

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	_____	К.Ф. Кучерчук
Ассистент		
преподавателя	_____	М.В. Водяницкий

## Задание

Выполнить задания 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.

Пример:

- 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 .....	3
1.4	Задание 4 .....	3
1.5	Задание 5 .....	4
1.6	Задание 6 .....	4
1.7	Задание 7 .....	5
1.8	Задание 8 .....	5
1.9	Задание 9 .....	6
1.10	Задание 10 .....	7

## 1 Выполнение работы

### 1.1 Задание 1

Создаём список `lst`, который наполняем числами. После при помощи цикла `for` проходимся по всем элементам, если этот элемент, переведённый в строку функцией `str()`, совпадает со строкой `'3'`, то мы этому элементу присваиваем значение 30. На рисунке 1 представлен код программы.

```
1 #task_1
2
3 lst = [2,5,3,6,9,8,4,5,30,4,6]
4 for i in range(len(lst)):
5     if str(lst[i]) == '3': lst[i] = 30
```

Рисунок 1 – Листинг программы для задания 1

### 1.2 Задание 2

Создаём список `lst`, который наполняем числами. При помощи функции `map()`, первым аргументом в который мы передаём лямбда-функцию, которая возводит аргумент в квадрат, а вторым сам список чисел, и оборачиваем в список, затем присваиваем его нашему первоначальному списку. На рисунке 2 представлен код программы.

```
1 #task_2
2
3 lst = [2,3,5,7,11]
4 lst = [*map(lambda x: x**2, lst)]
```

Рисунок 2 – Листинг программы для задания 2

### 1.3 Задание 3

Создаём список `lst`, который наполняем числами. Переменной `result` присваиваем число с точкой, полученное делением наибольшего числа из списка, полученного функцией `max()` со списком в качестве аргумента, на длину списка, полученной при помощи функции `len()` со списком в качестве аргумента. На рисунке 3 представлен код программы.

```
1 #task_3
2
3 lst = [1,9,5,7,6,2,4,5,5,15,4,55,45,64,6,56,1,54]
4 result = max(lst)/len(lst)
```

Рисунок 3 – Листинг программы для задания 3

### 1.4 Задание 4

Создаём кортеж чисел, присваивая его переменной `tpl`. Создаём новую переменную `tpl1`, в который заносим строку, полученную следующим образом: метод `join` применяется к строке и вставляет её между элементами итерируемого объекта со строками, который мы получаем применяя функцию `map` к кортежу `tpl` и функцией `str()`. Далее переменной `tpl`

тернарным оператором сортированный кортеж `tpl`(при помощи функции `sorted()`) если все элементы строки `tpl1` цифры(при помощи метода `isdigit()`), в противном случае оставляем кортеж неизменным. На рисунке 4 представлен код решения.

```
1 tpl = (1,5,4,6,2,8)
2 tpl1= ''.join(map(str,tpl))
3 tpl = sorted(tpl) if tpl1.isdigit() else tpl
4 print(tpl)
```

Рисунок 4 – Листинг программы для задания 4

## 1.5 Задание 5

Создаём словарь `pricelist`, ключами которого являются наименования товаров, а значениями их стоимость. Далее создаём переменные `maximum` и `minimum`, разницей в которых будет только применяемая функция, `min()` для минимального и `max()` для максимального. Первым аргументом передаём элементы словаря при помощи метода `items()`, а вторым ключ для функции, для того чтобы сортировка была по значению а не по ключу. Затем выводим это пользователю функцией `print()`. На рисунке 5 представлен код программы.

```
1 pricelist = {'Арбуз': 100,
2             'Ананас': 200,
3             'Кишмиш': 150,
4             'Кунжут': 300}
5 minimal = min(pricelist.items(),key=lambda x:x[1])
6 maximal = max(pricelist.items(),key=lambda x:x[1])
7 print(minimal,maximal)
```

Рисунок 5 – Листинг программы для задания 5

## 1.6 Задание 6

Список `lst` содержит элементы данных произвольных типов. Создаём словарь `dct`. Далее циклом `for` проходимся по всем элементам списка. Внутри него открывается блок `try except` для отлова ошибок. Ошибка же может быть при попытке создания элемента словаря с ключом `i` и значением `i`, так как существуют ограничения для его ключей. Если ошибка возникает, то мы просто идём на следующую итерацию. На рисунке 6 представлен код программы.

```
1 #Task_6
2
3 lst = [11, 'kfl', True, 5.5, 'dawd','dcvf', [1,2], None]
4 dct = dict()
5 for i in lst:
6     try:
7         dct[i] = i
8     except:
9         pass
10 print(dct)
```

