

## توضیحات تمرین اول :

طبق خواسته مسئله به منظور تشکیل یک تیم ورزشی، اپراتور اطلاعات افراد را شامل؛ کد فرد، سن، قد و وزن، در سیستم ثبت میکند به این منظور کلاس ذیل پارامترهای ورودی خواسته شده را می گیرد:

```
class Player_Selection:

    players_List=[]
    def __init__(self,personCode,age,weight,height):
        self.personCode=personCode
        self.age=age
        self.weight=weight
        self.height=height
```

برای چک کردن شرایط ثبت نام افراد مطابق خواسته مسئله توابع ذیل نوشته شد:

- برای سنین بین ۱۵ تا ۲۵ سال، وزن بین ۶۰/۰ تا ۸۰/۰ کیلوگرم
- برای سنین بین ۲۵ تا ۳۵ سال، وزن بین ۵۰/۰ تا ۷۵/۰ کیلوگرم
- برای هر دو گروه سنی قد بین ۱۷۰ تا ۱۹۰ سانتیمتر

```
def ageValidation(age):

    Player_Selection.check_Integer(age,'Age')
    age = int(age)
    if age<15 or 35<age:
        raise My_Exception_Handling('Age out of range\nThis person is not allowed to register')
    return age

def age_WeightValidation(age,weight):

    Player_Selection.check_Float(weight)
    weight=float(weight)

    if age>15 and 25>age:
        if weight>80 or 60>weight:
            raise My_Exception_Handling('Weight out of range\nThis person is not allowed to register')
    elif age>25 and 35>age:
        if weight>75 or 50>weight:
            raise My_Exception_Handling('Weight out of range\nThis person is not allowed to register')

    return age,weight

def heightvalidation(height):

    Player_Selection.check_Integer(height,'Height')
    height = int(height)

    if height<170 or height>190:
        raise My_Exception_Handling('Height out of range\nThis person is not allowed to register')

    return height
```

نمونه ای از ورودی اشتباه این سه پارامتر (رعایت نکردن شروط مسئله) و نتیجه آزمون به شکل زیر است:

```
Enter the player code (0 to Exit):1000
Enter the player age:14 سن کمتر و خارج محدوده وارد شده
Enter the player weight:60
Enter the player height:170
Error is: Age out of range
This person is not allowed to register
Enter the player code (0 to Exit):1000
Enter the player age:16 شرط ۲۵ < سن < ۱۵ رعایت شده
Enter the player weight:58.5 شرط ۸۰ < وزن < ۶۰ رعایت نشده
Enter the player height:170
Error is: Weight out of range
This person is not allowed to register
Enter the player code (0 to Exit):1000
Enter the player age:16 شرط ۲۵ < سن < ۱۵ رعایت شده
Enter the player weight:61 شرط ۸۰ < وزن < ۶۰ رعایت شده
Enter the player height:165 شرط ۱۹۰ < قد < ۱۷۰ رعایت نشده
Error is: Height out of range
This person is not allowed to register
Enter the player code (0 to Exit):1000
Enter the player age:26 شرط ۳۵ < سن < ۲۵ رعایت شده
Enter the player weight:49.5 شرط ۷۵ < وزن < ۵۰ رعایت نشده
Enter the player height:170
Error is: Weight out of range
This person is not allowed to register
Enter the player code (0 to Exit):1000
Enter the player age:26 شرط ۳۵ < سن < ۲۵ رعایت شده
Enter the player weight:51.5 شرط ۷۵ < وزن < ۵۰ رعایت شده
Enter the player height:169 شرط ۱۹۰ < قد < ۱۷۰ رعایت نشده
Error is: Height out of range
This person is not allowed to register
Enter the player code (0 to Exit):1000
Enter the player age:36 سن بزرگتر و خارج از محدوده وارد شده
Enter the player weight:60
Enter the player height:172
Error is: Age out of range
This person is not allowed to register
Enter the player code (0 to Exit):0
Program terminated.
```

