

## بقالی

در بقالی سر کوچه سیستم جدیدی قرار است پیاده سازی شود که با استفاده از آن صاحب مغازه می‌تواند اجناس موجود در انبار و قفسه‌های خود را مدیریت کند. صاحب مغازه فکر تبدیل شدن به یک فروشگاه زنجیره‌ای را در سر دارد. به او کمک کنید تا بتواند با مدیریت بهتر منابعش به هدف خود نزدیک‌تر شود.

در این تمرین شما باید برنامه‌ای را پیاده سازی کنید که بتواند از محیط کنسولی (command line) از کاربر ورودی بگیرد. هر ورودی در یک خط داده می‌شود که شامل چند بخش است. در بخش ابتدایی نام دستور می‌آید و سپس پارامترهای مربوط به آن داده می‌شود. مانند فرم زیر:

COMMAND argument1 argument2 ....

**نکته:** نام دستور و پارامترها با فاصله (Space) از یک دیگری جدا شده اند.

دستورات مورد نیاز در جدول زیر داده شده است.

Command Name	Argument 1	Argument 2	Argument 3
Define	Name	Buying Cost	Selling Cost
Delete	Name	-	-
Sell	Name	Count	-
Buy	Name	Count	-
Status	-	-	-
Financial	-	-	-
Exit	-	-	-

توضیحات مربوط به هر دستور:

**Define:**

این دستور مشخصات یک کالا را تعریف می کند. در پارامتر اول نام محصول مشخص می شود. پارامتر دوم قیمت خرید آن از کارخانه را نشان می دهد و پارامتر سوم قیمت فروش آن به خریدار است.

#### Delete:

این دستور مشخصات یک کالای تعریف شده را حذف می کند. در پارامتر اول این دستور اسم کالایی که مشخصاتش باید حذف شود داده می شود. توجه کنید که این دستور زمانی می تواند مشخصات محصول را حذف کند که هیچ نمونه ای از آن در انبار موجود نباشد. اگر نمونه ای از آن موجود بود باید در خروجی و در یک خط جداگانه یک علامت تعجب (!) چاپ کنید.

#### Sell:

این دستور به معنای فروختن یک نمونه جنس به خریدار است. در پارامتر اول اسم جنس می آید و در پارامتر بعدی تعداد آن می آید. اگر این تعداد محصول در انبار موجود نبود باید در یک خط جداگانه یک علامت تعجب نمایش داده شود. اگر این دستور اجرا شود تعداد کالاهای موجود در انبار کاهش می یابد. میزان کاهش برابر تعداد مشخص شده در پارامتر دوم است.

#### Buy:

این دستور به معنی خرید یک محصول از کارخانه است. با اجرای این دستور تعداد اجناس موجود در انبار افزایش می یابد. توجه شود که نام محصول خریداری شده در پارامتر اول می آید و تعداد خریداری شده در پارامتر دوم.

#### Status:

این دستور لیست تمام کالاهای تعریف شده را با اطلاعات آن ها نمایش می دهد. این اطلاعات شامل نام، قیمت خرید، قیمت فروش و تعداد موجود از آن در انبار است. به مثال زیر توجه کنید. (بخش \$ Status خط ورودی است که کاربر وارد کرده است.)

```
$ Status
A 1000 1350 10
B 2500 2700 0
C 5300 5600 2
```

#### Financial:

این دستور اطلاعات مالی را نمایش می دهد. شما باید میزان فروش، میزان خرید و سود حاصل را نمایش دهید. (سود حاصل همان تفاضل میزان فروش از میزان خرید است) فرمت نمایش به همان ترتیب گفته شده می باشد

و هر بخش با فاصله از هم جدا شده اند.

**Exit:**

این دستور اجرای برنامه را متوقف می‌کند.

## نکته

سعی کنید از تمام امکانات Object-Oriented و Inheritance در پیاده‌سازی خود استفاده کنید. لازم به ذکر است که پیاده‌سازی این تمرین تنها در یک کلاس با یک تابع main مجاز نمی‌باشد. برای حل مسئله حتما باید از روش Object-Oriented استفاده شود.

## نحوه ارسال

کل فایل‌های مورد نیاز برای اجرا شدن تمرین را به همراه یک فایل PDF که شامل توضیحات برنامه است را در قالب یک فایل ZIP ارسال کنید.

## شرکت نرم‌افزاری

قصد داریم برای یک شرکت نرم‌افزاری سامانه ای طراحی کنیم که اطلاعات کارمندان مختلف را ذخیره کند و همچنین فرآیند محاسبه حقوق آن‌ها را آسان تر کند. هر کارمند در یکی از سطوح‌های زیر می‌باشد:

- کارآموز
- جونیور
- سنیور
- مدیر ارشد

اطلاعات زیر نیز برای کارمندان باید ذخیره شود:

- نام
- نام خانوادگی
- تخصص
- کد ملی
- مدت زمان اضافه کاری (همه‌ی سطح‌ها غیر از کارآموزها)
- مدت زمان گذشته از کارآموزی (فقط مخصوص کارآموزان)
- کارآموزان تحت آموزش (فقط مخصوص سنیورها)

برای محاسبه حقوق، برای هر سطح پایه حقوقی تعیین می‌شود. و همچنین برای هر سطح میزان مالیات متفاوتی وجود دارد. برای هر کارمند می‌توان اضافه کاری بر حسب ساعت اضافه کرد. نرخ مزد اضافه کاری نیز قابل تغییر است.

