



مقدمه

هدف از این تمرین آشنایی شما با مفاهیم اولیه طراحی شیء‌گرا^۱ و طراحی یک نرم‌افزار به کمک آن است. از آنجایی که استفاده و درک درست این مفاهیم در پیاده‌سازی سایر تمرین‌های این درس لازم است، پیشنهاد می‌شود به این تمرین زمان کافی را اختصاص دهید.

برنامه مدیریت حقوق کارمندان

در این پروژه قصد داریم تا سیستمی طراحی و پیاده‌سازی کنیم که با در دست داشتن اطلاعات مربوط به کارکنان و تیم‌هایی که در آن مشغول به کار هستند، گزارش‌های مالی‌ای را فراهم کند که جزئیات آن در ادامه خواهد آمد. این برنامه به این صورت کار می‌کند که ابتدا اطلاعاتی مرتبط با کارکنان و ساعات کاری آن‌ها را از طریق یک سری مجموعه داده^۲ دریافت می‌کند؛ سپس از طریق رابط خط فرمان^۳ با اجرای دستورات مختلف، می‌توان گزارش‌هایی مرتبط با حقوق کارمندان و ساعات کاری آن‌ها از برنامه دریافت کرد.

مجموعه‌های داده

در ادامه درباره مجموعه داده‌هایی که این برنامه در قالب فایل‌های CSV^۴ دریافت می‌کند، توضیحاتی آورده شده است.

^۱ Object-Oriented Design

^۲ Dataset

^۳ Command Line Interface

^۴ Comma-Separated Values

● کارمندان

اطلاعات مربوط به کارمندان در فایل به نام employees.csv آورده شده است.

نام ویژگی	توضیح	نوع داده
id	شناسه یکتا	عدد صحیح
name	نام و نام خانوادگی	رشته‌ای از کاراکترها
age	سن	عدد صحیح
level	سطح تخصص	رشته‌ای از کاراکترها

○ ویژگی level، چهار مقدار مشخص junior, expert, senior, team_lead را می‌تواند بگیرد.

● ساعات کاری

در این تمرین فرض شده که کارمندان به صورت ساعتی حقوق دریافت می‌کنند. همچنین این فرض را در نظر بگیرید که تمامی این اطلاعات مربوط به ۳۰ روز کاری است که در آن هیچ تعطیلی‌ای وجود ندارد (یعنی تمام این ۳۰ روز قابلیت این را دارند که روز کاری باشند). همچنین در این پروژه نیازی به در نظر گرفتن مفاهیم ماه و سال نیست و صرفاً باید مفهوم روز و ساعت در نظر گرفته شود؛ اما داشتن ساختار داده‌ی مناسب برای نگهداری از زمان و انجام عملیات‌ها بر روی آن بسیار حائز اهمیت است. اطلاعات مربوط به ساعات کاری کارمندان در فایل به نام working_hours.csv آورده شده است.

نام ویژگی	توضیح	نوع داده
employee_id	شناسه کارمند	عدد صحیح
day	یک روز از ماه	عدد صحیح
working_interval	بازه ساعت کاری کارمند	رشته‌ای از کاراکترها

- ویژگی day، مقداری از 1 تا 30 را می‌تواند اختیار کند.
- ویژگی working_interval، بازه‌ی ساعتی است که کارمند در آن مشغول به کار بوده‌است. این بازه به صورت #start-#end هستند. (برای مثال 16-18 یک بازه‌ی ساعت کاری را مشخص می‌کند).
- دقت کنید ممکن است برای یک کارمند در یک روز، چند بازه کاری وجود داشته باشد؛ تضمین می‌شود بازه‌ها با یکدیگر همپوشانی⁵ ندارد.
- توجه داشته باشید که بازه بصورت [inclusive, exclusive] در نظر گرفته می‌شوند.
- مقداری که ساعت‌ها می‌توانند به خود بگیرند عددی صحیح از 0 تا 24 می‌باشد.
- در طراحی ساختار داده‌ی مناسب برای نگهداری اطلاعات مربوط به ساعات کاری با دقت عمل کنید، چون همانطور که در ادامه خواهید دید، این داده‌ها می‌توانند با استفاده از ورودی استاندارد تغییر کنند.

