Lab 2: Introduction to Object-Oriented Programming Solution

3) Implementing your design

```
from typing import List


class Runner:
    """A runner for the race

    === Attributes ===
    name: the name of runner.
    email: the email of runner.
    speed_category: speed category runner is racing in

    === Sample Usage ===
    Create a runner:
    >>> runner = Runner('Gerhard', 'gerhard@gmail.com','lt30')
    >>> runner.name
    'Gerhard'
    >>> runner.email
    'gerhard@gmail.com'
    >>> runner.speed_category
    'lt30'
    """
    name: str
    email: str
    speed_category: str

    def __init__(self, name: str, email: str, speed_category: str) -> None:
        """Initialize runner

        Precondition: <name> != ''
        Precondition: <email> != ''
        Precondition: <speed_category> in ['lt20','lt30','lt40','gt40','']

        >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
        >>> runner.name
```

```
            'Gerhard'
            >>> runner.email
            'gerhard@gmail.com'
            >>> runner.speed_category
            'lt30'
            """
            self.name = name
            self.email = email
            self.speed_category = speed_category

        def edit_email(self, email: str) -> None:
            """Edits runner email information

            Precondition: <email> != ''

            >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
            >>> runner.email
            'gerhard@gmail.com'
            >>> runner.edit_email('gerhard_2@gmail.com')
            >>> runner.email
            'gerhard_2@gmail.com'
            """

            self.email = email

        def edit_category(self, speed_category: str) -> None:
            """Edits runner speed category information

            Precondition: <speed_category> in ['lt20','lt30','lt40','gt40
    ']

            >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
            >>> runner.speed_category
            'lt30'
            >>> runner.edit_category('lt20')
            >>> runner.speed_category
            'lt20'
            """

            self.speed_category = speed_category

        def withdraw(self) -> None:
            """Withdraws runner from race

            >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
            >>> runner.speed_category
            'lt30'
            >>> runner.withdraw()
            >>> runner.speed_category
            ''
            """

            self.speed_category = ''

```

```
88
89      class Race:
90          """Race Registry
91
92          === Attributes ===
93          runners: a list of runners in race
94
95          === Sample Usage ===
96
97          Create a race registry:
98          >>> r = Race()
99          >>> r.runners
100         []
101
102         Registering runners:
103         >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
104         >>> r.register(runner_1)
105         >>> r.runners[0].name
106         'Gerhard'
107         >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt30')
108         >>> r.register(runner_2)
109         >>> r.runners[1].name
110         'Tom'
111         >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
112         >>> r.register(runner_3)
113         >>> r.runners[2].name
114         'Toni'
115
116         Updating runner in a speed category:
117         >>> runner_4 = r.get_runner('Gerhard')
118         >>> runner_4.edit_category('lt30')
119         >>> runner_4.speed_category
120         'lt30'
121
122         Get all runners in a speed category:
123         >>> runners = r.get_runners('lt30')
124         >>> runners[0].name
125         'Gerhard'
126         >>> runners[1].name
127         'Tom'
128         """
129         runners: List[Runner]
130
131         def __init__(self) -> None:
132             """Initializes race registry
133
134             >>> r = Race()
135             >>> r.runners
136             []
137             """
138             self.runners = []
139
140         def register(self, runner: Runner) -> None:
141             """Registers runner to race
```

```
142
143             >>> r = Race()
144             >>> runner = Runner('Gerhard','gerhard@gmail.com','lt30')
145             >>> r.register(runner)
146             >>> r.runners[0].name
147             'Gerhard'
148             """
149             self.runners.append(runner)
150
151         def get_runners(self, category: str) -> None:
152             """Returns list of runners in race category
153
154             Precondition: <speed_category> in ['lt20','lt30','lt40','gt40
    ']
155
156             >>> r = Race()
157             >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
158             >>> r.register(runner_1)
159             >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt20')
160             >>> r.register(runner_2)
161             >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
162             >>> r.register(runner_3)
163             >>> runners = r.get_runners('lt20')
164             >>> runners[0].name
165             'Tom'
166             >>> runners[1].name
167             'Toni'
168             >>> r.get_runners('lt30')
169             []
170             """
171             result = []
172
173             for runner in self.runners:
174                 if runner.speed_category != category:
175                     continue
176
177                 result.append(runner)
178
179             return result
180
181         def get_runner(self, name: str) -> None:
182             """Returns runner in race registry
183
184             Precondition: <name> != ''
185
186             >>> r = Race()
187             >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
188             >>> r.register(runner_1)
189             >>> fetched_runner_1 = r.get_runner('Gerhard')
190             >>> fetched_runner_1.name
191             'Gerhard'
192             >>> fetched_runner_2 = r.get_runner('Toni')
193             >>> fetched_runner_2
194             """
```

```
            for runner in self.runners:
                if runner.name != name:
                    continue

                return runner

            return None

    if __name__ == '__main__':
        import doctest
        doctest.testmod()
```