همچنین از آنجا که تمام پارامترهای ورودی به عنوان رشته حساب می شوند و طبق خواسته مسئله نیاز است ورودی سن عدد صحیح، ورودی وزن عدد صحیح و اعشاری و نیز ورودی قد به عنوان عدد صحیح آزمون شود از این رو به کمک دو تابع staticmethod به نام های check\_Integer و check\_Float این موضوع با فراخوانی در توابع validation بررسی می شود:

```
@staticmethod
def check_Integer(value,valueType):
    if value.isdigit():
        return value
    else:
        if not isinstance(value,int):
            raise RuntimeError(f'{valueType} is not valid...')

@staticmethod
def check_Float(weight):
    try:
        weight=float(weight)
    except:
        if not isinstance(weight,float):
            raise RuntimeError('Weight is not valid...')
```

نمونه ای از ورودی اشتباه این سه پارامتر و نتیجه آزمون به شکل زیر است که هیچ لیستی به عنوان ثبت نامی برگردانده نشده است:

```
Enter the player code (0 to Exit):1000
Enter the player age:12.5
Enter the player weight:61
Enter the player height:180
Age is not valid...
Enter the player code (0 to Exit):1000
Enter the player age:16
Enter the player weight:string
Enter the player height:189
Weight is not valid...
Enter the player code (0 to Exit):1000
Enter the player age:16
Enter the player weight:62.5
Enter the player height:1.80
Height is not valid...
Enter the player code (0 to Exit):0
Program terminated.
Following registrations is successfully completed:
[]
```

طبق خواسته مسئله آزمون صحت ورودی پارامترها در کلاس `register_Player` به شکل ذیل نوشته شد:

```
def register_Player(personCode,age,weight,height):

    try:
        age=Player_Selection.ageValidation(age)
        age,weight=Player_Selection.age_WeightValidation(age,weight)
        height=Player_Selection.heightvalidation(height)

    Player_Selection.players_List.append((personCode,age,weight,height))

    except My_Exception_Handling as error:
        print(error)

    except RuntimeError as error:
        print(error)
```

کلاس کنترل خطاهای استثنا نیز بصورت ذیل نوشته شد که پیغام مرتبط را به شکل دلخواه نمایش دهد:

```
class My_Exception_Handling(Exception):
    def __init__(self,message):
        super().__init__(message)
        self.message=message

    def __str__(self) :
        return f'Error is:\t{self.message}'
```

در نهایت در برنامه اصلی به کمک حلقه `while` و تا زمانی که کد فرد مخالف صفر باشد از اپراتور اقدام به گرفتن ورودی کرده و با وارد شدن کد صفر در صورت عدم خطا، اسامی افراد ثبت نامی که به شکل لیست درآمده بصورت تاپل نمایش داده می شود:

```
while True:

    personCode = int(input('Enter the player code (0 to Exit):'))
    if personCode == 0:
        print("Program terminated.")
        break
    age = input('Enter the player age:')
    weight =input('Enter the player weight:')
    height = input('Enter the player height:')

    Player_Selection.register_Player(personCode, age, weight, height)

print('Following registrations is successfully completed:')
print(Player_Selection.players_List)
```

در آخر با رعایت شروط مسئله و عدم ایجاد خطا لیستی از اسامی افراد به شکل افراد ثبت نامی مطابق تصویر ذیل است:

```
Enter the player code (0 to Exit):1000
Enter the player age:16
Enter the player weight:61.5
Enter the player height:172
Enter the player code (0 to Exit):1001
Enter the player age:20
Enter the player weight:65
Enter the player height:180
Enter the player code (0 to Exit):1002
Enter the player age:27
Enter the player weight:55
Enter the player height:189
Enter the player code (0 to Exit):0
Program terminated.
Following registrations is successfully completed:
[(1000, 16, 61.5, 172), (1001, 20, 65.0, 180), (1002, 27, 55.0, 189)]
PS D:\PythonClass\Advance>
```

توضیحات تمرین دوم :

خروجی مسئله طبق خواسته تمرین 2 به شکل ذیل است:

```
List of cities by population density
-----
City          Density
city1         30
city2         50
city3         76
city4         33
city5         63
city6         10
*****
List of high-Density cities
-----
City          Density
city6         10
city6         10
city6         10
PS D:\PythonClass\Advance>
```