

مینی پروژه - شرکت

در این مینی‌پروژه قصد داریم سیستم محاسبه و پرداخت حقوق یک شرکت را پیاده‌سازی کنیم.

هدف این مینی پروژه آشنایی با مباحث ارث‌بری و چندریختی است.

در این سیستم کاربران دو دسته‌اند: کارمندان و مدیر شرکت.

کارمندان خود شامل سه دسته هستند:

۱. کارمند تمام وقت

۲. کارمند نیمه وقت

۳. کارمند پروژه‌ای

توالی ارث‌بری مناسب برای این مینی‌پروژه را تعریف کنید.

دقت کنید که نباید توانست از کلاس `User` و `Employee` شی ساخت.

کلاس کاربر (User)

ویژگی‌ها:

۱. نام و نام خانوادگی (با یک فاصله جدا می شوند)

۲. شماره تلفن

• سازنده این دو فیلد را دریافت می‌کند.

عملکردها:

۱. متد `getBasicInformation`

• متد `getBasicInformation` اطلاعات هر کاربر که شامل نام و نام خانوادگی و شماره‌تلفن را در

قالب یک رشته برمی‌گرداند. همچنین این متد باید در کلاس کارمند نیز بازنویسی شود.

کلاس کارمند (Employee)

- از کلاس User ارث‌بری می‌کند.

ویژگی‌های اضافه:

۱. تاریخ استخدام

عملکردها:

۱. متد انجام کار جدید work

۲. محاسبه میزان حقوق دریافتی getSalary

۳. متد getBasicInformation

- متد getSalary حقوق هر کارمند را محاسبه و ریترن می‌کند این متود باید توسط فرزندان کلاس کارمند پیاده‌سازی شود و در این کلاس فاقد بدنه است.
- متد work عددی طبیعی را دریافت و بسته به نوع کارمند میزان کار او را محاسبه می‌کند. این متود باید توسط فرزندان کلاس کارمند پیاده‌سازی شود و در این کلاس فاقد بدنه است. توضیحات و راهنمایی‌های اضافه‌تر این دو متود را در آخر داک مطالعه کنید.
- متد getBasicInformation اطلاعات هر کاربر که شامل نام و نام خانوادگی و شماره‌تلفن و تاریخ استخدام را در قالب یک رشته برمی‌گرداند. همچنین این متد نباید در کلاس‌های فرزند امکان بازنویسی داشته باشد.

کلاس کارمند تمام‌وقت

- از کلاس Employee ارث‌بری می‌کند.

ویژگی‌ها:

۱. تعداد کل روزهای کاری که حقوقش محاسبه نشده است.

۲. نرخ حقوق ماهانه (این فیلد ثابت است و پس از دریافت از طریق سازنده دیگر قابل تغییر نیست).

عملکردها:

- متد `getSalary` بر اساس تعداد روز محاسبه نشده حقوق فرد را محاسبه می‌کند و ریترن می‌کند. (این متد به صورت ماهانه عمل می‌کند و پس از محاسبه حقوق و ریترن کردن آن، فیلد تعداد روز محاسبه نشده حقوق را به‌روزرسانی می‌کند. در صورتی که روزها کمتر از یک ماه بود متد 0 برمی‌گرداند و به معنای این است که فعلاً حقوق قابل محاسبه نیست.)

یک مثال عددی: اگر فردی 40 روز کار کرده و حقوقش محاسبه نشده، پس از فراخوانی این متد حقوق یک ماه او محاسبه و ریترن می‌شود، سپس روزهای محاسبه نشده حقوق این فرد به 10 روز تغییر پیدا می‌کند.

- متد `work` تعداد روزهای جدید را گرفته و با روزهای محاسبه نشده جمع می‌کند.

کلاس کارمند نیمه‌وقت

- از کلاس `Employee` ارث‌بری می‌کند.

ویژگی‌ها:

1. تعداد کل ساعاتی کاری که حقوقش محاسبه نشده است.
2. نرخ حقوق ساعتی (این فیلد ثابت است و پس از دریافت از طریق سازنده دیگر قابل تغییر نیست.)

عملکردها:

- متد `getSalary` بر اساس تعداد ساعات محاسبه نشده حقوق فرد را محاسبه و ریترن می‌کند. (بعد از اجرای این متد، فیلد کار محاسبه نشده باید آپدیت شود.)
- متد `work` تعداد ساعات جدید را گرفته و با ساعات محاسبه نشده جمع می‌کند.

کلاس کارمند پروژه‌ای

- از کلاس `Employee` ارث‌بری می‌کند.

ویژگی‌ها:

1. مجموع کل پروژه‌های انجام شده که حقوق آن‌ها محاسبه نشده.
2. حق الزحمه هر پروژه (این فیلد ثابت است و پس از دریافت از طریق سازنده دیگر قابل تغییر نیست.)

