

Л. р. 7 Словари. Кортежи.

Ч. 1 Словари

Цель работы: Изучение словарей в Python.

Словари в Python – неупорядоченные коллекции произвольных элементов с доступом по ключу. Словарь хранит коллекцию элементов, где каждый элемент имеет уникальный ключ и связанное с ним значение. Определение словаря имеет следующий вид:

```
dictionary = { ключ1: значение1, ключ2: значение2, ... }
```

Определение пустого словаря

```
user = {} # Вариант 1. С помощью фигурных скобочек
```

```
user = dict() # Вариант 2. С помощью создания экземпляра класса dict
```

Определение словаря со значениями

```
user = {"name": "Иванов Е.С.", "age": 35} # Вариант 1. ключами могут быть строки
```

```
user = {1: "Иванов Е.С.", 2: "Петров А.С."} # Вариант 2. ключами могут быть числа
```

Получение и изменение элементов словаря

Для обращения к элементам словаря необходимо после его названия указать в квадратных скобках ключ элемента. Ниже приведен пример получения и изменения элементов словаря:

```
user = {  
    "name": "Иванов Е.С.",  
    "age": 35}  
  
# Получение элемента с ключом "name"  
  
print(user["name"]) # Иванов Е.С.  
  
# Установка значения элемента с ключом "name"  
  
user["name"] = "Сидоров А.А."  
  
print(user["name"]) # Сидоров А.А.
```

Если при установке значения в словаре не окажется элемента с таким ключом, то произойдёт его добавление:

```
user["city"] = "Tomsk"
```

```
print(user["city"]) # Tomsk
```

Необходимо учитывать, что при попытке получить значение по ключу, которого нет в словаре, Python сгенерирует ошибку:

```
# Будет выведено сообщение об ошибке KeyError
```

```
user_profession = user["profession"]
```

Для того чтобы избежать этой ситуации перед обращением к элементу, необходимо проверять наличие ключа в словаре с помощью выражения «ключ» in «словарь». Если ключ имеется в словаре, то данное выражение вернет True:

```
key = "profession"
```

```
if key in user:
```

```
    user_profession = user[key]
```

```
    print(user_profession)
```

```
else:
```

```
    print("Элемент с таким ключом отсутствует в словаре")
```

Для получения элементов словаря можно использовать метод `get`, который имеет две формы:

- `get(key)`: возвращает из словаря элемент с ключом `key`. Если элемента с таким ключом нет, то возвращает значение `None`;

- `get(key, default)`: возвращает из словаря элемент с ключом `key`. Если элемента с таким ключом нет, то возвращает значение по умолчанию `default`.

```
user = { "name": "Иванов Е.С.", "age": 35 }
```

```
user_name = users.get("name")
```

```
print(user_name) # Иванов Е.С.
```

```
user_age = users.get("age", "Значение не найдено")
```

```
print(user_age) # 35
```

```
user_city = users.get("city", "Значение не найдено")
```

```
print(user_city) # Значение не найдено
```

Перебор словаря

Для перебора словаря можно воспользоваться циклом `for`. При переборе элементов мы получаем ключ текущего элемента и по нему можем получить сам элемент:

```
users = {1: "Иванов Е.С.", 2: "Петров А.С."}
```

```
for key in user:
```

```
    print(f"Number: {key} User: {users[key]} ")
```

Другой способ перебора элементов реализуется с помощью метода `items()`, который возвращает набор кортежей. Каждый кортеж содержит ключ и значение, которые мы можем получить в переменные `key` и `value`:

```
users = {1: "Иванов Е.С.", 2: "Петров А.С."}
```

```
for key, value in users.items():
```

```
    print(f"Number: {key} User: {value} ")
```

Также существуют возможности перебора ключей и значений. Для перебора ключей используется метод словаря `keys()`, для перебора значений метод `values()`:

```
for key in users.keys():
```

```
    print(key)
```

```
for value in users.values():
```

```
    print(value)
```