Рисунок 6 – Листинг программы для задания 6

## 1.7 Задание 7

Создаём словарь `trans_dict`, в котором в качестве ключей используются английские слова, а в качестве значений соответствующие им переводы. Далее при помощи функции `input` запрашиваем у пользователя строку на ввод и присваиваем её переменной `word`. Далее в блоке `try except` пробуем обратиться к элементу словаря с ключом `word` и вывести информацию, в случае ошибки выводится сообщение о том, что элемента нет в словаре. На рисунке 7 представлен код программы.

```

1 #Task_7
2
3 trans_dict = {
4     'apple': 'Яблоко',
5     'peach': 'Груша',
6     'ball': 'Мяч',
7     'snow': 'Снег',
8     'wood': 'Дерево'
9 }
10
11 word = input('Введите слово: ')
12 try:
13     print(f'Перевод слова {word} : {trans_dict[word.lower()]}')
14 except:
15     print(f'Слово {word} отсутствует в словаре')
```

Рисунок 7 – Листинг программы для задания 7

## 1.8 Задание 8

Импортируем модуль `random`, который позволяет компьютеру делать случайный выбор. Создает словарь с правилами игры. Каждый ключ (например, 'ножницы') содержит список того, что он побеждает. Создаем список `choices` всех возможных выборов ['ножницы', 'бумага', 'камень', 'ящерица', 'спок']. `rules.keys()` берет все ключи из словаря `rules`. Создаём функцию `print_choices` для вывода выборов игрока и компьютера. `global` означает, что функция использует переменные, созданные вне её. Далее Компьютер случайно выбирает один из вариантов (ножницы/бумага/камень/ящерица/спок) функцией `choice` модуля `random`. Потом показываем пользователю список доступных знаков. Запрашиваем у пользователя строку на ввод в переменную `player_choice`. Далее в блоке `try except` ищем выборы пользователя и ИИ в словаре и выводим на экран сообщение о результате, предварительно вызвав функцию `print_choices`. Если выбора игрока нету в ключах словаря, то выводим сообщение об этом на экран. На рисунке 8 представлен код программы.

```

1 #Task_8
2 import random
3
4 rules = {
5     'ножницы' : ['бумага', 'ящерица'],
6     'бумага' : ['спок', 'камень'],
7     'камень' : ['ножницы', 'ящерица'],
8     'ящерица' : ['спок', 'бумага'],
9     'спок' : ['камень', 'ножницы']
10 }
11
12
13 choices = [x for x in rules.keys()]
14
15 def print_choices():
16     global ai_choice
17     global player_choice
18     print(f"Выбор ии : {ai_choice}")
19     print(f'Выбор игрока : {player_choice}')
20
21 ai_choice = random.choice(choices)
22
23 print('Список знаков')
24 for i in choices:
25     print(i.capitalize())
26 player_choice = input('Выберите знак : ')
27
28 try:
29     if player_choice not in choices: raise
30
31     if player_choice in rules[ai_choice]:
32         print_choices()
33         print("Ты проиграл")
34
35     elif ai_choice in rules[player_choice]:
36         print_choices()
37         print('Ты победил')
38     else:
39         print_choices()
40         print('Ничья')
41 except:
42     print('Неверный знак')

```

Рисунок 8 – Листинг программы для задания 8

## 1.9 Задание 9

У нас есть список фруктов (слов) lst. Создаётся пустой словарь letter\_dict, который будет хранить группировку слов. Код начинает перебирать каждый элемент списка по порядку циклом for. Для каждого слова берётся его первая буква при помощи индекса [0]. Далее проверяется, есть ли уже такая первая буква в качестве ключа в словаре: Если буквы ещё нет в словаре, то создаётся для неё пустой список [], если буква уже есть, то ничего не меняется. Слово добавляется в список методом append, соответствующий его первой букве. На рисунке 9 представлен код программы.



```

1 #Task_9
2
3 lst = ["яблоко", "груша", "банан", "киви", "апельсин", "ананас",
4        'киви']
5 letter_dict = dict()
6
7 for x in lst:
8     letter = x[0]
9     if letter not in letter_dict.keys():
10         letter_dict[letter] = []
11         letter_dict[letter].append(x)
12
13 print(letter_dict)

```

Рисунок 9 – Листинг программы для задания 9

### 1.10 Задание 10

У нас есть список кортежей. Каждый кортеж содержит имя студента и список его оценок. Далее создаём новый словарь `stud_dict` для кортежей студентов и его средней оценки. После циклом `for` проходимся по первоначальному списку и заносим в него данные о студентах и их средней оценке. Далее заводим переменную `sorted_students`, содержащая отсортированный словарь по значениям. Далее в переменные `best_student_name`, `best_student_score` заносим значения первого кортежа, и `f` строкой выводим результат. На рисунке 10 представлен код программы.

```

1 #Task_10
2
3 stud_list = [("Анна", [5, 4, 5]), ("Иван", [3, 4, 4]), ("
4     Мария", [5, 5, 5])]
5 stud_dict = dict()
6
7 for name, grade in stud_list:
8     stud_dict[name] = (sum(i for i in grade) / len(grade)).
9         __round__(3)
10 sorted_students = sorted(stud_dict.items(), key=lambda
11     student: student[1], reverse=True)
12 best_student_name, best_student_score = sorted_students[0]
13 print(f'Лучший ученик: {best_student_name}, егоёё() средний
14     балл: {best_student_score}')

```

Рисунок 10 – Листинг программы для задания 10

Спасибо за внимание !