

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 _____ С.Е. Казюта
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: ““python (’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: ““python [”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: ““txt Мария имеет наивысший средний балл: 5.0 ““

Содержание

1 Выполнение работы

1.1 Задание 1

Задача данной программы это найти в списке число 3 и заменить на 30.

Алгоритм работы:

- 1) Программа заменяет в списке число 3 на 30 .
- 2) Программа выводит новый список.

```
1 listy = [1, 54, 87, 23, 65, 76, 987, 3, 1, 11]
2 new_listy = [str(i).replace(' 3','30") for i in listy]
3 print(new_listy)
```

Рисунок 1 – Листинг программы для задания 1

Ключевые элементы кода:

- 1) listy - сам список.
- 2) new-listy - мы создаём новый список
- 3) replace - метод меняет каждый элемент "3" в новом списке на "30".
- 4) print - программа выводит новый список

1.2 Задание 2

Задача данной программы это превратить список из 5 целых чисел в список квадратов этих чисел.

Алгоритм работы:

- 1) Программа заменяет в списке каждое число на его же, но во второй степени.
- 2) Программа выводит новый список.

```
1 listy = [1,4,7,11,9]
2 new_listy = [i**2 for i in listy]
3 print(new_listy)
```

Рисунок 2 – Листинг программы для задания 2

Ключевые элементы кода:

- 1) listy - сам список.
- 2) new-listy - мы создаём новый список
- 3) i^{**2} - меняет каждый элемент в новом списке на его же, но во второй степени.
- 4) print - программа выводит новый список

1.3 Задание 3

Задача данной программы это найти наибольшее из чисел списка и разделить его на длину списка.

Алгоритм работы:

- 1) Программа находит наибольший элемент списка.
- 2) Программа делит наибольший элемент списка на длинну списка.
- 3) Вывод результата.

```

1 listy = [1, 54, 87, 23, 65, 76, 987, 3, 1, 11]
2 max = listy[0]
3 for i in listy:
4     if i > max:
5         max = i
6 print(max/len(listy))

```

Рисунок 3 – Листинг программы для задания 3

Ключевые элементы кода:

- 1) listy - сам список.
- 2) max - определяем переменную под наибольший элемент списка.
- 3) for - цикл проходит по всему списку и определяет наибольший её элемент.
- 4) print - программа выводит максимальный элемент списка делённый на длинну списка.

1.4 Задание 4

Задача данной программы это сортировка кортежа и при этом, если хотя бы один элемент не является числом, то кортеж остается неизменным.

Алгоритм работы:

- 1) Программа проверяет есть ли в кортеже не числа.
- 2) Сортирует кортеж если в нём нет не чисел.
- 3) Выводит кортеж.

```

1 import string
2 corty = (1, 14, 5, 9, 'sadas', 32)
3 flagy = True
4 for i in corty:
5     if [j in str(i) for j in string.ascii_letters]:
6         flagy = False
7 if flagy == True:
8     new_corty = tuple(sorted(corty))
9 else:
10    new_corty = corty
11 print(new_corty)

```

Рисунок 4 – Листинг программы для задания 4

Ключевые элементы кода:

- 1) corty - сам кортеж.
- 2) flagy - флагок который пригодится нам позже.
- 3) for - проверяет есть ли в кортеже не число отмечая это с помощью flagy.
- 4) if else - если в кортеже нету не чисел то проводит сортировку.

5) print - выводит кортеж.

1.5 Задание 5

Задача данной программы это найти товар с минимальной и максимальной ценой.

Алгоритм работы:

- 1) Программа проверяет каждый элемент словаря в поисках самого маленького элемента и самого большого.
- 2) Программа выводит какой список является самым большим и самым маленьким.

```

1 dicty = {"milk": 100, "vine": 1000, "meat": 300, "cheese": 350}
2 mini, maxi, min_name, max_name = dicty["milk"], dicty["milk"]
3     , "milk", "milk"
4 for i, j in dicty.items():
5     if j < mini:
6         mini = j
7         min_name = i
8     if j > maxi:
9         maxi = j
10        max_name = i
10 print(f'min -> {min_name}, max -> {max_name}')

```

Рисунок 5 – Листинг программы для задания 5

Ключевые элементы кода:

- 1) dicty - сам словарь.
- 2) mini maxi min-name max-name - стартовые значения которые нужны для запуска программы.
- 3) for - прогоняет элементы словаря через if который в свою очередь определяет является элемент минимумом и максимумом.
- 4) print - выводит минимум и максимум.

1.6 Задание 6

Задача данной программы это на основе этого списка произвольных элементов создать словарь, где каждый элемент списка будет и ключом, и значением.

