نبرد حران

تمرین دوم برنامه نویسی پیشرفته

دانشکده مهندسی کامپیوتر دانشگاه شهید رجایی

مقدمه

حسام دیزی که پس از موفقیتی که در سفر به ایران کسب کرده بود(حل سوال دترمینان) به سمت اورشلیم بازگشت تا خبر پیروزی را به صلاح الدین بدهد اما فرمانروا به وسیله یاران با وفایی که داشت متوجه تقلب کردن حسام شده بود

لذا دستور داد که او را از مقامی که داشت برکنار کنند و تمام ثروت او را مصادره کرده و حسام دیزی را از سرزمین خود اخراج کردند. حسام که بسیار ناراحت بود به سرزمین رومیان به نزد کنستانتین لاسکاریس (امپراطور روم باستان) رفت و با او سوگند وفاداری بست. در آن زمان در دوره لاسکارید ها مارکوس کراسوس فرمانروای منطقه سوریه بدون هیچ اعلانی با هفت لشکر از لژیونر هایی که در اختیار داشت برای فتح سلوکیه لشکر کشی کرد. کراسوس از حسام خواست تا در این نبرد علیه اشکانیان به او کمک کند .
اما حسام تا آن زمان فقط تجربه جنگ آن هم پشت سیستم در بازی total war ***medieval*** 2 داشت.
حال امید حسام به دستان کد زن شماست تا بتواند به نبرد با ایران برود ولی چیزی در رابطه با نبرد به شما نمیگوید.

برای اطلاعات بیشتر درباره نبرد حران می توانید از لینک زیر استفاده نمایید.
نبرد حران

فرمانده

کلاس فرمانده را با ویژگی های زیر بنویسید.

فیلد ها :

نام فرمانده - لقب فرمانده - شماره شناسایی منحصر به فرد برای فرمانده

دقت کنید که همه ی فیلد ها برای فرمانده از جنس string می باشد و شماره شناسایی 10 رقم خواهد بود . چنان چه تعداد ارقام کمتر از 10 بود باید آنها را با 0 پر نمایید.

متد ها :

+ متد های Setter & Getter برای نام و لقب فرمانده
+ متد getter برای شماره فرمانده
+ تابع سازنده که هر سه فیلد ورودی را بگیرد.

تاریخ

از آنجایی که لشکر روم بسیار با نظم می باشد باید کلاسی موسوم به Date را پیاده سازی نمایید.

پیاده سازی این کلاس بسیار حیاتی است . دقت کنید که 6 ماه اول سال 31 روز و بقیه ماه ها 30 روزه هستند. همچنین تاریخ ها به شمسی می باشند.

فیلد ها :

روز - ماه - سال

متد ها :

+ تابع سازنده ورودی
+ Date (int year , int month , int day)
+ این تابع یک تاریخ می گیرد و مشخص میکند که آیا تاریخ فعلی از ورودی عقب تر است یا نه
+ Bool isBefore(Date date)
+ این تابع یک تاریخ می گیرد و مشخص میکند که آیا تاریخ فعلی از ورودی عقب تر یا با آن مساوی است یا نه
+ Boolean isBeforeOrSame(Date date)
+ این تابع یک تاریخ می گیرد و مشخص میکند که آیا تاریخ فعلی از ورودی جلو تر است یا نه
+ Boolean isAfter(Date date)
+ این تابع یک تاریخ می گیرد و مشخص میکند که آیا تاریخ فعلی از ورودی جلو تر یا با آن مساوی است یا نه
+ Bool isAfterOrSame(Date date)
+ دو تاریخ را باهم جمع می کند.
+ Date addDate(Date first , Date Second)
+ تاریخ بزرگتر را از کوچکتر کم میکند.(یه و خ تاریخ منفی ندید فرمانده خودکشی کنه!!!)
+ Date subtractDate(Date first , date second)
+ نمایش به فرمت YYYY/MM/DD
+ String toString()

دقت کنید که منظور از تاریخ فعلی شی ای است که تابع را برای آن صدا میزنید.

برای هر سه فیلد توابع Setter & Getter تعریف نمایید فقط دقت کنید اگر جاسوسی آمد و مقدار غیر قابل قبولی داد false برگردانده شود .

لژیونر

کلاس لژیونر یا سرباز را با ویژگی های زیر بنویسید.

فیلد ها :

نام سرباز - اسم مستعار سرباز - شماره شناسایی منحصر به فرد برای سرباز - تاریخ تولد از جنس کلاس Date - نوع لژیونر(سوارکار- کماندار - نیزه دار - شمشیر زن)

شماره شناسایی باید نه رقمی باشد و در صورت کمتر بودن صفر و در صورت بیشتر بودن فقط نه رقم ابتدایی در نظر گرفته شود.