عملکردها:

- متد `getSalary` بر اساس تعداد پروژه‌های محاسبه نشده حقوق فرد را محاسبه می‌کند و ریترن می‌کند. (فیلد کار محاسبه نشده باید آپدیت شود).
- متد `work` تعداد پروژه‌های جدید را گرفته و با پروژه‌های محاسبه نشده جمع می‌کند.

ارث‌بری از کلاس‌های کارمند تمام وقت و نیمه وقت و پروژه‌ای مجاز نیست و نباید این امکان وجود داشته باشد.

کلاس مدیر (Admin)

- از کلاس `User` ارث‌بری می‌کند.

ویژگی‌ها:

- ۱. یک لیست غیر استاتیک که همه کارمندان در آن ذخیره شوند.
- با ساخته شدن هر فرزندی از `Employee` این لیست به‌روزرسانی شده و آن شی به این لیست اضافه می‌شود.
- توجه کنید که نباید ست کردن این لیست خارج از کلاس مدیر مجاز باشد.

عملکردها:

- متد نمایش همه کاربران شرکت
 - متد تعریف کارمند جدید - تعریف کارمند جدید باید توسط مدیر سامانه انجام شود.
- در این شرکت فقط یک مدیر وجود دارد، پس پیاده‌سازی کلاس مدیر باید بگونه‌ای باشد که از این کلاس تنها بتوان یک شی ساخت و در صورت گرفتن نمونه از این کلاس در خارج از آن با کامپایل ارور مواجه شویم. (برای پیاده‌سازی این بخش از الگوی سینگلتون `Singleton Pattern` استفاده کنید.)

راهنمایی و توضیح دو متد اصلی این مینی پروژه:

متد `getSalary`

هر کارمندی براساس نوعش یک فیلد مخصوص توی کلاسش داره برای مثال: کارمند تمام وقت به صورت روزانه کار میکنه و یه فیلد داره که در اون تعداد روزهای کارکردنش ثبت میشه. و وقتی متد `getSalary` برای این نوع کارمند صدا زده میشه براساس تعداد روزهایی که روی اون فیلد داره حقوقش محاسبه میشه. نکته اینجاست که براساس تعریف این متد برای کارمند تمام وقت باید حتما چک بشه که آیا روزهای کار کردن این فرد به یک ماه رسیده یا نه؟ اگه رسیده باشه که به صورت ماهانه این حقوق محاسبه میشه و ریترن میشه. اما اگه نرسیده باشه باید صفر ریترن کنه چون یک ماهش (حداقل) کامل نشده. منطقا وقتی این حقوق این روزها محاسبه میشه فیلد روزهای کاری این فرد باید آپدیت بشه و کم بشه...

- مثال عددی: اگر کارمند تمام وقتی 40 روز کار کرده، بعد از فراخوانی متد `getSalary` حقوق 30 روزش محاسبه و ریترن میشه و فیلد روزهای محاسبه نشده حقوقش به 10 تغییر پیدا می‌کنه.

متد `work`

یک ورودی میگیره که مقدار کارکرد اون کارمند رو در واقع باهاش ثبت میکنیم. مثلا عددی که متد `work` برای کارمند تمام وقت میگیره مشخصه تعداد روزهای کارکردش و با روزهای کارکرد قبلش قراره که جمع بشه و به همین ترتیب برای کارمند پاره وقت این عدد مقدار ساعات کاری جدیدشه.

روند کلی:

این مینی پروژه نیازی به رابط کاربری ندارد. تست تمامی امکانات برنامه را در مین انجام دهید. به این صورت که در ابتدا با فراخوانی متد `Admin.getAdmin("name" , "phoneNumber")` تنها مدیر سامانه را بسازید و سپس 5 کارمند با نوع شغلی متفاوت بسازید. در نهایت از طریق اری لیست همه کارمندان ابتدا متد `work` و سپس `getSalary` را فراخوانی کنید. و نتایج محاسبه حقوق هر کارمند را چاپ کنید.

دقت کنید کپسوله سازی متغیرها و پیاده سازی متدهای `getter` و `setter` بخشی از معیار نمره دهی می باشد.

مثل همیشه رعایت نکات کدنویسی تمیز، (خصوصا استفاده نکردن از اسکرنر و پرینتر در کلاس ها) بسیار ضروری خواهد بود و در صورت عدم رعایت از یک سوم تا نصف نمره کسر خواهد شد!

توجه کنید که بستر اصلی پیاده‌سازی پروژه در گیت‌هاب است و کامیت‌های درست در بازه‌های زمانی کوتاه، بخشی از معیار نمره‌دهی می‌باشد.