Алгоритм работы:

- 1) Программа создаёт а после сортирует список произвольных элементов.
- 2) Программа определяет каждый элемент списка как ключом так и значением в свежесозданном словаре.
- 3) Программа выводит словарь.

```

1 import random, string
2 listy = {string.ascii_lowercase[random.randint(0, len(string.
    ascii_lowercase))-1] for i in range(20)}
3 listy = sorted(listy)
4 dicty = dict()
5 for i in range(len(listy)):
6     dicty[listy[i]] = listy[i]
7 print(dicty.items())

```

Рисунок 6 – Листинг программы для задания 6

Ключевые элементы кода:

- 1) listy - программа создаёт а после сортирует список произвольных элементов.
- 2) sorted() - программа сортирует список.
- 3) dicty - программа создаёт пустой словарь.
- 4) for - программа определяет каждый элемент listy как ключом так и значением в dicty.
- 5) print - программа выводит dicty.

1.7 Задание 7

Задача данной программы это получать на ввод русское слово и результатом выдавать перевод на английский.

Алгоритм работы:

- 1) Программа получает от пользователя русское слово.
- 2) Программа находит перевод слова в базе данных.
- 3) Программа ввыводит перевод.

```

1 dicty = {
2     'apple': 'яблоко',
3     'pencil': 'карандаш',
4     'milk': 'молоко',
5     'chocolate': 'шоколад',
6     'coffee': 'кофе',
7     'number': 'число',
8     'family': 'семья',
9     'meme': 'мем'
10 }
11 ino = input(f'Words we can translate: {dicty.values()},
    please enter your word: ')
12 for i,j in dicty.items():
13     if j == ino:
14         print(i)
15         break

```

Рисунок 7 – Листинг программы для задания 7

Ключевые элементы кода:

- 1) dicty - база данных в формате словаря.
- 2) input - ввод пользователя.
- 3) for - программа проходится по dicty в поисках перевода input с помощью if.

- 4) print - программа выводит перевод input.

1.8 Задание 8

Задача данной программы это 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.

Алгоритм работы:

- 1) Программа получает от пользователя: Камень, Ножницы, Бумага, Ящерица, Спок.
- 2) Программа случайный образом выбирает случайный вариант.
- 3) Программа выводит кто победил.

```

1 import random
2 user_inp = input('Выберите: Камень Ножницы Бумага Ящерица
Спок (1-5): ')
3 enemy_inp = random.randint(1,5)
4 who_beats_who = {'1':{2,4}, '2':{3,4}, '3':{1,5}, '4':{3,5},
'5':{1,2}}
5 if enemy_inp in who_beats_who[user_inp]:
6     print('Вы победили!')
7 elif user_inp == enemy_inp:
8     print('ничья!')
9 else:
10    print('Вы проиграли! ')

```

Рисунок 8 – Листинг программы для задания 8

Ключевые элементы кода:

- 1) input() - ввод пользователя.
- 2) enemy-inp randint() - случайное значение.
- 3) who-beats-who - словарь с игровой логикой.
- 4) if - определяет кто победил на основе информации из словаря.
- 5) print() - выводит победителя(если таковой иммется).

1.9 Задание 9

Задача данной программы это создать словарь на основе списка слов в котором ключом будет являться первая буква слова, а значением - список всех слов, начинающихся с этой буквы.

Алгоритм работы:

- 1) Программа создаёт ключ из каждой первой буквы.
- 2) Программа присваивает соответствующие слова как значения.
- 3) Программы выводит словарь.

```

1 listy = ["яблоко", "груша", "банан", "киви", "апельсин", "ананас"]
2 new_listy = {}
3 for i in [j[0] for j in listy]:
4     new_listy[i] = []
5 for i in new_listy:
6     for k in listy:
7         if i == k[0]:
8             new_listy[i].append(k)
9 print(new_listy.items())

```

Рисунок 9 – Листинг программы для задания 9

Ключевые элементы кода:

- 1) listy - список слов.
- 2) new-listy - словарь.
- 3) первый for - создаёт ключ из каждой первой буквы.
- 4) двойной for - для каждого ключа, присваивает как значение слово начинающееся с буквы равной его значению.
- 5) print() - выводит словарь.

1.10 Задание 10

Задача данной программы создать словарь и в нём найти студента с наибольшей средней оценкой и вывести его имя и средний балл.

Алгоритм работы:

- 1) Программа создаёт словарь на основе списка.
- 2) Программа находит студента с наибольшей средней оценкой и выводит его имя и средний балл.

```

1 listy = [("Анна", [5, 4, 5]), ("Иван", [3, 4, 4]), ("Мария",
2 [5, 5, 5])]
3 new_listy = {}
4 for i in listy:
5     new_listy[i[0]] = sum(i[1])/len(i[1])
6 for i, j in new_listy.items():
7     max = new_listy["Анна"]
8     if j > max:
9         max_name = i
10        max = j
10 print(f'{max_name} имеет наивысший средний балл: {max}')

```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

- 1) listy - список с учениками и их оценками.
- 2) new-listy - пустой словарь.
- 3) первый for - заполняет словарь информацией из списка.
- 4) второй for - находит студента с наибольшей средней оценкой с помощью if.

5) print() - выводит студента с наибольшей средней оценкой на экран.