در این تمرین شما باید برنامه ای را پیاده سازی کنید که بتواند از محیط کنسولی (command line) از کاربر ورودی بگیرد. هر ورودی در یک خط داده می‌شود که شامل چند بخش است. در بخش ابتدایی نام دستور می‌آید و سپس پارامترهای مربوط به آن داده می‌شود. مانند فرم زیر:

```
COMMAND argument1 argument2 ....
```

**نکته:** نام دستور و پارامترها با فاصله (Space) از یک دیگری جدا شده اند.

دستورات مورد نیاز در جدول زیر داده شده است.

--

Command Name	Argument 1	Argument 2	Argument 3	Argument 4	Argument 5
AddEmployee	First Name	Last Name	Level	National ID	Profession
AddTraineeToSenior	National ID for Trainee	National ID for Senior	-	-	-
SetTraineeTime	National ID for Trainee	Time Elapsed	-	-	-
PrintEmployee	National ID	-	-	-	-
SetBaseSalary	Level	Base Salary	-	-	-
SetTax	Level	Tax Percentage	-	-	-
SetExtraWorkFee	Fee	-	-	-	-
SetExtraWork	National ID	Extra Work (h)	-	-	-
ComputeSalary	National ID	-	-	-	-
Exit	-	-	-	-	-

توضیحات هر دستور

## AddEmployee

در این دستور با گرفتن نام، نام خانوادگی، سطح، کد ملی و تخصص یک کارمند به سیستم اضافه کنید. رشته‌ی Level یکی از رشته‌های زیر خواهد بود.

- Trainee
- Junior
- Senior
- Manager

## AddTraineeToSenior

این دستور با گرفتن کد ملی کارمند سنیور و کد ملی کارآموز، کارآموز را به لیست کارآموزان تحت آموزش سنیور مربوطه اضافه می‌کند.

## SetTraineeTime

این دستور با گرفتن کد ملی کارآموز و مدت زمان سپرانده شده به عنوان کارآموز (بر حسب ساعت) متغیر مربوطه را آپدیت می‌کند.

## PrintEmployee

این دستور با گرفتن کد ملی، باید اطلاعات مربوط کارمند مربوطه را چاپ کند. توجه داشته باشید که با توجه به سطح کارمند این اطلاعات چاپ شده می‌تواند متفاوت باشد. برای مثال تعداد کارآموز های تحت آموزش یک سنیور نیز برای یک سنیور باید چاپ شود.

## SetBaseSalary

این دستور با گرفتن سطح و میزان حقوق پایه (بر حسب تومان) اطلاعات مورد نظر را جهت محاسبه حقوق افراد ذخیره می‌کند.

## SetTax

این دستور مانند دستور SetBaseSalary عمل می‌کند . با این تفاوت که ورودی دوم میزان مالیات (بر حسب درصد) می‌باشد.

## SetExtraWorkFee

این دستور با ورودی گرفتن میزان حقوق اضافه برای هر ساعت، اطلاعات مورد نظر را برای محاسبه حقوق ذخیره می‌کند.

## SetExtraWork

این دستور با ورودی گرفتن کد ملی کارمند و میزان اضافه کاری (بر حسب ساعت) اطلاعات لازم جهت محاسبه درست حقوق را ذخیره می‌کند.

## ComputeSalary

این دستور نیز با ورودی گرفتن کد ملی کارمند، حقوق او را محاسبه کرده و چاپ می‌کند.

## Exit

این دستور اجرای برنامه را متوقف می‌کند.

## نمایش پیغام مناسب برای عملیات موفق و ناموفق

برنامه شما باید هنگام وجود خطای منطقی در دستور پیغام خطای مناسبی نمایش دهد. برای مثال در صورت وارد کردن کد ملی اشتباه برای دستور ComputeSalary باید پیغام مناسب مبنی بر عدم وجود چنین کارمندی نمایش داد. همچنین برای اعمال موفقیت آمیز نیز پیغام مناسب چاپ کنید.

## نکته

سعی کنید از تمام امکانات Object-Oriented و Inheritance در پیاده‌سازی خود استفاده کنید. لازم به ذکر است که پیاده‌سازی این تمرین تنها در یک کلاس با یک تابع main مجاز نمی‌باشد. برای حل مسئله حتما باید از روش Object-Oriented استفاده شود.

## نحوه ارسال

کل فایل‌های مورد نیاز برای اجرا شدن تمرین را به همراه یک فایل PDF که شامل توضیحات برنامه است را در قالب یک فایل ZIP ارسال کنید.