پروژه نهایی: تاکسی تلفنی

میخواهیم برنامه ای جهت مدیریت یک تاکسی تلفنی بنویسیم. شرکت تاکسی تلفنی متشکل از اطلاعات سازمان، رانندگان، خودرو ها، مشتریان، رانندگان سرویس مدارس می‌باشد. در پیوست نمونه فایل هایی به عنوان ورودی برنامه در اختیار شما قرار داده شده است. جهت نوشتن این نرم افزار لازم است گام های زیر پیاده سازی فرمائید.

گام اول Enum:

1. یک Enum برای تحصیلات بنویسید که شامل دیپلم، فوق دیپلم، لیسانس، فوق لیسانس و دکتری
2. یک Enum برای کارخانه سازنده بنویسید که شامل سایپا، ایران خودرو، بهمن موتر و یا کرمان موتور

گام دوم interface:

1. یک واسط IBirthday جهت محاسبه تعداد روز های مانده تا تولد بنویسید. در این واسط متد CalculateBirthday قرار دارد که یک تاریخ را از ورودی دریافت میکند و روز های مانده تا تولد را محاسبه می‌کند.

گام سوم Class:

این برنامه شامل کلاس های شخص، راننده، مشتری، راننده سرویس و خودرو است کلاس های مرتبط را پیاده سازی کنید. از ارث بری برای تعریف روابط استفاده نمائید.

کلاس Person

کلاس شخص کلاسی انتزاعی است. نمی توان از روی کلاس شخص نمونه ساخت.

فیلدها: نام، نام خانوادگی، شماره موبایل، تاریخ تولد، آدرس

****تاریخ تولد فیلدی از نوع DateTime است.DateTime یک کلاس پیشفرض است که میتوانید از ان در کدهای خود استفاده کنید.**

متد ها:

1. برای کلاس سازنده مناسب بنویسید.
2. متد UpdateMobile جهت به روزرسانی شماره موبایل.

3. متد GetAddress جهت نمایش آدرس دارد. این متد **باید** توسط کلاس هایی پیاده سازی شود که از کلاس Person ارث می‌برند.
4. استفاده از واسط IBirthday جهت پیاده سازی متدCalculateBirthday از متد CompareTo کلاس DateTime استفاده کنید

هدف : یادگیری abstract, sealed, interface

کلاس Driver

کلاس راننده از کلاس Person ارث می‌برد.

فیلدها: کد راننده، کد ملی، تحصیلات، درصد شراکت

متدها:

1. برای کلاس سازنده مناسب بنویسید.
2. متد CalculateSalary را برای محاسبه حقوق بنویسید. ورودی این متد یک لیست از درآمدهایی است که راننده از مشتریان دریافت نموده است. خروجی متد حقوق راننده می‌باشد. در این متد:
   a. مجموع درآمدها محاسبه شود.
   b. با توجه به فیلد درصد شراکت و فرمول زیر حقوق راننده محاسبه شود.

حقوق راننده = درصد شراکت * مجموع درآمدها

   c. نوع داده ای آرگومان درآمدها می‌تواند String و یا double باشد. متد شما باید هر دو نوع را پشتیبانی و محاسبه نماید.

هدف : Generic, inheritance

کلاس School Driver

کلاس راننده مدارس از کلاس**Driver** ارث می‌برد.

فیلدها: سابقه کار (تعداد بر حسب سال)، مسیررفت/برگشت (رفت false یا رفت و برگشتtrue)

متدها:

1. برای کلاس سازنده مناسب بنویسید.

2. متد CalculateSalary را برای محاسبه حقوق override کنید. ورودی این متد یک لیست از درآمدهایی است که راننده از مشتریان دریافت نموده است. خروجی متد حقوق راننده می‌باشد. در این متد:
   a. مجموع درآمدها محاسبه شود.
   b. اگر فیلد مسیررفت/برگشت true باشد، مجموع درآمدها دو برابر شود.
   c. با توجه به فیلد درصد شراکت و فرمول زیر حقوق راننده محاسبه شود.

حقوق راننده = درصد شراکت * مجموع درآمدها

   d. نوع داده ای آرگومان درآمدها می‌تواند String و یا double باشد. متد شما باید هر دو نوع را پشتیبانی و محاسبه نماید.

هدف:override virtual

کلاس Customer

کلاس مشتری از کلاس Person ارث می‌برد.

فیلد: کد مشتری

متدها:

1. برای کلاس سازنده مناسب بنویسید.
2. تعریف عملگر + برای دو مشتری(c1+c2). حاصل این عملگر ارایه ای از مشتریان است. توجه داشته باشد کد شما باید برای جمع زدن سه مشتری (c1+c2+c3) نیز جواب دهد.

هدف : operator overloading

کلاس Car

فیلدها: سال تولید، کارخانه، مدل، کد راننده خودرو، رنگ تاریخ انقضا و شماره بیمه نامه

متدها:

1. برای کلاس سازنده مناسب بنویسید.
2. متدی که دو خودرو را از نظر مدل مقایسه میکند در صورت یکسان بودن true بر میگرداند.
3. indexer درنظر بگیرید که با استفاده دریافت تاریخ انقضا بیمه نامه سال انقضا را برگرداند.

هدف: indexer,static

گام چهارم Exception:

خطاهای زیر ممکن است در برنامه رخ دهد انها را در طول کد مدیریت کنید.

1. اگر شماره موبایل 12 رقمی نباشد پیام مناسب چاپ شود.
2. اگر موبایل وارد شده حروف باشد یاعددی نباشد خطای مناسب چاپ شود.
3. اگر درخواندن فایل ها مشکلی وجود داشته باشد خطای مناسب چاپ شود.

نمره اضافه

با استفاده از ویندوز فرم و برنامه ای که تاکنون نوشته اید را توسعه دهید.