[CS 11] Prac 11d – MTRCB

Problem Statement

MTRCB has a lot of movies to review this year!

Given the raw movie list, please organize the data for MTRCB by classifying them by *genre*, and then for each genre, sorting the movies by year, and then by name.

Task Details

In this problem, you will be provided a module named `oj` whose contents are exactly as follows:

```
from dataclasses import dataclass
from enum import Enum, auto

class Genre(Enum):
    HORROR = auto()
    SCIFI = auto()
    COMEDY = auto()
    ACTION = auto()
    THRILLER = auto()
    DRAMA = auto()
    ROMANCE = auto()
    FANTASY = auto()

@dataclass(frozen=True)
class Movie:
    name: str
    year: int
    genre: Genre
```

Import them using

```
from oj import Genre, Movie
```

Your task is to implement a function called `classify_movies`. The function takes a single argument, a `tuple` or `list` of n `Movie`s describing the movies.

The function must return a `dict[Genre, list[Movie]]` denoting the organized data. The keys must be all the distinct genres listed in the `Genre` Enum, and for each genre, the corresponding value is the sorted list of movies.

Restrictions

(See 11a for more restrictions)

For this problem in particular:

- The following imports are allowed: `Genre` and `Movie` from `oj`.
- The source code limit is 3000.

Example Calls

Example 1 Function Call

```
classify_movies((
    Movie(
        name='Happy Death Day',
        year=2017,
        genre=Genre.HORROR,
    ),
    Movie(
        name="Rosemary's Baby",
        year=1968,
        genre=Genre.HORROR,
    ),
    Movie(
        name='Morbius',
        year=2022,
        genre=Genre.HORROR,
    ),
    Movie(
        name='John Wick',
        year=2014,
        genre=Genre.ACTION,
    ),
    Movie(
        name='Wicked',
        year=2024,
        genre=Genre.FANTASY,
    ),
    Movie(
        name='M3GAN',
        year=2022,
        genre=Genre.HORROR,
    ),
))
```

Example 1 Return Value

```
{
    Genre.HORROR: [
        Movie(
            name="Rosemary's Baby",
            year=1968,
            genre=Genre.HORROR,
        ),
        Movie(
            name='Happy Death Day',
            year=2017,
            genre=Genre.HORROR,
        ),
        Movie(
            name='M3GAN',
            year=2022,
            genre=Genre.HORROR,
        ),
        Movie(
            name='Morbius',
            year=2022,
            genre=Genre.HORROR,
        ),
    ],
    Genre.SCIFI: [],
    Genre.COMEDY: [],
    Genre.ACTION: [
        Movie(
            name='John Wick',
            year=2014,
            genre=Genre.ACTION,
        ),
    ],
    Genre.THRILLER: [],
    Genre.DRAMA: [],
    Genre.ROMANCE: [],
    Genre.FANTASY: [
        Movie(
            name='Wicked',
            year=2024,
            genre=Genre.FANTASY,
        ),
    ],
}
```

Testing

To test your program locally, you should create a file called `oj.py` and save the code above to it. Note that this `oj.py` is **not** to be submitted! The judge has its own version of `oj.py`. The `oj.py` you create is only for your own testing.

Constraints

- The function `classify_movies` will be called at most $70{,}000$ times.
- The sum of the ns will be at most $250{,}000$.
- $0 \le n \le 250{,}000$
- Each movie name is a nonempty string of at most 15 characters.
- Each year is a positive integer at most 10^8.

Scoring

- You get 140 ❤ points if you solve all test cases where:
  - $n \le 50$
  - the sum of the ns is at most 600.
- You get 30 ❤ points if you solve all test cases where:
  - $n \le 5{,}000$
  - the sum of the ns is at most $5{,}000$.
- You get 30 ❤ points if you solve all test cases.

Submit solution [CS 11] Practice 11

Points: 200 (partial)
Time limit: 8.0s
Memory limit: 1G

Author: kvatienza (Kevin Atienza)

Problem type

Allowed languages ~~NONE~~, py3

Clarifications

Report an issue

No clarifications have been made at this time.