

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 №4  
  
по дисциплине  
«Информатика и программирование»

Студент  
гр. БИС-25-3

---

Л.С.Лавриненко

Ассистент  
преподавателя

---

М. В. Водяницкий

Владивосток 2025

## Задание

Выполнить задание 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:

- 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, а значением - список всех слов, начинающихся с этой буквы.

**Задание 10.** Дан список кортежей, где каждый кортеж содержит имя студента и его оценки. Необходимо создать словарь, где ключ - имя студента, значение - его средняя оценка, и найти студента с наибольшей средней оценкой.

\*\*\*

## 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 .....	5
1.6 Задание 6 .....	6
1.7 Задание 7 .....	7
1.8 Задание 8 .....	8
1.9 Задание 9 .....	9
1.10 Задние 10 .....	9

## 1 Выполнение работы

### 1.1 Задание 1

В данном задании необходимо было написать программу, которая определяла бы как будет вести себя кондиционер. Если температура, которую ввел пользователь 20 градусов и выше, то выводится надпись, что кондиционер выключен, если меньше - включен. На рисунке 1 представлен код программы.

```
s='1 2 3 4 5 6 7 8 9 10'
while '3' in s :
    if '3' in s :s=s.replace('3','30',1)
    print(s)
    break
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Создается строка `s` с числами от 1 до 10 через пробел.
- 2) Запускается цикл `while`, который работает пока в строке есть символ '3'.
- 3) Внутри цикла проверяется условие: если в строке есть символ '3'.
- 4) Если условие выполняется, первое вхождение '3' заменяется на '30'.
- 5) Выводится измененная строка.
- 6) Команда `break` прерывает цикл после первой итерации.

### 1.2 Задание 2

В данном задании необходимо было написать программу, которая запрашивала бы у пользователя номер месяца и выводила к какому сезону этот месяц относится. На рисунке 2 представлен код программы.

```
b = [2, 4, 6, 8, 10]
c = []
for i in range(5):
    c.append(b[i] * b[i])
print(b)
print(c)
```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Создается список из 5 целых чисел.
- 2) Создается пустой список `squares` для хранения квадратов.

- 3) Запускается цикл **for**, который выполняется 5 раз.
- 4) На каждой итерации берется число из списка по текущему индексу.
- 5) Число умножается само на себя для получения квадрата.
- 6) Квадрат добавляется в список **squares**.
- 7) Выводятся исходные числа и их квадраты.

### 1.3 Задание 3

В этом задании необходимо было запросить у пользователя возраст собаки (в годах), который бы программа переводила в человеческие годы. Также программа должна была корректно обрабатывать входные данные и выводить соответствующие сообщения об ошибках: если вводится не число, если вводится число меньше, если вводится число большее 22. На рисунке 3 представлен код программы.

```
d = [12, 45, 23, 67, 34]
max_num = d[0]
for num in d:
    if num > max_num:
        max_num = num
result = max_num / 5
print(d)
print(max_num)
print(result)
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Создается список различных целых чисел.
- 2) Переменной **max\_num** присваивается значение первого элемента списка.
- 3) Запускается цикл **for**, который перебирает все числа в списке.
- 4) На каждое число проверяется условие: больше ли оно текущего максимума.
- 5) Если число больше, оно становится новым максимальным.
- 6) После цикла максимальное число делится на количество элементов .
- 7) Выводятся исходный список, максимальное число и результат деления.

## 1.4 Задание 4

В данном задании необходимо было запросить у пользователя число и с помощью свойств деления числа на цело на 6 (последняя цифра четная, сумма всех цифр делиться на 3) понять делится ли введенное число на 6. На рисунке 4 представлен код программы.

```
t = (5, 2, 8, 1)
all_ok = True
for x in t:
    if not type(x) == int:
        all_ok = False

if all_ok:
    lst = list(t)
    n = len(lst)
    for i in range(n):
        for j in range(i+1, n):
            if lst[i] > lst[j]:
                lst[i], lst[j] = lst[j], lst[i]
    print(tuple(lst))
else:
    print(t)
```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Создается кортеж из нескольких элементов.
- 2) Устанавливается флаг **all\_numbers** в **True**.
- 3) Запускается цикл, проверяющий каждый элемент кортежа.
- 4) Если элемент не является целым числом, флаг меняется на **False**.
- 5) Если все элементы - числа, кортеж преобразуется в список.
- 6) Выполняется сортировка пузырьком для упорядочивания элементов.
- 7) Отсортированный список преобразуется обратно в кортеж.
- 8) Выводится отсортированный кортеж или исходный, если были нечисловые элементы.

## 1.5 Задание 5

В следующем задании необходимо было запросить у пользователя пароль, который бы программа проверяла на надежность по следующим критериям: есть

заглавные буквы латиницы, строчные буквы латиницы, числа, специальные знаки и пароль содержит больше 8 символов. Если пароль не содержит всех условий, то выводится, что он ненадежный, если содержит, то выводится, что он надежный. На рисунке 5 представлен код программы.

```
shop = {"молоко": 50, "хлеб": 30, "сыр": 200}
min_price = 999
max_price = 0
for item in shop:
    price = shop[item]
    if price < min_price:
        min_price = price
        min_item = item
    if price > max_price:
        max_price = price
        max_item = item
print(min_item, min_price)
print(max_item, max_price)
```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Создается словарь товаров с ценами.
- 2) Устанавливаются значения для минимальной и максимальной цены.
- 3) Запускается цикл по всем товарам в словаре.
- 4) Для каждого товара получается его цена.
- 5) Сравнивается цена с текущими минимумом и максимумом.
- 6) Если цена меньше минимальной, запоминаются новая цена и товар.
- 7) Если цена больше максимальной, запоминаются новая цена и товар.
- 8) Выводятся товары с минимальной и максимальной ценой.

