

Задание для выполнения в классе: создайте для файла av1fpt1.TextGrid дополнительный уровень "ideal x-sampa", где продублируйте транскрипцию с уровня ideal в системе X-SAMPA.

```
!pip install tgt praat-parselmouth phonecodes
```

```
!wget https://pkholyavin.github.io/mastersprogramming/av1fpt1.TextGrid
```

```
import tgt
import parselmouth
from phonecodes import phonecodes

tg = parselmouth.read("av1fpt1.TextGrid")
parselmouth.praat.call(tg, "Convert to Unicode")
tgt_grid_uni = tg.to_tgt()
ideal_tier = tgt_grid_uni.get_tier_by_name("ideal")
new_tier = tgt.IntervalTier(name="ideal x-sampa")
tgt_grid_uni.add_tier(new_tier)
for interval in ideal_tier:
    # заменим обозначение мягкости на символ МФА, дужку заменим на нижнее подчёркн
    text = phonecodes.ipa2xsampa(interval.text.replace("'", "j")).replace("\u03
    new_tier.add_interval(tgt.Interval(interval.start_time, interval.end_time,

print(" ".join([i.text for i in tgt_grid_uni.get_tier_by_name("ideal x-sampa")])

tgt.io.write_to_file(tgt_grid_uni, "av1fpt1_xsampa.TextGrid", "long")
```

Задание для выполнения в классе: создайте для файла av1fpt1.TextGrid дополнительный уровень "vc", где интервалы будут обозначать стечения гласных (V) и согласных (C).

```
!pip install panphon==0.5
```

```
import panphon._panphon as panphon
ft = panphon.FeatureTable()

def vc_label(feats):
    if feats is None:
        return ""
    if ("+", "cons") in feats or ("-", "syl") in feats:
        return "c"
    return "v"
```

```

vc_tier = tgt.IntervalTier(name="vc")
tgt_grid_uni.add_tier(vc_tier)
acoustic_tier = tgt_grid_uni.get_tier_by_name("acoustic")
for interval in acoustic_tier:
    segment = interval.text.replace("'", "j") # обозначим мягкость согласно МФА
    segment = segment.replace("ε", "ε") # исправим символ на правильный
    segment = segment.replace("-", "\u0325") # обозначим глухость кружочком снизу
    label = vc_label(ft.fts(segment))
    if not vc_tier.intervals or vc_tier[-1].text != label:
        vc_tier.add_interval(tgt.Interval(interval.start_time, interval.end_time, label))
    else:
        vc_tier[-1].end_time = interval.end_time

tgt.io.write_to_file(tgt_grid_uni, "av1fpt1_new.TextGrid", "long")

```

Того же самого можно добиться с помощью метода `.get_copy_with_same_intervals_merged`, который склеивает соседние интервалы с одинаковыми метками в один. Также, вместо того, чтобы создавать пустой уровень и заполнять его интервалами, можно создать копию старого и изменить в ней названия.

```

from copy import deepcopy

acoustic_tier = tgt_grid_uni.get_tier_by_name("acoustic")
vc_tier = deepcopy(acoustic_tier)
vc_tier.name = "vc"
for interval in vc_tier:
    interval.text = interval.text.replace("'", "j")
    interval.text = interval.text.replace("ε", "ε")
    interval.text = interval.text.replace("-", "\u0325")
    interval.text = vc_label(ft.fts(interval.text))
tgt_grid_uni.add_tier(vc_tier.get_copy_with_same_intervals_merged())

```

Задание для выполнения в классе: переберите все интервалы на уровне "syll" и создайте файл `.TextGrid` на его основе, предварительно преобразовав транскрипцию из X-SAMPA в IPA.

```

from bs4 import BeautifulSoup
with open("ab_pol_eng_f_free_c1.xml") as f:
    soup = BeautifulSoup(f.read(), features="xml")

data = []

layers = soup.find_all("layer")
for lr in layers:
    metadata = {}

```

```
for name in lr.find_all("name"):
    val = name.find_next_sibling("val")
    metadata["".join(name.contents)] = "".join(val.contents)
if metadata.get("layer-type") != "syll":
    continue
events = lr.find_all("event")
for ev in events:
    data.append([float(ev.attrs["start"]), float(ev.attrs["end"]), "".join(ev.contents)])

grid = tgt.TextGrid()
tier = tgt.IntervalTier()
grid.add_tier(tier)
for el in data:
    el[-1] = phonecodes.xsampa2ipa(el[-1])
    tier.add_interval(tgt.Interval(*el))

tgt.io.write_to_file(grid, "../leap.TextGrid", "long")
```