

Разбор домашнего задания

Роман Булгаков

Спикер курса

Skillbox

Задача «Палиндром»

Входные данные:

- строка

Выходные данные:

сообщение:

- «Можно сделать палиндромом»

либо

- «Нельзя сделать палиндромом»

Строка: aab

Можно сделать палиндромом

Строка: aabc

Нельзя сделать палиндромом

Разбор задачи

Симметричная последовательность = 1 2 3 **2 1** — палиндром.

Пример_1:

aab

Перестановка: **aba** — палиндром → “Можно сделать палиндромом”

Алгоритм решения:

1. Сгенерировать все возможные перестановки строки
2. Проверить, нет ли среди них палиндромов

Крайне неэффективно

Пример_2:

Очень много перестановок

aaabacd.....ab

Разбор задачи

Пример_1:

abc не палиндром

'a': 1

'b': 1

'c': 1

abcba — палиндром

'a': 2 — чётное

'b': 2 — чётное

'c': 1 — нечётное

Пример_2:

aaabb не палиндром

'a': 3

'b': 2

aaabbaaa — палиндром

'a': 6 — чётное

'b': 2 — чётное

Ввод строки:

aabbc

Перестановка:

abcba — палиндром

Алгоритм решения:

1. Посчитать кол-во каждого символа

2. Если нечетных значений ≤ 1 . то

“Можно сделать палиндромом”

иначе

“Нельзя сделать палиндромом”

Кортежи

Роман Булгаков

Спикер курса

Skillbox

Особенности кортежей

- 0. Безопасность хранимых данных
- 1. Занимают меньше памяти
- 2. Работают быстрее

```
origin_list = [3, 1, 4, 1, 5]  
  
origin_tuple = (3, 1, 4, 1, 5)
```

Итоги урока

- `nums = (3, 1, 4, 1, 5)`
`nums = tuple(nums_list)`
- `nums[2] = 5` — **нельзя**
`num = nums[2]`
`slice = nums[1:]`
- Меньше памяти, быстрее работа
- `def get_user():`
...
 `return name, surname, age`

`user = get_user()` # ('Bob', 'Ivanov', 25)

Функция `enumerate`.

Перебор нескольких значений

Роман Булгаков

Спикер курса

Skillbox

Чистый код с enumerate

```
scores = [54, 67, 48, 99, 27]
for i_player in range(len(scores)):
    print(i_player, scores[i_player])
```

2 функции: `range` и `len`

`scores[i_player]` — взятие элемента,
используя обращение по индексу

Старый код

```
scores = [54, 67, 48, 99, 27]
for i_player, i_score in enumerate(scores):
    print(i_player, i_score)
```

1 функция: `enumerate`

Новый код (pythonic way)

Итоги урока

- `enumerate(scores)`
- `for index, value in enumerate(nums):`
 `print(index, value)`
- `for index, value in enumerate(nums):`
 `nums[index] += 10`
 `print(index, value)`
- `for index, sym in enumerate('abc'):`
 `print(index, sym)`

Перебор ключей и значений в словаре. Метод `items`

Роман Булгаков

Спикер курса

Skillbox

Задача «Товары»

Входные данные:

- 2 словаря: коды и товары

Выходные данные:

- сумма каждого товара на складе

```
goods = {
    'Лампа': '12345',
    'Стол': '23456',
    'Диван': '34567',
    'Стул': '45678',
}

store = {
    '12345': [
        {'quantity': 27, 'price': 42},
    ],
    '23456': [
        {'quantity': 22, 'price': 510},
        {'quantity': 32, 'price': 520},
    ],
    '34567': [
        {'quantity': 2, 'price': 1200},
        {'quantity': 1, 'price': 1150},
    ],
    '45678': [
        {'quantity': 50, 'price': 100},
        {'quantity': 12, 'price': 95},
        {'quantity': 43, 'price': 97},
    ],
}
```

Составные ключи

Роман Булгаков

Спикер курса

Skillbox

Ключи в базах данных

```
phonebook_dict = {  
    ('Петров', 'Ваня'): 88006663636,  
    ('Егоров', 'Ваня'): 88003333333,  
    ('Ульянов', 'Петя'): 88005553535,  
    ('Сидорова', 'Лена'): 88007773737  
}
```

Решение с помощью кортежей

```
phonebook_dict_2 = {  
    1: {  
        'surname': 'Петров',  
        'name': 'Ваня',  
        'phone': 88006663636,  
    },  
    2: {  
        'surname': 'Егоров',  
        'name': 'Ваня',  
        'phone': 88003333333,  
    },  
    ...  
}
```

Решение с помощью ID и словарей

('Сидорова', 'Лена') — составной ключ.

Функция ZIP

Роман Булгаков

Спикер курса

Skillbox

Итоги модуля

- `nums = (3, 1, 4, 1, 5)`
`nums = tuple(nums_list)`
- `user = get_user() # ('Bob', 'Ivanov', 25)`
`name, surname, age = user`
- `for index, value in enumerate(nums):`
- `for key, value in new_dict.items():`
- `phonebook_dict = {`
 `('Петров', 'Ваня'): 88006663636`
 `}`
- `people = dict(zip(names, ages))`