Удаление элементов словаря

Для удаление элементов словаря можно использовать метод `pop()`, он имеет два возможных

варианта использования:

- `pop(key)`: удаляет элемент по ключу `key` и возвращает удаленный элемент. Если элемент с данным ключом отсутствует, то генерируется исключение `KeyError`

- `pop(key, default)`: удаляет элемент по ключу `key` и возвращает удаленный элемент. Если элемент с данным ключом отсутствует, то возвращается значение `default`

Ниже приведён пример использования метода `pop()` для удаления элементов словаря:

```
user = { "name": "Иванов Е.С.", "age": 35}

key = "name"

user_name = user.pop(key)

print(user_name) # Иванов Е.С.

user_city = users.pop("city", "Значение отсутствует")

print(user) # Значение отсутствует
```

Ч. 2 Кортежи

Цель работы: Изучение кортежей в Python.

Кортеж Python — используется для хранения последовательности упорядоченных и неизменяемых элементов.

По аналогии со списками кортежи в Python — это стандартный тип, позволяющий хранить значения в виде последовательности. Они полезны в тех случаях, когда необходимо передать данные, не позволяя изменять их.

Кортежи создают с помощью круглых скобок `()`. Для создания нужно написать следующее:

```
cake = ('c','a','k','e')
```

```
print(type(cake))
```

```
» <class 'tuple'>
```

Примечание: `type()` — это встроенная функция для проверки типа данных переданного параметра.

Кортежи могут включать однородные и разнородные значения. Но после объявления их уже нельзя будет поменять:

```
mixed_type = ('C', 0, 0, 'K', 'I', 'E')
```

```
for i in mixed_type:
```

```
    print(i, ":", type(i))
```

```
>>
```

```
C : <class 'str'>
```

```
0 : <class 'int'>
```

```
0 : <class 'int'>
```

```
K : <class 'str'>
```

```
I : <class 'str'>
```

```
E : <class 'str'>
```

```
# Попробуйте изменить 0 на «O» mixed_type[1] = "O"
```

```
TypeError Traceback (most recent call last) <ipython-input-16-dec28c299a95> in  
<module>() ----> 1 mixed_type[1] = 'O' # Попробуйте изменить 0 на «O» TypeError:  
'tuple' object does not support item assignment
```

Последняя ошибка появилась из-за попытки поменять значения внутри кортежа.

Кортежи можно создавать и вот так:

```
numbers_tuple = 1,2,3,4,5
print(type(numbers_tuple))
» <class 'tuple'>
```

Операции с кортежами

Срезы

Значение индекса первого элемента в кортеже — 0. По аналогии со списками эти значения можно использовать с квадратными скобками [] для получения доступа к кортежам:

```
numbers = (0,1,2,3,4,5)
numbers[0]
```

```
» 0
```

Можно использовать и отрицательные значения:

```
numbers[-1]
```

```
» 5
```

Индексы позволяют получать отдельные элементы, а с помощью срезов становятся доступны и подмножества. Для этого нужно использовать диапазоны индексов:

[Начальный индекст (включен):Конечный индекс (исключен):Частота]

Частота в данном случае является опциональным параметром, а его значение по умолчанию равно 1.

```
# Элемент с индексом 4 исключен
```

```
numbers[1:4]
```

```
» (1, 2, 3)
```

```
# Это возвращает все элементы в кортеже
```

```
numbers[:]
```

» (0, 1, 2, 3, 4, 5)

Частота = 2

numbers[::2]

» (0, 2, 4)

Совет: значение частоты может быть и негативным, чтобы развернуть кортеж.

numbers[::-1]

» (5, 4, 3, 2, 1, 0)

Объединение кортежей

Можно объединять кортежи для создания нового объекта. Операция объединения выполняет конкатенацию двух кортежей.

x = (1,2,3,4)

y = (5,6,7,8)

Объединение двух кортежей для формирования нового кортежа

z = x + y

print(z)

» (1, 2, 3, 4, 5, 6, 7, 8)

y = [5,6,7,8]

z = x + y

print(z)

» TypeError: can only concatenate tuple (not "list") to tuple

Разрешается объединять только определенные типы данных. Так, попытка соединить кортеж и список закончится ошибкой.