● تیم‌ها

اطلاعات مربوط به تیم‌های شرکت، در فایل به نام teams.csv آورده شده است.

⁵ overlap

نام ویژگی	توضیح	نوع داده
team_id	شناسه یکتای تیم	عدد صحیح
team_head_id	شناسه کارمند مسئول تیم	عدد صحیح
member_ids	لیست شناسه‌های اعضای تیم	رشته‌ای از کاراکترها
bonus_min_working_hours	حداقل ساعت کاری برای دریافت پاداش	عدد صحیح
bonus_working_hours_max_variance	حداکثر واریانس ساعت کاری برای دریافت پاداش	عدد اعشاری

- تضمین می‌شود سطح تخصص (level) مسئول تیم، team_lead است.
- ویژگی member_ids، لیستی از شناسه اعضای تیم است. این شناسه‌ها به صورت #id هستند که با کاراکتر \$ از هم جدا شده‌اند، همچنین شناسه‌ی مسئول تیم هم در این لیست خواهد بود.
- همچنین تیم‌ها دارای یک ویژگی به نام bonus از جنس عدد صحیح هستند که مقدار پاداش ثابتی که به اعضای تیم تعلق می‌گیرد را به صورت درصدی نشان می‌دهند. این ویژگی در ابتدا مقداری برابر صفر دارد حداکثر مقداری که می‌تواند بگیرد برابر ۱۰۰ خواهد بود.
- دو ویژگی آخر، در بخش امتیازی پروژه کاربرد خواهند داشت که توضیحات مربوط به آن‌ها در همان بخش آمده است.

● پیکربندی پارامترهای محاسبه‌ی حقوق

اطلاعات مربوط به پارامترهای محاسبه حقوق در فایل به نام salary_configs.csv آورده شده است.

نام ویژگی	توضیح	نوع داده
level	سطح تخصص	رشته‌ای از کاراکترها
base_salary	مقدار حقوق پایه	عدد صحیح
salary_per_hour	مقدار حقوق ساعتی	عدد صحیح
salary_per_extra_hour	مقدار حقوق اضافه کاری	عدد صحیح
official_working_hours	مقدار ساعت کاری رسمی	عدد صحیح
tax_percentage	درصد مالیات	عدد صحیح

- ویژگی level، چهار مقدار مشخص junior, expert, senior, team_lead را می‌تواند بگیرد.
- تضمین می‌شود برای هر سطح تخصص، پارامترهای محاسبه حقوق در فایل آمده است.
- اگر مجموع ساعت کاری یک کارمند در ماه از official_working_hours بیشتر باشد، حقوق ساعات‌های اضافه با پارامتر salary_per_extra_hour محاسبه می‌شود.
- ویژگی tax_percentage میزان مالیاتی‌ست که از حقوق یک کارمند به صورت درصدی کسر می‌شود. تضمین می‌شود که این مقدار همواره عددی بین 0 تا 100 خواهد بود.

دستورات گزارش‌گیری

در این قسمت مجموعه دستوراتی که در گزارش‌گیری کاربرد دارند معرفی می‌شوند.

● گزارش حقوق کارمندان

با وارد کردن این دستور، برای تمام کارمندان اطلاعات شناسه، نام، مجموع ساعت کاری و حقوق دریافتی نمایش داده می‌شود. فرمت درست خروجی را می‌توانید در نمونه‌هایی که در کنار صورت پروژه به شما داده می‌شود چک کنید.