متد ها :

+ متد های Setter & Getter برای همه فیلد ها
+ تابع سازنده با تمامی ورودی ها
+ تابع سازنده که همه ورودی برای نام و نام مستعار و تاریخ تولد بگیرد . شماره لژیونر باید به صورت یکتا و رندوم ایجاد شود.
+ برای محاسبه و برگرداندن سن تابعی پیاده سازی کنید . ورودی این تابع یک Date دیگر است که نشان زمان فعلی می باشد.
+ متد Equals که یک سرباز میگیرد و مشخص می کند آیا فیلد ها ی ورودی با فیلد های آبجکت برابر هستند یا نه ؟؟؟

دسته جنگی لژیونر ها

کلاس دسته جنگی را با ویژگی های زیر بنویسید.

فیلد ها :

فرمانده از جنس کلاس خود - ظرفیت سربازان - شماره شناسایی منحصر به فرد برای دسته جنگی - اسم -

محدودیت سنی از جنس کلاس Date (اگر این فیلد null نباشد نشان دهنده این هست که سربازان ای که ثبت نام می کنند نباید تاریخ تولدشان جلوتر از این فیلد باشد)

دقت کنید تعداد دسته های جنگی نباید از هفت تا بیشتر باشد و اعداد برای شماره شناسایی نیز باید تک رقمی باشند.

متد ها :

+ تابع سازنده سه ورودی : اسم لشکر و ظرفیت و محدودیت سنی
+ تابع سازنده دو ورودی : اسم لشکر و ظرفیت
+ تابع سازنده یک ورودی : اسم دسته جنگی و در این حالت ظرفیت رو به صورت پیشفرض روی 25 بزارید.
+ ثبت نام سرباز : یک سرباز در ورودی میگیرد و در صورتی که شرایط ثبت نام فراهم بود(محدودیت سنی-ظرفیت-و اینکه قبلا ثبت نام نکرده باشد) مقدار true را برمی گرداند و سرباز را اضافه می کند بدیهی است لیست لژیونر ها یک دسته جنگی را باید در جایی ذخیره کنید و در غیر این صورت False برمی گرداند.
+ تابعی برای افزایش ظرفیت . این تابع یک عدد می گیرد و به اندازه آن ظرفیت را زیاد می کند.
+ توابع Setter & Getter برای فرمانده
+ یک تابع get SoldireList بنویسید که یک آرایه از کلاس لژیونر برگرداند.

برنده نبرد

فیلد ها :

نام امپراطور - نام کشور - تعداد کل نیرو ها (مجموع سربازان دسته های جنگی) - روحیه سربازان

هر کشور شامل آرایه ای از دسته های جنگی می باشد.
دقت کنید روحیه یک عدد اعشاری کوچکتر از 1 و بزرگتر از 0 می باشد.

متد ها :

+ علاوه بر متد های معمولی یک تابع به اسم Winner باید بنویسید که یک کشور دیگر را در ورودی دریافت نماید و در انتها با بررسی تعداد سربازان و نیز روحیه ی سپاهیان هر کشور برنده نهایی نبرد را اعلام نماید. برای این منظور باید تعداد سربازان را در عدد مربوط به روحیه ضرب نماید و عدد حاصل را مقایسه کنید.

+ یک تابع بنویسید که تمامی اطلاعات (نام امپراطور - تعداد کل نیرو ها - میزان روحیه سربازان را در خروجی چاپ نماید)

توجه

در main

+ تعدادی لژیونر بسازید با تمامی حالت ها مختلف
+ چهار دسته نبرد جنگی بسازید (با هر سازنده یکی)
+ برای حالتی که فرمانده مشخص شده است از تابع با سه ورودی استفاده کنید و سپس با استفاده از Setter فرمانده را اضافه نمایید.
+ در انتها سرباز ها را اضافه و لیست آنها را در انتها چاپ کنید.
+ دو شی از کلاس برنده نبرد برای کشور روم و ایران بسازید و نتیجه نبرد را با استفاده از تابع Winner چاپ نمایید.

توجه نمایید که پاسخ را تنها در یک فایل cpp بایستی بارگذاری نمایید اما پیشنهاد می شود برای راحتی ابتدا هر کلاس را در هدر فایل مربوط به خود بنویسید سپس در انتها همه فایل ها را در یک فایل cpp مرج نمایید.

موفق باشید
آبان 1399