```
Main ✕

/Library/Java/JavaVirtualMachines/jdk1.8.0_202.jdk/Contents/Home/bin/java ...
if you want to strat the program ENTER "start":
junk
ok
cd
start
Enter information of employees and  at the end enter "exit":
DataScientist Maryam Alipour 20 9612037  100 12
ProductManager susan simani 43 4563827  2000 3
DataScientist Mohammad Malekpour 21 9613425  200 13
DataScientist Armin TaheriFard 22 9523045  300 23
SoftwareEngineer mahsa atayi 34 176534  150 5
DataScientist hassan lmozafari 27 8832016  400 25
SoftwareEngineer zahra giviani 22 9613045  100 18
ProductManager reyhane mohseni 25 9812345  150 1
SoftwareEngineer ali amiri 32 123123  100 62
exit
Mean of dataScientists salaries : 250
Mean of softwareEngineers salaries: 116
Mean of productManagers salaries: 1075

Process finished with exit code 0
```

ابتدا یک کلاس برای Person ساختیم که در آن اطلاعاتی مانند lastname , firstname

, age , و nationalID به صورت متغیرهای Private تعریف شده، به قلیم که

Encapsulation را رعایت کرده باشیم و با توابع getter ، Setter از زیر کلاس ها به آن ها

دسترسی داشته باشیم. سپس کلاس Employee را از کلاس Person ارث بردم

اسپت برای سازی، متغیرهای department ID , Salary را به صورت ID تعریف می کنم.

حال کلاس های Product Manager, Software Engineer, Data Scientist را که از کلاس

Employee ارث برده کرده اند (ریجین نرث از کلاس Person) ی سازیم و Constructor آن ها

بطوری ی سازیم که با استفاده از توابع Setter : Supechss خود اطلاعات کارمندان را set کند.

حال نوبت کلاس Calculate است. در این کلاس سه متغیر Private و ساختار

Arraylist از جنس Data Scienti, Software Engine, Product Manager, تعریف کنیم.

Compute Method و نحو محاسبات و دریافت ورودی ها از کاربر در این کلاس انجام می شود

برای شروع گرفتن اطلاعات مورد نظر، از کاربر منتظر ورود کلمه‌ی start هستیم.

```
while (true) {
order = input.nextline();
if (order.equals ("start"))
    break;
}
```

اینقدر از کاربر ورودی می‌گیریم تا کلمه‌ی start را وارد کند.

حال از کاربر می‌خواهیم اطلاعات خود را وارد کند.

اطلاعات را به صورت یک خط از String دریافت می‌کنیم و تا وقتی که "exit" را وارد نکرده

```
while (!info.equals ("exit")) {
info = input.nextline();
}
```
منتظر اطلاعات شریکی هستیم.

کار دیگری که در حلقه‌ی while انجام می‌دهیم این است که تک تک تام split کلاس String

هر info را (که در این اطلاعات مورد نظر را کاربر space وارد کرده) در آرایه‌ای از String

که تعریف کردیم بروی space موجود بین کلمات آن می‌ریزیم. حل هر خانه این آرایه‌ها

firstname, role و .... هستند. درایه‌های 3، 4، 5، 6 آرایه‌ها ما integer هایی هستند

به صورت String ذخیره شده‌اند پس با استفاده از تابع parseInt کلاس Integer آن‌ها را به

int (Primitive int) تبدیل می‌کنیم و درایه‌ی 0 آرایه را به عنوان role نگه می‌کنیم. اگر role

info وارد شده برابر با DataScientis بود یک objet از DataScientist به نام tmp می‌سازیم

که با تابع سازنده‌ی مورد نظر firstName و .... را Set می‌کند و در یک Arraylist از DataScientist

آدرسی کند و درختری در Software Engineer و یا در Product Manager بوده به همین ترتیب در Arraylist

مربوطهرای خودشان add میکند. حال خارج از حلقهی while سر حلقه برای پیمایش

اعضای هر یک از Arraylist ها مینویسیم. برای مثال در Arraylist ای از DataScientist ها

با پیمایش آن در هر بار پیمایش یک object داریم ! تابع ؟ getSalary و Salary آن را

میگیریم و در یک متغیره SumDS که تعریف کرده اضافه میکنیم. و بعد از حلقهی for

میانگین حقوق دانشمندان داده را به صورت SumDS/DS.Size() حساب میکنیم.

(زیرا برای دریافت طول یک Arraylist تابع مناسب Size است) و به همین ترتیب برای

دو نوع دیگر میانگین حقوقشان را محاسبه میکنیم . و در خود کلاس Print میکنیم.

و در کلاس Main فقط با ساختن object از کلاس Calculate و صدا زدن

تابع calcalate آن برنامه را اجرا میکنیم.