

# کارمندان سلطان

- مبحث مرتبط: جاوای پیشرفته

شرکت سلطان کارمندان زیادی دارد، برای همین برای سلطان مهم است که از عملکرد آنها با خبر باشد و کارمندان خود را بشناسد. وظیفه طراحی این برنامه کارمند سنج به عهده شماست.

فایل اولیه پروژه را از [این لینک](#) دانلود کنید.

برنامه شما باید به صورت زیر باشد:

یک کلاس `Employee` بنویسید که دارای فیلدهای زیر است:

- `name`
- `age`
- `rate`

یک کانستراکتور برای این کلاس بنویسید. تابع `getRate` را برای این کلاس بنویسید. تابع `toString` را برای این کلاس به صورت زیر Override کنید:

```
1 public String toString() {
2     return "name: " + name + " * age: " + age + " * rate: " + rate;
3 }
```

یک `FunctionalInterface` بنویسید به صورتی که متدهای زیر را داشته باشد:

- یک ابسترکت متد به نام `employees` دارد که پارامتری ندارد و جنس برگشتی آن یک `ArrayList` از `Employee` ها باشد.
- متد `sortByName` : این متد را به شکلی بنویسید که از ابسترکت متد `employees` استفاده کرده و لیست کارمندان را به ترتیب حروف الفبا مرتب کرده و چاپ کند.
- متد `sortByRate` : این متد همانند متد قبلی است ولی این بار لیست کارمندان را بر اساس `rate` و به صورت نزولی چاپ کنید.
- متد `firstThree` : این متد شبیه متد قبلی است با این تفاوت که فقط سه کارمند اول با بیشترین `rate` را چاپ میکند.
- متد `youngAndStrong` : این متد هم از ابسترکت متد `employees` استفاده کرده، و لیست کارمندانی را چاپ میکند که دو ویژگی `age < 25` و `rate > 8` را به صورت همزمان داشته باشند.
- متد `print` : این متد از قبل برای شما نوشته شده است:

```
1 default void print(){
2     System.out.println("-");
3     System.out.println("sort by name:");
4     System.out.println(sortByName());
5     System.out.println("-");
6     System.out.println("sort by rate:");
7     System.out.println(sortByRate());
8     System.out.println("-");
9     System.out.println("first three:");
10    System.out.println(firstThree());
11    System.out.println("-");
12    System.out.println("young and strong:");
13    System.out.println(youngAndStrong());
14    System.out.println("-");
15 }
```

آن را در فایل خود عیناً کپی کنید.

## توجه

\*برای کامل کردن تمامی قسمت های ذکر شده، باید از امکانات جدید جاوا نسخه های ۸ به بالا (مثلا عبارات `lambda`، `کلیدواژه var`، `فانکشنال اینترفیس`، `رفرنس به متد` و `stream` و `string repeat` و `predicate` و `Function` و `Optional`) استفاده کنید. در غیر این صورت نمره ای به شما تعلق نمیگیرد.\*

## مثال

اگر تابع `main` برنامه را به صورت زیر قرار دهید: (به نحوه `new` کردن آبجکت در این مثال توجه داشته باشید).

```
1 public static void main(String[] args) {
2
3     Lists myEmployees = new Lists() {
4         @Override
5         public ArrayList<Employee> employees() {
6             return new ArrayList<Employee>(
7                 Arrays.asList(
8                     new Employee("Hasan", 19, 10),
9                     new Employee("Vali", 32, 8),
10                    new Employee("Parsa", 43, 5),
11                    new Employee("Ali", 30, 9),
12                    new Employee("Bijan", 53, 6),
13                    new Employee("Ahmad", 29, 7),
14                    new Employee("Naghi", 26, 3),
15                    new Employee("Taghi", 21, 6)
16                ));
17         }
18     };
19     myEmployees.print();
20 }
```

خروجی شما باید به شکل زیر باشد:

```
-
sort by name:
name: Ahmad * age: 29 * rate: 7
name: Ali * age: 30 * rate: 9
name: Bijan * age: 53 * rate: 6
name: Hasan * age: 19 * rate: 10
name: Naghi * age: 26 * rate: 3
name: Parsa * age: 43 * rate: 5
name: Taghi * age: 21 * rate: 6
name: Vali * age: 32 * rate: 8

-
sort by rate:
name: Hasan * age: 19 * rate: 10
name: Ali * age: 30 * rate: 9
name: Vali * age: 32 * rate: 8
name: Ahmad * age: 29 * rate: 7
name: Bijan * age: 53 * rate: 6
name: Taghi * age: 21 * rate: 6
name: Parsa * age: 43 * rate: 5
name: Naghi * age: 26 * rate: 3

-
first three:
name: Hasan * age: 19 * rate: 10
name: Ali * age: 30 * rate: 9
name: Vali * age: 32 * rate: 8

-
young and strong:
name: Hasan * age: 19 * rate: 10

-
```