Умножение кортежей

Операция умножения приводит к тому, что кортеж повторяется несколько раз.

```
x = (1,2,3,4)
```

```
z = x*2
```

```
print(z)
```

```
» (1, 2, 3, 4, 1, 2, 3, 4)
```

Функции кортежей

В отличие от списков у кортежей нет методов, таких

как `append()`, `remove()`, `extend()`, `insert()` или `pop()` опять-таки из-за их неизменяемости.

Но есть другие:

`count()` и `len()`

`count()` возвращает количество повторений элемента в кортеже.

```
a = [1,2,3,4,5,5]
```

```
a.count(5)
```

```
» 2
```

`len()` — длину кортежа:

```
a = (1,2,3,4,5)
```

```
print(len(a))
```

```
» 5
```

`tuple()`

Функция `tuple()` используется для конвертации данных в кортеж. Например, так можно превратить список в кортеж.


```
a_list = [1,2,3,4,5]
b_tuple = tuple(a_list)
print(type(b_tuple))
```

» <class 'tuple'>

min() и max()

Функция `max()` возвращает самый большой элемент последовательности, а `min()` — самый маленький:

```
print(max(a))
print(min(a))
```

Эти функции можно использовать и для кортежей со строками.

Строка «Apple» автоматически преобразуется в последовательность символов.

```
a = ('Apple')
print(max(a))
```

» p

sum()

С помощью этой функции можно вернуть сумму элементов в кортеже. Работает только с числовыми значениями.

```
sum(a)
```

» 28

sorted()

Чтобы получить кортеж с отсортированными элементами, используйте `sorted()` как в следующем примере:

```
a = (6,7,4,2,1,5,3)
sorted(a)
```

» [1, 2, 3, 4, 5, 6, 7]

Но важно отметить, что возвращаемый тип — список, а не кортеж. При этом последовательность в оригинальном объекте неизменна, а сам он остается кортежем.

Присваивание несколько кортежей

Кортежи можно использовать для присваивания нескольких значений одновременно. Вот так:

```
a = (1,2,3)
(one,two,three) = a
print(one)
```

» 1

`a` — это кортеж из трех элементов и `(one, two, three)` — кортеж трех переменных. Присваивание `(one, two, three)` кортежу `a` присваивает каждое значение `a` каждой переменной: `one`, `two` и `three` по очереди. Это удобно, если нужно присвоить определенному количеству переменных значений в кортеже.

Пример задачи User

1. Пользователь вводит имя, фамилия, возраст. Создайте словарь `user` и запишите данные пользователя в него:

Решение:

```
1  firstname = input('Enter your firstname: ')
2  lastname = input('Enter your lastname: ')
3  age = input('Enter your age: ')
4  user = dict(firstname=firstname, lastname=lastname, age=age)
5  print(user)
6
```

2. Создайте функцию которая принимает объект в виде словаря и записывает в `list`.

Должен получиться list с названием users содержащий в себе последовательность словарей типа user

```
1  firstname = input('Enter your firstname: ')
2  lastname = input('Enter your lastname: ')
3  age = input('Enter your age: ')
4  user = dict(firstname=firstname, lastname=lastname, age=age)
5  print(user)
6
7  users = []
8
9
10 1 usage
11 def add_user(user_dict: dict):
12     users.append(user_dict)
13
14 add_user(user)
15
16 print(f"{users=}")
```

Варианты заданий

Вариант Задача

1

1

См пример задачи User

1. Создайте функцию которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст. Выведите полученные данные в консоль

2. Создайте функцию которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст, сделайте возврат из функции введенных данных в виде словаря.

