

---

---

# “Общи” приказки

7 ноември 2024

---

---

# За какво ще си говорим днес

- Домашните
- Въпросите, които сте задали (*за съжаление нямаме кой знае колко*)
- Темите, които искате да обсъдим
- Но най-вече...
- **Live coding\***
  - Реално в контекста на истински задачи ще си говорим за ООП, декоратори, ламбди и прочие
  - Слайдовете днес са кът
  - Който присъства - ще е видял, който не - кофти
  - Хубаво де, кода пак ще го качим, просто сам по себе си не е супер ценен

# Речници

Какъв е проблемът?

```
INTERVALS = {0: "unison", 1: "minor 2nd", 2: "major 2nd", 3: "minor 3rd",  
             4: "major 3rd", 5: "perfect 4th", 6: "diminished 5th", ...}
```

- Същото е като:

```
INTERVALS = ["unison", "minor 2nd", "major 2nd", "minor 3rd", "major 3rd",  
            "perfect 4th", "diminished 5th", ...]
```

- Така де, не е баш същото, но функционално върши същата работа и го използваме по същия начин

# Концептуално

Какъв е проблемът?

```
def __eq__(self, other):  
    if isinstance(other, Tone):  
        return self.name == other.name  
    return False
```

- Trick question, няма проблем
- На много от вас им върнах обратна връзка за това, но се оказа, че всъщност лъжа!

```
print(5 == 'пет')  
# False
```

- Сори 😬

# Проверки

Какъв е проблемът?

```
def __sub__(self, other):  
    ...  
    if len(new_tones) < 2:  
        raise TypeError("Cannot have a chord made of only 1 unique tone")
```

- Че в повечето случаи го имате в `__init__`

```
def __init__(self, root_tone, *tones):  
    ...  
    if len(unique_tones) < 2:  
        raise TypeError("Cannot have a chord made of only 1 unique tone")
```

# Ффффффффф

Какъв е проблемът?

```
def __str__(self):  
    return f"{self.name}"
```

- Че е излишно да ползвате f-string ако няма да форматирате и е само една стойност

```
def __str__(self):  
    return self.name
```

# Consent

Какъв е проблемът?

```
if tone.name == root.name:  
    ...  
  
return '-'.join([tone.name for tone in self.tones])
```

- Че във външни за класа обекти бъркате по чужди атрибути
- Кое не е окей, освен ако не е по взаимно съгласие
- `__eq__`, `__str__` и прочие ви позволяват да енкапсулирате вътрешната имплементация на обектите и външният свят да разчита просто на операции `==`, `str` и т.н.

```
if tone == root:  
    ...  
  
return '-'.join(map(str, self.tones))
```

# По темата с контролните

- Въпроси
- От този вид:

Кое от следните принтира "Hello World" в Python?

a. `println("Hello World")`

b. `echo "Hello World"`

c. `fprintf("Hello World")`

d. `print("Hello World")`

- Уви, не е толкова шаренко
- 25 на брой
- Само затворени отговори (*от a до най-много k*)
- Теория + примери с код (*както в началото на почти всяка лекция*)
- Поне 8 варианта
- Въпроси?



# Малка вметка преди да започнем да пишем

## Types of Headaches

**Migraine**



**Hypertension**



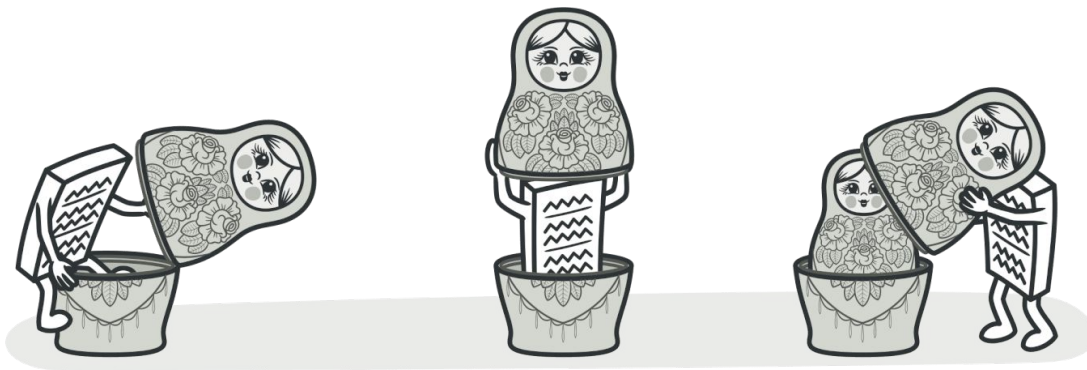
**Stress**



**Декоратори**



# Декораторите като design pattern



- Позволяват ни да надградим функционалността на един обект, "опаковайки" го по някакъв начин
- В контекста на питонските декоратори - надграждаме функционалността на функция с допълнително поведение
- Това, вярвам, ви е ясно
- Объркващото идва когато започнем да правим `@fun1("arg1")("arg2")("arg3")`

# Стига толкова слайдове, да пишем!

- [Задачи за Г. Кунчев](#)
- [Задачи за В. Бечев](#)
- Задачи за Йоан

**Въпроси?**