

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-25-3 _____ Герцов Д.Е.
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:

- 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. Написать программу, которая определяет, является ли введенное число простым. Число называется простым, если оно больше 1 и делится только на 1 и само себя. Программа должна корректно обрабатывать некорректный ввод и выводить соответствующие сообщения об ошибках.

Содержание

1 Выполнение работы	3
1.1 Задание 1	3
1.2 Задание 2	3
1.3 Задание 3	3
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	8
1.10 Задание 10	9

1 Выполнение работы

1.1 Задание 1

Функция изменяет оригинальный список(работает «in-place»). Щаменяется только первое вхождение числа 3. Используется прямой доступ по индексу для модификации элементов. На рисунке 1 представлен полученный код.

```
1 def task_1():
2     spisok = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
3     for i in range(len(spisok)):
4         if spisok[i] == 3:
5             spisok[i] = 30
6     print(spisok)
```

Рисунок 1 – Листинг программы для задания 1

- 1) Создание списка
- 2) Цикл по индексам
- 3) Проверка и замена
- 4) Выведение результата

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.2 Задание 2

Создаем список чисел от 1 до 5. Проходим по всем индексам списка. Каждый элемент возводим в квадрат. Функция ничего не возвращает и не выводит результат. На рисунке 2 представлен полученный код.

```
1 def task_2():
2     spisok = [1,2,3,4,5]
3     for i in range(len(spisok)):
4         spisok[i] = spisok[i]**2
```

Рисунок 2 – Листинг программы для задания 2

- 1) Создание списка
- 2) Цикл по индексам
- 3) Возведение в квадрат
- 4) Конечный результат изменяется

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.3 Задание 3

Создаем список из 7 чисел. Выводим результат деления максимального элемента на длину списка. На рисунке 3 представлен полученный код.

```

1 def task_3():
2     spisok = [23,55,12,23,7,8,100]
3     print(max(spisok)/len(spisok))

```

Рисунок 3 – Листинг программы для задания 3

- 1) Создание списка
- 2) Вычисление максимального элемента
- 3) Вычисление длины списка
- 4) Деление

После выполнение программы в зависимости от введеного числа, будет получен соответствующий ответ

1.4 Задание 4

Создаем счетчик, создаем кортеж, перебираем элементы, проверяем тип, считаем числа, сравниваем счетчик с длиной, выводим результат. На рисунке 4 представлен полученный код.

```

1 def task_4():
2     schet = 0
3     kortej = (1, "2", 3, "попа", 66, 43)
4     for i in kortej:
5         if type(i) is int:
6             schet += 1
7     if schet == len(kortej):
8         print('Все элементы числа')
9     else:
10        print("В кортежи не только числа")

```

Рисунок 4 – Листинг программы для задания 4

- 1) Создание счетчика и кортежа (1, "2 3, "попа 66, 43)
- 2) Перебор каждого элемента кортежа
- 3) Проверка типа элемента (int или нет)
- 4) Подсчет количества целых чисел (4 из 6)
- 5) Сравнение счетчика с длиной кортежа
- 6) Вывод «В кортеже не только числа»

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.5 Задание 5

Создаем словарь, находим товар с максимальной ценой, находим товар с минимальной ценой, выводим результаты. На рисунке 5 представлен полученный код.

```

1 def task_5():
2     dicto = {"Молоко":1000, "Пепси кола-":1, "Мясо козла":9999}
3     highest_price = max(dicto, key = dicto.get)
4     lowest_price = min(dicto, key = dicto.get )
5     print("Самый дорогой товар:", highest_price,"-", dicto[
highest_price])
6     print("Самый дешевый товар:", lowest_price,"-", dicto[
lowest_price])

```

Рисунок 5 – Листинг программы для задания 5

- 1) Создание словаря товаров и цен
- 2) Поиск товара с максимальной ценой (Мясо козла – 9999)
- 3) Поиск товара с минимальной ценой (Пепси-кола – 1)
- 4) Вывод самого дорогого товара
- 5) Вывод самого дешевого товара

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.6 Задание 6

Создаем список, создаем пустой словарь, перебираем элементы списка, добавляем каждый элемент как ключ и значение. На рисунке 6 представлен полученный код.

```

1 def task_6():
2     list = ["propang", "клертон", 'флиртон', 123, 444, 323]
3     dict = {}
4     for i in list:
5         dict[i] = i
6     print(dict)

```

Рисунок 6 – Листинг программы для задания 6

- 1) Создание списка [«propang», «клертон», «флиртон», 123, 444, 323]
- 2) Создание пустого словаря
- 3) Перебор каждого элемента списка
- 4) Добавление элемента как ключа и значения в словарь
- 5) Формирование словаря элемент: элемент
- 6) Вывод готового словаря