Создайте словарь user и запишите данные пользователя в него. Выведите полученные данные в консоль

2

См пример задачи User

1. Создайте функцию которая вызывается для ввода данных в консоли. Пользователь вводит

имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию которая принимает объект в виде словаря и записывает в list.

Должен получиться list с названием users содержащий в себе последовательность словарей типа user.

3

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 5 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. Вывести получившийся список словарей в консоли.

4

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 5 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. В результирующем list который содержит 5 словарей, сделайте поиск по ключу "возраст" и выведите запись из list с самым минимальным возрастом.

5

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 6 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. В результирующем list который содержит 6 словарей, сделайте поиск по ключу "возраст" и выведите запись из list с самым большим возрастом.

6

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 6 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. В результирующем list который содержит 6 словарей, сделайте поиск по ключам "фамилия" и "имя" найдите совпадение по одному из полей с другими словарями хранящимися в list:

- есть совпадение: вывести совпадающие записи

- нет совпадений: вывести текст совпадений не найдено

Пример задачи "Фрукты"

Описание задачи с примером:

Пользователь вводит число K - количество фруктов. Затем он вводит K фруктов в формате: название фрукта и его количество. Добавьте все фрукты в словарь, где название фрукта - это ключ, а количество - значение. Например:

```
# Ввод:
>> 3 # Количество фруктов
>> Яблоко
>> 3
>> Апельсин
>> 3
>> Мандарин
>> 10
# Вывод:
>> {'Яблоко': 3, 'Апельсин': 3, 'Мандарин': 10}
```

Вариант Задача

7 См пример задачи “Фрукты”

1. Создайте функцию `input_fruit` которая вызывается для ввода данных в консоли. Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)
2. Создайте бесконечный цикл `while` который вызывает функцию `input_fruit` до тех пор пока кол-во полученных яблок не станет больше или равно 10

8 См пример задачи “Фрукты”

1. Создайте функцию `input_fruit` которая вызывается для ввода данных в консоли. Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)
2. Создайте бесконечный цикл `while` который вызывает функцию `input_fruit` до тех пор пока кол-во полученных яблок и апельсин в сумме не станет больше или равно 20

9 См пример задачи “Фрукты”

1. Создайте функцию `input_fruit` которая вызывается для ввода данных в консоли. Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)
2. Создайте бесконечный цикл `while` который вызывает функцию `input_fruit` прервать цикл если будет введен фрукт “Фейхоа”
3. Выведите результирующий список в виде словаря в консоль, не добавлять в результирующий список “Фейхоа”.

10 См пример задачи “Фрукты”

1. Создайте функцию input_fruit которая вызывается для ввода данных в консоли. Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте бесконечный цикл while который вызывает функцию input_fruit в зависимости от наименования фрукта, помещать данные в разные переменные sklad_1 (Абрикос, Авокадо, Алыча, Апельсин, Арбуз)
sklad_2 (Гранат, Грейпфрут, Груша, Дыня, Инжир, Кешью)

Прервать цикл когда на одном из складов будет 50 фруктов

3. Выведите результирующий список в виде словаря в консоль, Каждый склад ввести отдельным словарем.

11 См пример задачи “Фрукты”

На складе имеются фрукты в кол-ве штук:

sklad_1 (Абрикос=50, Авокадо=50, Алыча=50, Апельсин=50, Арбуз=50)

sklad_2 (Гранат=20, Грейпфрут=4, Груша=10, Дыня=10, Инжир=20, Кешью=100)

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте бесконечный цикл while который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов, пользователь указывает название фрукта и кол-во, в цикле проверяется сколько осталось на складе.. Сообщать пользователю сколько осталось данного фрукта, если пользователь запросил больше чем было на складе Остановить приём заявок когда будет больше 4 наименования в заказе

3. Выведите результирующий список в виде словаря в консоль, Каждый склад ввести отдельным словарем. Вывести исходное кол-во на складе. Вывести остаток на складе. Вывести заказ.

