

تمرین شماره ۲

مکتب ۶۷



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تمرین اول : کلاسی به نام Rational بصورت زیر پیاده سازی کنید:

این کلاس، در حقیقت نشان دهنده اعداد کسری می‌باشد، کلاس شما باید سازنده ای داشته باشد که صورت و مخرج عدد اعشاری را دریافت نماید و آن را به ساده ترین شکل ممکن ذخیره کند.
برای مثال، در مورد کسر $2/4$ ، صورت ۲ و مخرج ۴ دریافت شود و عددی که در کلاس ذخیره می‌شود باید برابر $1/2$ باشد، یعنی صورت ۱ و مخرج ۲ باشد.

دقت کنید در حالتی که سازنده بالا صدا زده نشده باشد، مقدار کسر برابر صفر می‌باشد. (یعنی صورت برابر صفر و مخرج برابر ۱)

متدهایی برای هر کدام از اعمال زیر پیاده سازی کنید:

الف) جمع کسر فعلی با یک کسر دیگر با متد add

fun add (Rational num): Rational

ب) تفریق کسر فعلی از یک کسر دیگر با متد sub

fun sub (Rational num): Rational

ج) ضرب کسر فعلی با یک کسر دیگر با متد mul

fun mul (Rational num): Rational

د) تقسیم کسر فعلی بر یک کسر دیگر با متد div

fun div (Rational num): Rational

توجه کنید که در تمامی متدهای بالا نتایج در شیئی که متد بر روی آن صدا زده شده است، ذخیره نمی‌شوند بلکه باید در یک شی کسر جدید ذخیره شوند که توسط این متدها برگردانده می‌شود. برای مثال اگر اعداد کسری x و y را داشته باشیم و مقدار اولیه هر کدام از آنها $1/2$ باشد، عمل $x.add(y)$ بر روی مقادیر x و y تاثیری نخواهد داشت و مقادیر هر دوی آن‌ها $1/2$ باقی می‌ماند.

ه) یک عدد کسری را به فرمت اعشاری تبدیل کنید، با متد toFloatingPoint

fun toFloatingPoint(): Double

و) یک عدد کسری را به رشته متنی با قالب " a/b " تبدیل کنید که a صورت کسر و b مخرج کسر است.

fun toString(): String

ز) یک عدد کسری را به صورت یک عدد مخلوط در رشته متنی نمایش دهید. برای مثال عدد $\frac{7}{2}$ را به صورت رشته متنی " $1\frac{1}{2}+3$ " نمایش دهید.

`fun toMakhlotString(): String`

ضمناً باید بتوانید در یک متد `main` اشیای کسری تولید کرده و متدهای بالا را بر روی آن امتحان نمایید.

تمرین دوم : می‌خواهیم چند جز کوچک از سیستم بانکی را شبیه سازی کنیم. چهار کلاس برای شعبه، مشتری، حساب و کارت اعتباری با توضیحاتی که برای ویژگی هایشان در زیر آمده است، بسازید.

شعبه:

یک کلاس با نام `Branch` ایجاد کنید که سه ویژگی زیر را داشته باشد :

`code` : یک رشته از حروف و اعداد است که کد شعبه را نشان میدهد.

`city` : نام شهری که شعبه در آن قرار دارد را نشان میدهد.

`rate` : یک عدد صحیح از بین اعداد ۱ و ۲ و ۳ است که درجه ی شعبه را نشان میدهد.

حساب:

یک کلاس با نام `Account` ایجاد کنید که سه ویژگی زیر را داشته باشد :

شماره حساب `accountNumber` : رشته ای (`String`) از اعداد است که شماره حساب را نشان میدهد .

مانده حساب `balance` : یک عدد اعشاری (`double`) است که میزان مانده ی حساب را نشان میدهد .

شعبه افتتاح حساب `branche` : یک شی از جنس `Branch` است که شعبه ای که در آن حساب افتتاح شده است را نشان میدهد.

کارت اعتباری:

یک کلاس با نام CreditCard ایجاد کنید که دو ویژگی زیر را داشته باشد :

شماره کارت cardNumber : رشته ای (String) است از اعداد که شماره ی کارت را نشان میدهد .

اعتبار credit : یک عدد اعشاری (double) است که میزان اعتبار باقیمانده در کارت را نشان میدهد.

مشتری:

یک کلاس با نام Customer ایجاد کنید که سه ویژگی زیر را داشته باشد :

نام name : رشته ای از حروف است که نام مشتری را نشان میدهد .

حساب account : یک شی از جنس Account است که حساب مشتری را نشان میدهد.

کارت اعتباری creditCard : یک شی از جنس CreditCard است که کارت اعتباری مشتری را نشان میدهد .

بعد از این که کلاس ها و ویژگی هایشان را تعریف کردید، متدهای لازم برای مقداردهی و فراخوانی ویژگی ها (setter , getter) را نیز بنویسید .

یک متد با نام calCustomerBalance در کلاس مشتری ایجاد کنید که مجموع اعتبار باقیمانده در حساب و کارت اعتباری مشتری را محاسبه کرده و برمیگرداند.

در نهایت و در متد main ، ۲۰ شی از جنس Customer ساخته و با مقادیر دلخواه فیلد های آنها را مقدار دهی نمایید. بدیهی است برای مقدار دهی فیلد های کلاس Customer باید شی هایی از کلاس های Branch (۵ شعبه ساخته شود) ، Account (به ازای هر مشتری یک اکانت) و CreditCard (به ازای هر مشتری یک کارت اعتباری) نیز ساخته شود. اشیای ساخته شده از جنس Customer را در یک آرایه ذخیره نمایید.

سپس با فراخوانی متد calCustomerBalance مجموع اعتبار باقیمانده در حساب و کارت اعتباری هر یک از مشتری ها را چاپ نمایید.

همچنین مجموع اعتبار باقیمانده مشتریان هر شعبه را نیز به تفکیک شعبه چاپ نمایید.

نکات :

در هر کلاس، ویژگی ها را در صورت امکان به صورت خصوصی (private) و متدها را به صورت عمومی (public) تعریف کنید . در صورت لزوم برای ویژگی ها دو متد setter و getter را تعریف کنید .

در این سوال فرض شده که کارت اعتباری و حساب برای مشتری جدا از هم هستند یعنی میزان باقیمانده ی آنها ربطی به یکدیگر ندارد.

تمرین سوم : سوال لینک زیر را حل کنید.

<https://quera.ir/problemset/617/>

تمرین چهارم: سوال لینک زیر را حل کنید.

<https://quera.ir/problemset/49028/>

تمرین امتیازی: سوال لینک زیر را حل کنید.

<https://quera.ir/problemset/31021/>

مهلت ارسال تمرین تا ساعت ۵۹ : ۲۳ چهارشنبه ۱۴۰۰/۰۹/۱۷ می باشد. پاسخ تمارین را در کارتابل آموزش خود ارسال کنید.

- به ازای هر روز تاخیر، ۵ درصد نمره کسر خواهد شد.
- در صورت لزوم یک فایل word به عنوان توضیح در کنار کدهای خود قرار دهید.
- نام فایل خود را به این صورت قرار دهید، Name_HW2_Maktab67 به عنوان مثال :
Mohammadreza_Ghaderi_HW2_Maktab67
- در صورتیکه تمرین شامل چند فایل و فولدر می باشد حتما آنها را در قالب یک فایل zip ارسال نمایید.
- تمیزی فایلها و کدنویسی نیز بخشی از نمره را تشکیل میدهد.
- در صورتیکه سوالی دارید در گروه تلگرام بپرسید.