаём цикл for word in a, чтобы из каждого слова взять первый символ и создать словарь, состоящий из ключа - первый символ и значения - одно слово, если такая буква уже есть, то добавляется новое слово в существующий ключ
- 6) Командой print() выводим полученный словарь

Программа успешно создаёт словарь по заданным условиям.

1.10 Задание 10

В данном задании сделан код, который создаёт словарь, в котором ключ это имя студента, а значение его средний балл, также выводит имя и средний балл студента с лучшей средней оценкой. Программа изображена на Рисунке 10.

```

1      #Made dict where studens and their avg grade, where output is student with best grade
2      li = [("Anna", [5, 4, 5]), ("Ivan", [3, 4, 4]), ("Maria", [5, 5, 5])]
3      dic = {}
4      avg = {}
5      for stud, grades in li:
6          avg = sum(grades) / len(grades)
7          dic[stud] = avg
8      print('Stud with best avg grade: ')
9      best_stud = ''
10     best_avg = 0
11     for stud, avg in dic.items():
12         if avg > best_avg:
13             best_avg = avg
14             best_stud = stud
15     print(f'{best_stud} -- {best_avg:.2f}')

```

Рисунок 10 - Код для задания 10

Пояснение работы программы:

- 1) Создаём список кортежей с именами студентов и их оценками
- 2) Создаём пустые словари dic и avg
- 3) Создаём цикл for stud, grades in li, для каждого студента находим его средний балл и записываем в переменную avg, после в словарь dic записываем средний балл каждого студента поочереди
- 4) Создаём пустую переменную best_stud типа str и best_avg типа int
- 5) Создаём цикл for stud, avg in dic.items(), в нём прописываем условие: если avg больше best_avg, то заменяем значение best_avg на проверяемый avg, а best_stud на stud
- 6) Выводим результат командой print() с функцией f-строки

7) Best_avg:.2f, а именно :.2f делает так, чтобы выводилось 2 значения после запятой

Данная программа успешно справляется с нахождением студента с лучшей средней оценкой.