Таблица Склад фруктов

Склад №	Наименование	Вес в кг,	Цена за 1 кг
1	Абрикос	30	120
1	Авокадо	19	90
1	Алыча	32	67
1	Апельсин	67	129
1	Арбуз	129	19
2	Гранат	36	56
2	Грейпфрут	23	78
2	Груша	57	20
2	Дыня	48	45
2	Инжир	12	54
2	Кешью	31	73

Вариант Задача

12 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

Магазину необходимо оформить заявку на каждый из складов.

В магазине имеется некоторый товар:

1. Апельсин, в наличии 12 кг нужно 20
2. Груша, в наличии 3 кг нужно 10
3. Арбуз, в наличии 23 кг нужно 35
4. Авокадо, в наличии 2 кг нужно 8
5. Дыня, в наличии 0 кг нужно 14

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию `input_fruit` Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию `input_fruit` и оформляет заявку на необходимое кол-во фруктов, пользователь указывает название фрукта и кол-во, в цикле проверяется сколько осталось на складе. А так же проверяется что Одна заявка не может быть больше 30кг весом. Если больше оформить как дополнительную заявку

3. Выведите результирующий список в виде словаря в консоль, Вывести заказ с указанием веса каждой заявки и списком необходимых фруктов.

13 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

В онлайн магазин поступил заказ, на банкет необходимо доставить заказ:

1. Апельсины, 4 кг
2. Груша, 3 кг
3. Арбуз, 23 кг
4. Авокадо, 2 кг
5. Дыня, 9 кг

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию `input_fruit` Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию `input_fruit` и оформляет заявку на необходимое кол-во фруктов.
Расчитайте необходимое кол-во доставщиков если каждый может взять не более 9кг

3. Выведите результирующий список в виде словаря в консоль: Вывести для каждого доставщика заказ-наряд с указанием веса каждой заявки и товарами которые он доставляет.
Вывести остатки на складе(Название и вес)

14 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

В онлайн магазин поступил заказ, на банкет необходимо доставить заказ:

1. Апельсины, 4 кг
2. Груша, 3 кг
3. Арбуз, 23 кг
4. Авокадо, 2 кг
5. Дыня, 9 кг

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов.

Расчитайте цену заказа:

Ночной тариф, все товары *2 кроме Авокадо и груши

3. Выведите результирующий список в виде словаря в консоль: Вывести с указанием наименования, веса и ценой заявки.

Вывести остатки на складе(Название и вес)

15

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли.

Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 5 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. Вывести получившийся список словарей в консоли.

16 См пример задачи "Фрукты"

1. Создайте функцию input_fruit которая вызывается для ввода данных в консоли.

Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте бесконечный цикл while который вызывает функцию input_fruit до тех пор пока

кол-во полученных яблок не станет больше или равно 10

- 17 3. Выведите результирующий список в виде словаря в консоль
См пример задачи “Фрукты”

1. Создайте функцию `input_fruit` которая вызывается для ввода данных в консоли. Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте бесконечный цикл `while` который вызывает функцию `input_fruit` в зависимости от наименования фрукта, помещать данные в разные переменные `sklad_1` (Абрикос, Авокадо, Алыча, Апельсин, Арбуз)
`sklad_2` (Гранат, Грейпфрут, Груша, Дыня, Инжир, Кешью)

Прервать цикл когда на одном из складов будет 50 фруктов

- 18 3. Выведите результирующий список в виде словаря в консоль, Каждый склад ввести отдельным словарем.
см. таблицу Склад фруктов

На складе имеются запасы фруктов.