قالب ورودی

```
report_salaries
```

قالب خروجی

```
//(for each employee)
ID: <employee_id>
Name: <employee_name>
Total Working Hours: <total_WH>
Total Earning: <total_earning>
---
```

نمونه ورودی

```
report_salaries
```

نمونه خروجی

```
ID: 1
Name: Mahdi
Total Working Hours: 158
Total Earning: 50000
---
ID: 2
Name: Mahmood
Total Working Hours: 124
Total Earning: 14000
---
```

● گزارش حقوق یک کارمند

با وارد کردن این دستور، برای کارمند با شناسه مشخص شده، اطلاعات جزئی مربوط به ساعات کاری و حقوق او نمایش داده می‌شود.

قالب ورودی

```
report_employee_salary <employee_id>
```

قالب خروجی

```
ID: <employee_id>
Name: <employee_name>
Age: <employee_age>
Level: <employee_level>
Team ID: <team_id>
Total Working Hours: <total_WH>
Absent Days: <days_with_zero_WH>
Salary: <salary>
Bonus: <bonus_amount>
Tax: <tax_amount>
Total Earning: <total_earning>
```

نمونه ورودی

```
report_employee_salary 1
```

نمونه خروجی

```
ID: 1
Name: Mahdi
Age: 19
Level: junior
Team ID: 1
Total Working Hours: 158
Absent Days: 4
Salary: 55000
```

Bonus: 0 Tax: 5000 Total Earning: 50000

- اگر کارمندی متناظر با شناسه‌ای که در آرگومان دستور وارد می‌شود در سیستم یافت نشود، پیغام EMPLOYEE_NOT_FOUND نمایش داده می‌شود.
- اگر کاربر عضو تیمی نباشد، مقدار فیلد Team ID برابر N/A خواهد بود.
- در رابطه با فیلد Absent Days دقت کنید که فرض شده که روز تعطیل نداریم و کارمند می‌تواند در تمام ۳۰ روز ماه کار کند.
- ویژگی Salary بیانگر حقوق خالص کارمند است که با استفاده از پارامترهای مربوط به حقوق محاسبه شده‌اند و عدد صحیح می‌باشد. اگر در محاسبات به عدد اعشاری رسیدید، آن را به عدد صحیح گرد کنید.
- ویژگی Bonus مقدار پاداشی است که کارمند اگر عضو تیمی بوده باشد و برای آن تیم یک درصد پاداش مشخص شده باشد، به آن کارمند تعلق می‌گیرد. در بخش گزارش حقوق تیم بیشتر درباره این ویژگی توضیح داده شده است. دقت کنید که در این فیلد، درصد نمایش ندهید و مقدار دقیق پاداش را با توجه به حقوق خالص کارمند نمایش دهید. اگر در محاسبات به عدد اعشاری رسیدید، آن را به عدد صحیح گرد کنید.
- ویژگی Tax، مقدار مالیاتی که روی مجموع Salary و Bonus اعمال می‌شود را نمایش می‌دهد. این مقدار یک عدد صحیح است. اگر در محاسبات به عدد اعشاری رسیدید، آن را به عدد صحیح گرد کنید.
- منظور از Total Earning، مقدار دریافتی کارمند، پس اعمال پاداش و مالیات روی حقوق اولیه است.

● گزارش حقوق یک تیم

با وارد کردن این دستور، برای یک تیم با شناسه مشخص شده، اطلاعات مربوط به ساعات کاری آن تیم و حقوق اعضا نمایش داده می‌شود.

قالب ورودی

```
report_team_salary <team_id>
```

قالب خروجی

```
ID: <team_id>
Head ID: <head_team_id>
Head Name: <head_team_name>
Team Total Working Hours: <sum_of_all_members_WH>
Average Member Working Hour: <avg_member_WH>
Bonus: <bonus_percentage>
---
// (for each member)
Member ID: <member_id>
Total Earning: <total_earning>
---
```

نمونه ورودی

```
report_team_salary 1
```

نمونه خروجی

```
ID: 1
Head ID: 1
Head Name: Mahdi
Team Total Working Hours: 388
Average Member Working Hours: 180
Bonus: 0
---
Member ID: 1
Total Earning: 50000
---
Member ID: 2
Total Earning: 14000
---
```

