

## متد Compare

یک متد به نام Compare به تاریخ اضافه کنید طوری که

### **d1.Compare(d2)**

در حالتی که  $d1 < d2$ ، مقدار منفی یک، در حالتی که  $d1 = d2$ ، مقدار صفر و در حالتی که  $d1 > d2$  مقدار یک را برگرداند.

(نیازی به ارسال کل کلاس نیست. صرفا کدهایی که خودتان نوشته اید را ارسال نمایید) کد ارسالی شما توسط کوئرا کامپایل نخواهد شد.

## تابع days\_between جدید

در تعریف تابع days\_between فرض شده است که پارامتر اول کوچکتر یا مساوی پارامتر دوم است. تعریف این تابع را طوری تغییر دهید که این محدودیت برداشته شود.

(نیازی به ارسال کل کلاس نیست. صرفا کدهایی که خودتان نوشته اید را ارسال نمایید) کد ارسالی شما توسط کوئرا کامپایل نخواهد شد.

## کلاس Time

هدف این تمرین طراحی کلاسی به نام Time است که همان طور که از نام آن پیداست، قرار است برای ذخیره کردن زمان استفاده می‌شود. این کلاس را طوری طراحی و پیاده‌سازی کنید که امکانات زیر را داشته باشد:

- فیلدهای مناسب برای ذخیره سازی دقیقه و ساعت
- یک سازنده که ورودی آن دقیقه و ساعت باشد
- یک سازنده که ورودی آن تنها ساعت باشد (در این حالت دقیقه باید صفر در نظر گرفته شود)
- یک متد برای تغییر دقیقه و ساعت
- دو متد برای برگرداندن مقدار دقیقه و ساعت
- یک متد که ورودی آن شیء دیگری از کلاس Time است و مانند تمرین اول دو زمان را با هم مقایسه می‌کند
- متدی که بر اساس زمان رشته ای برگرداند که بگوید چه وقتی از روز است. فرض کنید از ابتدای یک روز تا ساعت ۱۱:۵۹ صبح، ۱۲:۰۰ ظهر، از ۱۲:۰۱ تا ۱۶:۵۹ بعدازظهر، از ۱۷:۰۰ تا ۱۹:۵۹ عصر و بعد از آن شب محسوب می‌شود.

نکات:

- هر جا قرار است زمان ذخیره شده تغییر کند، باید بررسی شود که اعداد وارد شده قابل قبول باشد و در غیر این صورت اجرای برنامه خاتمه یابد.
  - برای آزمودن امکانات این کلاس یک برنامه ساده در تابع main بنویسید.
  - دقت کنید که فیلدهای کلاس نباید به صورت عمومی (public) قابل دسترسی باشند.
- کد ارسالی شما توسط کوئرا فقط کامپایل خواهد شد و امتیاز آن بعدا به صورت دستی محاسبه می‌شود.