В онлайн магазин поступил заказ, на банкет необходимо доставить заказ:

1. Апельсины, 4 кг
2. Груша, 3 кг
3. Арбуз, 23 кг
4. Авокадо, 2 кг
5. Дыня, 9 кг

Все данные из таблицы “Склад фруктов” должны храниться в вашем коде в виде словаря

1. Создайте функцию `input_fruit` Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию `input_fruit` и оформляет заявку на необходимое кол-во фруктов.
Расчитайте необходимое кол-во доставщиков если каждый может взять не более 9кг

3. Выведите результирующий список в виде словаря в консоль: Вывести для каждого доставщика заказ-наряд с указанием веса каждой заявки и товарами которые он доставляет. Вывести остатки на складе(Название и вес)

- 19 См пример задачи User

1. Создайте функцию которая вызывается для ввода данных в консоли. Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию которая принимает объект в виде словаря и записывает в `list`.

Должен получиться list с названием users содержащий в себе последовательность словарей типа user.

20

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли.

Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 6 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. В результирующем list который содержит 6 словарей, сделайте поиск по ключам "фамилия" и "имя" найдите совпадение по одному из полей с другими словарями хранящимися в list:

- есть совпадение: вывести совпадающие записи

- нет совпадений: вывести текст совпадений не найдено

21 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

Магазину необходимо оформить заявку на каждый из складов.

В магазине имеется некоторый товар:

1. Апельсин, в наличии 12 кг нужно 20
2. Груша, в наличии 3 кг нужно 10
3. Арбуз, в наличии 23 кг нужно 35
4. Авокадо, в наличии 2 кг нужно 8
5. Дыня, в наличии 0 кг нужно 14

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов, пользователь указывает название фрукта и кол-во, в цикле проверяется сколько осталось на складе. А так же

проверяется что Одна заявка не может быть больше 30кг весом. Если больше оформить как дополнительную заявку

22

3. Выведите результирующий список в виде словаря в консоль, Вывести заказ с указанием веса каждой заявки и списком необходимых фруктов.

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли.

Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 5 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. В результирующем list который содержит 5 словарей, сделайте поиск по ключу "возраст" и выведите запись из list с самым минимальным возрастом.

23

см. таблицу Склад фруктов

На складе имеются запасы фруктов.

Магазину необходимо оформить заявку на каждый из складов.

В магазине имеется некоторый товар:

1. Апельсин, в наличии 12 кг нужно 20
2. Груша, в наличии 3 кг нужно 10
3. Арбуз, в наличии 23 кг нужно 35
4. Авокадо, в наличии 2 кг нужно 8
5. Дыня, в наличии 0 кг нужно 14

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов, пользователь указывает название фрукта и кол-во, в цикле проверяется сколько осталось на складе. А так же проверяется что Одна заявка не может быть больше 30кг весом. Если больше оформить как дополнительную заявку

3. Выведите результирующий список в виде словаря в консоль, Вывести заказ с указанием веса каждой заявки и списком необходимых фруктов.

24 См пример задачи “Фрукты”

1. Создайте функцию `input_fruit` которая вызывается для ввода данных в консоли. Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)
2. Создайте бесконечный цикл `while` который вызывает функцию `input_fruit` до тех пор пока кол-во полученных яблок и апельсинов в сумме не станет больше или равно 20
3. Выведите результирующий список в виде словаря в консоль

25 См пример задачи “Фрукты”

На складе имеются фрукты в кол-ве штук:

`sklad_1` (Абрикос=50, Авокадо=50, Алыча=50, Апельсин=50, Арбуз=50)

`sklad_2` (Гранат=20, Грейпфрут=4, Груша=10, Дыня=10, Инжир=20, Кешью=100)

1. Создайте функцию `input_fruit` Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)
2. Создайте бесконечный цикл `while` который вызывает функцию `input_fruit` и оформляет заявку на необходимое кол-во фруктов, пользователь указывает название фрукта и кол-во, в цикле проверяется сколько осталось на складе.. Сообщать пользователю сколько осталось данного фрукта, если пользователь запросил больше чем было на складе Остановить приём заявок когда будет больше 4 наименования в заказе
3. Выведите результирующий список в виде словаря в консоль, Каждый склад ввести отдельным словарем. Вывести исходное кол-во на складе. Вывести остаток на складе. Вывести заказ.

