- **Developer**: Swathi
- **Collaborator**: Ruchitha

Design of `constructor`

- `self` is the current instance of the class
- `programmers` and `languages` are the two list parameters of the constructor, passed while creating the class instance
- instance variables `self.programmers`, `self.languages` are intialized to empty dictionary and are used as accumulators
- for `index` in range of the length of the `programmers` list, iterate through the following:
    - assign the element at `index` position in the `languages` to the `self.programmers` dictionary for the `key programmers[index]`
    - for `language` in `languages[index]` sublist of `languages`, do the following:
        - if `language` is not already present as `key` in `self.languages`:
            - intialize `self.languages` with key `language` to an empty list
        - concatenate the value at `programmer[index]` to the key `language` of `self.languages`

Design of `find_languages()` method

- `self` is the current instance of the class
- `programmer` is the name of the programmer of type string is passed as an argument to the method
- get the language list from `self.programmers` dictionary associated with the key `programmer`
    - `self.programmers` is populated with required data during class instance creation.
- type cast it to a list and return the resulted list

Design of `find_programmers()` method

- `self` is the current instance of the class
- `language` is a programming language of type string is passed as an argument to the method
- get the programmers list from `self.languages` dictionary associated with the key `language`
    - `self.languages` is populated with required data during class instance creation.
- type cast it to a list and return the resulted list