- اگر تیمی متناظر با شناسه‌ای که در آرگومان دستور وارد می‌شود در سیستم یافت نشود، پیغام TEAM_NOT_FOUND نمایش داده می‌شود.
- میانگین گزارش شده باید با دقت یک دهم اعشار گرد شده باشد.
- فیلد Bonus درصد پاداشی که به اعضای آن تیم داده شده است را به صورت عدد صحیح نمایش می‌دهد. این پاداش به صورت درصدی به حقوق اعضا اضافه می‌شود. مثلاً اگر یک عضو حقوقش 5000000 باشد و درصد پاداش 10 باشد، به حقوق او 500000 اضافه می‌شود.
- دقت کنید که مسئول تیم نیز یکی از اعضای تیم است.

● گزارش مجموع ساعات کاری در روز

با وارد کردن این دستور، مجموع ساعات کاری تمام کارمندان در روزهای مشخص شده نمایش داده می‌شود.

قالب ورودی

```
report_total_hours_per_day <start_day> <end_day>
```

قالب خروجی

```
// (for each day)
Day #num_of_day: <total_working_hours_of_that_day>
---
Day(s) with Max Working Hours: <space_separated_list_of_days>
Day(s) with Min Working Hours: <space_separated_list_of_days>
```

نمونه ورودی

```
report_total_hours_per_day 1 4
```

نمونه خروجی

```
Day #1: 4
Day #2: 8
Day #3: 8
Day #4: 7
---
Day(s) with Max Working Hours: 2 3
Day(s) with Min Working Hours: 1
```

○ اگر مقدار آرگومان‌های وارد شده بین 1 تا 30 نباشند یا اگر مقدار آرگومان اول بیشتر از آرگومان دوم باشد، باید

پیغام INVALID_ARGUMENTS چاپ شود.

○ دقت کنید که بازه که آرگومان‌ها مشخص می‌کنند، شامل خود آرگومان‌ها نیز می‌شود.⁶

● گزارش میانگین تعداد کارمندان مشغول به کار در ساعات مشخص

با وارد کردن این دستور، میانگین تعداد افراد مشغول به کار در کل ماه، در ساعت‌های مشخص شده نمایش داده می‌شود.

قالب ورودی

```
report_employee_per_hour <start_hour> <end_hour>
```

قالب خروجی

```
// (for hour_i in [start_hour, end_hour))
<hour_i>-<hour_i+1>:
<period_average_working_employees_in_month>
---
Period(s) with Max Working Employees: <list_of_these_periods>
Period(s) with Min Working Employees: <list_of_these_periods>
```

⁶ inclusive

نمونه ورودی

```
report_employee_per_hour 8 11
```

نمونه خروجی

```
8-9: 6.4  
9-10: 6.0  
10-11: 14.5  
---
```

```
Period(s) with Max Working Employees: 10-11
```

```
Period(s) with Min Working Employees: 9-10 10-11
```

○ میانگین گزارش شده باید با دقت یک رقم اعشار اعلام شود.

○ اگر مقدار آرگومان های وارد شده از 0 تا 24 نباشند یا اگر مقدار آرگومان اول کمتر از آرگومان دوم نباشد، باید

پیغام INVALID_ARGUMENTS چاپ شود.

دستورات پیکربندی

در این قسمت با دستوراتی که بر پارامترهای محاسباتی یا داده های سیستم تاثیر می گذارند آشنا می شوید.

● نمایش پارامترهای محاسبه حقوق

با وارد کردن این دستور، تمام پارامترهای محاسبه حقوق برای level مشخص شده در آرگومان نمایش داده می شوند.

