

تابع خفن

موردی که انتظار می‌رود در این سوال استفاده کنید: `function overloading`

توی این سوال قراره کار های جالبی انجام بدیم که کمکی به ++C کرده باشیم!

فرض کنید تابعی دارید که میتونید برای هر نوع متغیری اون رو صدا بزنید!

خب بریم سراغ خود تابع:

فرض کنید تابع دو ورودی میگیره که هرکدام از این ورودی ها میتونن حالت های زیر را داشته باشند:

- عدد فقط از جنس های `float` و `int`

- آرایه فقط از جنس های `float` و `int`

- رشته

- بولین (`bool`)

خروجی تابع به ازای ورودی های مختلف فرق داره و:

۱. اگر ورودی ها رشته باشند باید دو ورودی به یکدیگر چسبانده شوند و به صورت رشته `return` شود.

۲. اگر ورودی ها بولین باشند باید دو ورودی را باهم `and` کرده و `return` کنید.

۳. اگر ورودی ها عدد باشند باید دو عدد را جمع کنید و `return` کنید.

۴. اگر ورودی ها دو آرایه هم جنس باشند می بایست دو آرایه با یکدیگر ادغام شده و آرایه حاصل `return` شود.

۵. اگر ورودی ها دو آرایه غیر هم جنس باشند می بایست پیام خطای "`Different Types`" چاپ شود.

پردازش اعداد کسری

برنامه ای بنویسید که شامل کلاسی به نام Rational است. اشیا این کلاس اعداد کسری را نمایش می دهند. به این صورت که کلاس یک عضو داده خصوصی به نام a از نوع int برای نگهداری صورت و یک عضو داده خصوصی به نام b از نوع int برای نگهداری مخرج دارد. هدف نهایی انجام عمل تفریق بر روی اعداد کسری است، برای این منظور باید از کاربر دو ورودی از جنس Rational دریافت کنیم. به طور مثال عدد کسری سه چهارم به صورت $3/4$ وارد می شود. عدد کسری یک دهم به صورت $1/10$ وارد می شود.

هدف نهایی انجام عمل تفریق بر روی اعداد کسری است. به این منظور برنامه ای که می نویسید باید دارای تابعی به نام Subtract باشد که عضو کلاس Rational است. این تابع عملیات تفریق بر روی دو شی را انجام می دهد و نتیجه را به صورت یک کسر ساده شده برمی گرداند.

نکات مهم:

۱. تابع Subtract خروجی را چاپ نمی کند. فقط آن را بر می گرداند.
 ۲. برنامه شما باید یک تابع سازنده داشته باشد که ورودی آن صورت و مخرج کسر است.
 ۳. برنامه ی شما باید شامل توابع getter و setter برای صورت و مخرج کسر باشد.
- در نهایت برنامه باید حاصل تفریق را در خروجی چاپ کند (کسر به صورت ساده شده نمایش داده شود).

مثال

ورودی نمونه

2/16

3/4

خروجی نمونه

زمان شی است

هدف

طراحی کلاسی به نام Time برای ذخیره کردن زمان است.

توابع جهت پیاده سازی

- فیلد های مناسب برای ذخیره سازی ساعت و دقیقه انتخاب کنید
- 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)
- متد های getter برای ساعت و دقیقه
- متد های setter برای ساعت و دقیقه
- یک متد برای مقایسه که ورودی آن از جنس کلاس Time است
- متدی که بر اساس زمان رشته ای برگرداند که بگوید چه وقتی از روز است.(از ابتدای یک روز تا ساعت 11:59 صبح، 12:00 ظهر، از 12:01 تا 16:59 بعد از ظهر، از 17:00 تا 19:59 عصر و بعد از آن شب محسوب می شود)

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.