

ارث بری (Inheritance)

ارث بری :

ارث بری (Inheritance)، کاربر را قادر می‌سازد تا کلاسی را تعریف کند که همه عملکردها را از کلاس والد داشته باشد و در عین حال، امکان افزودن موارد بیشتر را نیز برای خود فراهم می‌آورد.

ارث بری یک ویژگی قدرتمند در برنامه‌نویسی شی‌گرا (Object Oriented Programming) محسوب می‌شود. کلاس جدید، کلاس مشتق شده یا کلاس فرزند یا subclass نامیده می‌شود و کلاسی که از آن مشتق می‌شود را کلاس والد یا superclass می‌نامند.

حالا قرار است که ما یک کلاس به عنوان شخص (Person) تعریف کنیم که فرزندان این کلاس معلم‌ها (Teacher)، دانش‌آموزان (Student) هستند. ویژگی مشترک همه این کلاس‌ها این است که همه آنها نام، نام خانوادگی، سن و کد ملی دارند و همچنین همه آنها دو دستور ساخت ایمیل و همچنین چاپ اسم و فامیل بصورت کامل را دارند. اما معلم علاوه بر این یک رتبه معلمی دارد که بر اساس آن مبلغ حقوقش مشخص می‌شود. دانش‌آموز علاوه بر صفات و متدهای کلاس Person، صفت grade و همچنین تکلیف دارد و یک متد هم برای ست کردن تکلیف برای دانش‌آموزان وجود دارد.

برای نوشتن چنین کلاسی باید متغیر نام، نام خانوادگی، سن و کد ملی و همچنین متدهای مورد نیاز را درون کلاس والد (Person) قرار دهیم و برای ویژگی خاص هر یک از کلاس‌های فرزند متغیر و یا متدی درون آنها تعریف کنیم. نکته قابل توجه درباره کلاس‌ها این است که کلاس فرزند به همه توابع کلاس والدش دسترسی داشته و همه ویژگی‌های آنرا به ارث می‌برد.

تعریف کلاس Person :

در این کلاس مطابق جزوه جلسه گذشته، صفات و متدها برای کلاس Person نوشته شده‌اند.

یک متغیر pi هم به عنوان متغیر کلاس تعریف شده است و مقدار آن را برابر با ۳.۱۴ قرار دادیم.

در ادامه کد بعد از تعریف تابع init، دو متد full_name و email_maker نوشته شده است که وظیفه اولی پرینت کردن نام و نام خانوادگی کامل فرد است و متد دوم برای شخص ایمیل مربوط به خودش را پرینت میکند.

```
class Person:
    pi = 3.14

    def __init__(self, name, family, idNumber, age):
        self.name = name
        self.family = family
        self.idNumber = idNumber
        self.age = age

    def full_name(self):
        print(self.name, self.family)

    def email_maker(self):
        print(self.name + self.family + "@rouzbeh.info")
```

تعریف کلاس Student با ارث بری از کلاس Person :

برای اینکه کلاس Student از کلاس Person ارث بری کند، فقط کافیست که داخل پرانتز جلوی اسم کلاس، نام کلاسی که می‌خواهیم از آن ارث بری کنیم را بنویسیم. [class Student(Person):]

داخل تابع `init` کلاس `Student` ، ورودی های مورد نظر نوشته شده است که همانطور که میبینید دیگر نیاز نیست برای مشترکات تعریف ها را انجام دهیم و فقط کافیه کلاس `Person` و سپس متد `init` آن را مطابق شکل زیر صدا بزنیم.

میتوانیم به جای استفاده از کلاس `Person` برای صدا زدن متد `init` از `super()` استفاده کنیم که استفاده از این دستور به این شرط است که دیگر به عنوان ورودی از `self` استفاده نکنیم :

```
super().__init__(name, family, idNumber, age)
```

در ادامه ی کد مقدار `homework` برابر `None` قرار داده شده است و همچنین `grade` تعریف شده است.

یک متد هم برای تعریف کردن تکلیف برای دانش آموز تعریف شده است.

```
class Student(Person):

    def __init__(self, name, family, idNumber, age, grade):
        Person.__init__(self, name, family, idNumber, age)
        self.homework = None
        self.grade = grade

    def set_homework(self, homework):
        self.homework = homework
```

تعریف کلاس `Teacher` با ارث بری از کلاس `Person` :

به همان شیوه بالا ، کلاس `Teacher` هم برای ارث بری از `Person` تعریف شده است.

در ادامه کد بعد از تعریف ورودی ها ، یک تابع محاسبه حقوق تعریف شده است که حقوق معلم را با توجه به رتبه آن محاسبه میکند.

```
class Teacher(Person):
    def __init__(self, name, family, idNumber, age, rank):
        Person.__init__(self, name, family, idNumber, age)
        self.rank = rank

    def set_salary(self):
        if self.rank == 1:
            print("100 M T")
        elif self.rank == 2:
            print("20 M T")
        else:
            print("1 M T")
```

حال در ادامه میتوانیم با ساختن شی (object) از هر کلاس ، از متد های کلاس ها استفاده کنیم. نکته قابل توجه این است که شی های ساخته شده از کلاس معلم و دانش آموز میتوانند از متدهای کلاس شخص یعنی ساختن ایمیل و همچنین چاپ نام و نام خانوادگی استفاده کنند و این متد ها را صدا بزنند. **چرا؟** چون این کلاس ها از کلاس `Person` ارث بری میکنند و به تمام صفت ها ، متد ها و حتی متغیر ها مانند متغیر `pi` دسترسی دارند.

همانطور که در کلاس هم توضیح داده شد ، مدل های مختلفی از ارث بری وجود دارد که همگی آنها با توجه به تعریف مسئله قابل استفاده است و نکته خاصی ندارد.

نحوه تعریف کلاس هایی که از ۱ کلاس بیشتر ارث بری میکنند به شکل زیر است (مثلا کلاس معلم اگر از دو کلاس شخص و دانش آموز ارث بری کند) :

```
class Teacher(Person, Student):
```