قالب ورودی

```
show_salary_config <level>
```

قالب خروجی

```
Base Salary: <base_salary>  
Salary Per Hour: <salary_per_hour>  
Salary Per Extra Hour: <salary_per_extra_hour>  
Official Working Hours: <official_working_hours>
```

Tax: <tax_percentage>%

نمونه ورودی

```
show_salary_config expert
```

نمونه خروجی

```
Base Salary: 1000
Salary Per Hour: 1000
Salary Per Extra Hour: 600
Official Working Hours: 7
Tax: 5%
```

○ اگر برای آرگومان level مقداری غیر چهار مقدار مشخص شده وارد شده باشد، باید پیغام

INVALID_LEVEL چاپ شود.

● تغییر پارامترهای محاسبه حقوق

با وارد کردن این دستور، می‌توان پارامترهای محاسبه حقوق برای یک سطح تخصص را تغییر داد.

قالب ورودی

```
Update_salary_config <level> <base_salary> <salary_per_hour>
<salary_per_extra_hour> <official_working_hours> <tax_percentage>
```

قالب خروجی

OK

نمونه ورودی

```
update_salary_parameters senior 1000 900 800 6 6
```

نمونه خروجی

OK

- دستور به همراه همه آرگومان‌هایش در یک خط به برنامه داده می‌شود.
- اگر برای آرگومان level مقداری غیر چهار مقدار مشخص شده وارد شده باشد، باید پیغام INVALID_LEVEL چاپ شود.
- تضمین می‌شود برای دیگر آرگومان‌های دستور، مقداری معقول و متناسب با آن آرگومان وارد شود.
- اگر بخواهیم مقدار یک پارامتر بی‌تغییر باقی بماند، مقدار آرگومان متناظر آن را در دستور برابر کاراکتر '-' (خط فاصله) قرار می‌دهیم.

● اضافه کردن بازه ساعت کاری

این دستور، یک بازه ساعت کاری برای یک کارمند مشخص در یک روز مشخص اضافه می‌کند.

قالب ورودی

```
add_working_hours <employee_id> <day> <period_start> <period_end>
```

قالب خروجی

OK

نمونه ورودی

```
add_working_hours 1 2 22 23
```

نمونه خروجی

```
OK
```

- اگر شناسه کاربر وارد شده در سیستم موجود نباشد باید پیغام `EMPLOYEE_NOT_FOUND` چاپ شود.
- اگر مقدار آرگومان‌های عددی معقول نباشند یا ساعت شروع بازه از ساعت پایان بازه کمتر نباشد، باید پیغام `INVALID_ARGUMENTS` چاپ شود.
- اگر بازه وارد شده در آن روز برای یک کارمند با شناسه مشخص شده، با بازه ساعت کاری دیگری از آن کارمند (در همان روز) که زمان وارد کردن دستور در سیستم ثبت شده و موجود می‌باشد، همپوشانی داشته باشد باید پیغام `INVALID_INTERVAL` چاپ شود. مثلاً اگر برای یک کارمند در روز 1 بازه 12-18 در سیستم موجود باشد و در دستور، بازه 16-20 در روز 1 مشخص شده باشد، اجازه ثبت بازه جدید داده نمی‌شود و پیغام مذکور چاپ می‌شود.

● حذف بازه‌های ساعت کاری

با وارد کردن این دستور، تمام بازه‌های ساعت کاری یک کارمند مشخص در یک روز مشخص حذف می‌شود.

قالب ورودی

```
delete_working_hours <employee_id> <day>
```

قالب خروجی

```
OK
```

نمونه ورودی

```
delete_working_hours 1 2
```

نمونه خروجی

OK

○ اگر شناسه کاربر وارد شده در سیستم موجود نباشد باید پیام EMPLOYEE_NOT_FOUND چاپ شود.

○ اگر مقدار آرگومان day معقول نباشد باید پیام INVALID_ARGUMENTS چاپ شود.

● تغییر درصد پاداش

با وارد کردن این دستور، درصد پاداشی که بر روی حقوق اعضای تیم اعمال می‌شود به مقدار تعیین شده تغییر می‌کند.