26 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

В онлайн магазин поступил заказ, на банкет необходимо доставить заказ:

1. Апельсины, 5 кг
2. Груша, 8 кг
3. Арбуз, 13 кг
4. Авокадо, 2 кг
5. Дыня, 9 кг

Все данные из таблицы “Склад фруктов” должны храниться в вашем коде в виде словаря

1. Создайте функцию `input_fruit` Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)
2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию `input_fruit` и оформляет заявку на необходимое кол-во фруктов. Расчитайте цену заказа:
Ночной тариф, все товары *2 кроме Авокадо и груши

3. Выведите результирующий список в виде словаря в консоль: Вывести с указанием наименования, веса и ценой заявки.

Вывести остатки на складе(Название и вес)

27 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

В онлайн магазин поступил заказ, на банкет необходимо доставить заказ:

1. Апельсины, 7 кг
2. Груша, 7 кг
3. Арбуз, 32 кг
4. Авокадо, 6 кг
5. Дыня, 10 кг

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов.
Расчитайте необходимое кол-во доставщиков если каждый может взять не более 8кг

3. Выведите результирующий список в виде словаря в консоль: Вывести для каждого доставщика заказ-наряд с указанием веса каждой заявки и товарами которые он доставляет.
Вывести остатки на складе(Название и вес)

28 см. таблицу Склад фруктов

На складе имеются запасы фруктов.

В онлайн магазин поступил заказ, на банкет необходимо доставить заказ:

1. Апельсины, 4 кг
2. Груша, 3 кг
3. Арбуз, 23 кг
4. Авокадо, 2 кг
5. Дыня, 9 кг

Все данные из таблицы "Склад фруктов" должны храниться в вашем коде в виде словаря

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте цикл со счетчиком (кол-во позиций в заявке) который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов.
Расчитайте цену заказа:
Ночной тариф, все товары *2 кроме Авокадо и груши

3. Выведите результирующий список в виде словаря в консоль: Вывести с указанием

наименования, веса и ценой заявки.
Вывести остатки на складе(Название и вес)

29

См пример задачи User

1. Создайте функцию input_data которая вызывается для ввода данных в консоли.

Пользователь вводит имя, фамилия, возраст, сделайте возврат данных из функции в виде словаря.

2. Создайте функцию add_user которая принимает объект в виде словаря и записывает в list.

3. Создайте цикл со счетчиком в котором функция input_data будет вызываться 6 раз, данные полученные из функции Input_data должны быть записаны с помощью функции add_user в list.

4. В результирующем list который содержит 6 словарей, сделайте поиск по ключу

"возраст" и выведите запись из list с самым большим возрастом.

30 См пример задачи "Фрукты"

На складе имеются фрукты в кол-ве штук:

sklad_1 (Абрикос=50, Авокадо=50, Алыча=50, Апельсин=50, Арбуз=50)

sklad_2 (Гранат=20, Грейпфрут=4, Груша=10, Дыня=10, Инжир=20, Кешью=100)

1. Создайте функцию input_fruit Пользователь вводит НАЗВАНИЕ_ФРУКТА, КОЛ-ВО_ФРУКТОВ, сделайте возврат данных из функции в виде кортежа (название, кол-во)

2. Создайте бесконечный цикл while который вызывает функцию input_fruit и оформляет заявку на необходимое кол-во фруктов, пользователь указывает название фрукта и кол-во, в цикле проверяется сколько осталось на складе.. Сообщать пользователю сколько осталось данного фрукта, если пользователь запросил больше чем было на складе Остановить приём заявок когда будет больше 4 наименования в заказе

3. Выведите результирующий список в виде словаря в консоль, Каждый склад ввести отдельным словарем. Вывести исходное кол-во на складе. Вывести остаток на складе. Вывести заказ.