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.7 Задание 7

Создаем словарь переводов, получаем русское слово от пользователя, ищем английский перевод, выводим результат или сообщение об ошибке. На рисунке 7 представлен полученный код.

```

1 def task_7():
2     rus = {"Russian": "Русский", "Doctor": "Доктор", "Piggy": "
Свинка", "cucumber": "Огурец"}
3     a = input("Введите русские слова:")
4     c = None
5     for key, value in rus.items():
6         if value == a:
7             c = key
8     if c:
9         print(c)
10    else:
11        print("В словаре отсутствует перевод этого слова")

```

Рисунок 7 – Листинг программы для задания 7

- 1) Создание словаря английский→русский
- 2) Получение русского слова от пользователя
- 3) Перебор пар ключ-значение в словаре
- 4) Поиск совпадения с введенным словом
- 5) Вывод английского перевода или сообщения об ошибке

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.8 Задание 8

Создаем список вариантов, получаем выбор пользователя, генерируем случайный выбор компьютера, проверяем условия победы, выводим результат. На рисунке 8 представлен полученный код.

```

1 import random
2
3 def task_8():
4     data = ["Ножницы", "Бумага", "Ящирица", "Камень", "Спок"]
5     HumanInput = input("Введите\nКаменьНожницыБумагаЯщирицаСпок----, что то- одно: ")
6     if HumanInput in data:
7         rand = random.choice(data)
8         if (HumanInput == "Ножницы" and rand == "Бумага") or
9             \
10             (HumanInput == "Бумага" and rand == "Камень") \
11             or (HumanInput == "Камень" and rand == "Ящирица") \
12             \
13             or (HumanInput == "Ящирица" and rand == "Спока") \
14             or (HumanInput == "Спок" and rand == "Ножницы") \
15             or (HumanInput == "Ножницы" and rand == "Ящирица") \
16             \
17             or (HumanInput == "Ящирица" and rand == "Бумага") \
18             or (HumanInput == "Бумага" and rand == "Спок") \
19             or (HumanInput == "Спок" and rand == "Камень") \
20             or (HumanInput == "Камень" and rand == "Бумага"):
21                 print("Вы победили")
22             else:
23                 print("Лошпед продул")
24         else:
25             print("Нету таких значений из перечисленных")

```

Рисунок 8 – Листинг программы для задания 8

- 1) Создание списка возможных вариантов
- 2) Получение выбора пользователя
- 3) Генерация случайного выбора компьютера
- 4) Проверка всех условий победы
- 5) Определение победителя или ничьи
- 6) Вывод результата игры

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.9 Задание 9

Создаем список слов, создаем пустой словарь, перебираем слова, извлекаем первую букву, добавляем слово в список соответствующей буквы. На рисунке 9 представлен полученный код.

```
1 def task_9():
2
3     words = ["яблоко", "груша", "банан", "киви", "апельсин", "ананас"]
4     result = {}
5
6     for word in words:
7
8         first_letter = word[0]
9
10        if first_letter not in result:
11            result[first_letter] = []
12
13        result[first_letter].append(word)
14
15
16 print(result)
```

Рисунок 9 – Листинг программы для задания 9

- 1) Создание списка слов
- 2) Создание пустого словаря для результата
- 3) Перебор каждого слова из списка
- 4) Извлечение первой буквы слова
- 5) Проверка наличия буквы в словаре
- 6) Добавление слова в список соответствующей буквы
- 7) Формирование словаря буква: [слова]

После выполнение программы в зависимости от введеного числа, будет получен нужный результат

1.10 Задание 10

Создаем список студентов с оценками, вычисляем средние оценки, находим студента с лучшим средним баллом, выводим результаты. На рисунке 10 представлен полученный код.

```

1 def task_10():
2     students = [("Анна", [5, 4, 5]), ("Иван", [3, 4, 4]), ("Мария", [5, 5, 5])]
3     average_grades = {}
4
5     for name, grades in students:
6         average = sum(grades) / len(grades)
7         average_grades[name] = average
8
9     print("Словарь со средними оценками:")
10    print(average_grades)
11
12
13    best_student = None
14    best_average = 0
15
16    for name, average in average_grades.items():
17        if average > best_average:
18            best_average = average
19            best_student = name
20
21    print(f"\Студент с наибольшей средней оценкой:")
22    print(f"Имя: {best_student}, Средний балл: {best_average}")
23

```

Рисунок 10 – Листинг программы для задания 10

- 1) Создание списка студентов с оценками
- 2) Вычисление средней оценки для каждого студента
- 3) Запись средних оценок в словарь
- 4) Поиск студента с максимальной средней оценкой
- 5) Вывод словаря со средними оценками
- 6) Вывод лучшего студента и его балла