قالب ورودی

```
update_team_bonus <team_id> <bonus_percentage>
```

قالب خروجی

OK

نمونه ورودی

```
update_team_bonus 1 10
```

نمونه خروجی

OK

- در ابتدای شروع برنامه، درصد پاداش تمام تیم‌ها صفر است. همچنین این درصد فقط عدد صحیح بین 0 تا 100 می‌تواند به خود بگیرد.
- اگر شناسه تیم وارد شده در سیستم موجود نباشد باید پیغام TEAM_NOT_FOUND چاپ شود.
- اگر مقدار آرگومان درصد پاداش بین 0 تا 100 نباشد باید پیغام INVALID_ARGUMENTS چاپ شود.

● بخش امتیازی - یافتن تیم‌های لایق پاداش

در این بخش قصد داریم تا با اطلاعاتی که از تیم‌ها و ساعت کاری اعضای هر تیم داریم، تیم‌هایی را پیدا کنیم که مستعد دریافت پاداش هستند. به این منظور از دو عنصر مجموع و واریانس ساعت کاری هر تیم استفاده می‌کنیم؛ به این گونه که مجموع و واریانس ساعت کاری هر تیمی را با استفاده از ساعات کاری اعضای آن تیم محاسبه می‌کنیم. در داده‌های مربوط به هر تیم دو ویژگی `bonus_min_working_hours` و `bonus_working_hours_max_variance` داده شده است؛ به ازای هر تیم، اگر مجموع ساعت کاری کل تیم از `bonus_min_working_hours` بیشتر باشد و همچنین واریانس ساعات کاری افراد تیم از `bonus_max_working_hours_variance` کمتر باشد، آن تیم مستعد دریافت پاداش می‌باشد. در نهایت باید شناسه تیم‌های لایق به صورت مرتب شده بر اساس مجموع ساعت کاری نمایش داده شوند.

اگر هیچ تیمی لایق پاداش نبود، باید پیغام NO_BONUS_TEAMS چاپ شود.

قالب ورودی

```
find_teams_for_bonus
```

قالب خروجی

```
// (for each team that could get bonus)
Team ID: <team_id>
```

نمونه ورودی

find_teams_for_bonus

نمونه خروجی

Team ID: 1

Team ID: 2

نحوهٔ تحویل

- می‌توانید این تمرین را در قالب گروه‌های دو نفره انجام دهید. گروهی انجام دادن این تمرین، الزامی نیست و می‌توانید به صورت تکی نیز آن را تحویل دهید.
- توجه داشته باشید که در صورت تمایل، می‌توانید گروه خود را در طی تمرین‌ها تغییر دهید.
- توجه داشته باشید که در صورت گروهی انجام دادن تمرین، هر دو عضو گروه باید اشراف کامل روی تمرین و کد آپلود شده داشته باشند و این موضوع در تحویل تمرین در نظر گرفته می‌شود.
- به غیر از خطاهای ذکر شده در صورت پروژه، نیاز به رسیدگی به هیچ خطای دیگر نمی‌باشد و تضمین می‌شود ورودی‌ها به درستی به برنامه داده می‌شوند.
- کد خود را در قالب یک فایل با نام A4-SID.cpp در صفحهٔ eLearn درس بارگذاری کنید که SID شمارهٔ دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۱۰۰۰ باشد، نام پروندهٔ شما باید A4-810101000.cpp باشد که شامل کد شما است. همچنین در صورتی که به صورت گروهی این تمرین را تحویل می‌دهید و شماره دانشجویی اعضای تیم، ۸۱۰۱۰۱۹۹۸ و ۸۱۰۱۰۱۹۹۹ باشد، نام این پرونده را A4-810101998-810101999.cpp بگذارید.
- برنامهٔ شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.

- تمیزی کد، ذخیره کردن اطلاعات در ساختارهای مناسب، شکستن مرحله به مرحله مسئله و طراحی مناسب، در کنار تولید خروجی دقیق و درست، بخش مهمی از نمره شما را تعیین خواهد کرد.
- درستی برنامه شما از طریق آزمون های خودکار سنجیده می شود؛ بنابراین پیشنهاد می شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.