## 1.6 Задание 6

В следующем задании необходимо написать программу, которая определяет, является ли введенный пользователем год високосным. На рисунке 6 представлен код программы.

```
lst = ["a", "b", "c"]
d = {}
for x in lst:
    d[x] = x
print(d)
```

### Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Создается список произвольных элементов.
- 2) Создается пустой словарь **result\_dict**.
- 3) Запускается цикл по всем элементам списка.
- 4) Для каждого элемента добавляется запись в словарь.
- 5) Элемент становится одновременно ключом и значением.
- 6) Выводится созданный словарь.

### 1.7 Задание 7

В данном задании необходимо было без специальных функций из введенных трех чисел вывести наименьшее. На рисунке 7 представлен код программы.

```

words = {"apple": "яблоко", "dog": "собака"}
r_words = {}
for eng in words:
    rus = words[eng]
    r_words[rus] = eng

w = input("Введите слово: ")
if w in r_words:
    print(r_words[w])
else:
    print("Нет")

```

### Рисунок 7 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Создается словарь перевода с английского на русский.
- 2) Создается пустой словарь для обратного перевода.
- 3) Запускается цикл по английским словам.
- 4) Для каждого слова создается обратная запись: русское → английское.
- 5) Пользователь вводит русское слово.
- 6) Проверяется наличие слова в словаре перевода.
- 7) Если слово найдено, выводится английский перевод.
- 8) Если слово не найдено, выводится сообщение об ошибке.

## 1.8 Задание 8

В следующем задании необходимо было написать программу, которая запрашивала бы у пользователя сумму покупки, а позже выводила скидку, основанную на его сумме, и итоговую сумму с учетом скидки. На рисунке 8 представлен код программы.

```

import random
options = ["камень", "ножницы", "бумага", "ящерица", "спок"]
player = input("Выбери: ")
comp = random.choice(options)
print("Компьютер:", comp)

if player == comp:
    print("Ничья")
elif player == "камень" and comp in ["ножницы", "ящерица"]:
    print("Ты выиграл")
elif player == "ножницы" and comp in ["бумага", "ящерица"]:
    print("Ты выиграл")
elif player == "бумага" and comp in ["камень", "спок"]:
    print("Ты выиграл")
elif player == "ящерица" and comp in ["бумага", "спок"]:
    print("Ты выиграл")
elif player == "спок" and comp in ["камень", "ножницы"]:
    print("Ты выиграл")
else:
    print("Компьютер выиграл")

```

Рисунок 8 – Листинг программы для задания 8

- 1) Пояснение работы программы:
- 2) Импортируется модуль **random** для случайного выбора.
- 3) Создается список возможных вариантов игры.
- 4) Пользователь вводит свой выбор.
- 5) Компьютер случайно выбирает свой вариант.
- 6) Выводится выбор компьютера.
- 7) Проверяется условие ничьи.
- 8) Проверяются все варианты победы игрока по правилам игры.
- 9) Выводится результат: кто победил или ничья.

## 1.9 Задание 9

В данном задании необходимо было написать программу, которая запрашивала бы у пользователя целое число от 0 до 23 и выводила к какой части дня это время относится. На рисунке 9 представлен код программы.

```

fruits = ["яблоко", "груша", "банан", "киви", "апельсин", "ананас"]
res = {}
for fruit in fruits:
    letter = fruit[0]
    if letter not in res:
        res[letter] = []
    res[letter].append(fruit)
print(res)

```

Рисунок 9 – Листинг программы для задания 9

Пояснение работы программы:# Задание 9 - Пояснение работы программы:

- 1) Создается список слов.
- 2) Создается пустой словарь для результата.
- 3) Запускается цикл по всем словам в списке.
- 4) Для каждого слова берется первая буква.
- 5) Проверяется наличие буквы в словаре.
- 6) Если буквы нет, создается пустой список для нее.
- 7) Слово добавляется в список соответствующей буквы.
- 8) Выводится сгруппированный словарь.

## 1.10 Задание 10

В последнем задании необходимо было создать программу, которая определяла бы является ли введенное пользователем число простым или составным. На рисунке 10 представлен код программы.

```
students = [("Анна", [5,4,5]), ("Иван", [3,4,4]), ("Мария", [5,5,5])
best_name = ""
best_avg = 0

for name, marks in students:
    total = 0
    for mark in marks:
        total += mark
    avg = total / 3
    if avg > best_avg:
        best_avg = avg
        best_name = name

print(best_name, best_avg)
```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

- 1) Создается список с данными студентов.
- 2) Создается пустой словарь для средних оценок.
- 3) Устанавливаются начальные значения для лучшего студента.
- 4) Запускается цикл по всем студентам.
- 5) Для каждого студента вычисляется сумма оценок.
- 6) Вычисляется средняя оценка.
- 7) Добавляется запись в словарь: имя → средняя оценка.
- 8) Проверяется, является ли эта оценка лучшей.
- 9) Выводятся все средние оценки и лучший студент.