مثلث قانونی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

ما کلاسی تحت عنوان IllegalTriangleException داریم که زمانی که یک مثلث غیر قانونی داشته باشیم پرتاب میشه.یک مثلث زمانی غیر قانونیه که اضلاعش توی قضیه شرط وجود مثلث صدق نکنه.حالا از شما میخوایم یه برنامه بنویسین که ۳ ضلع یک مثلثو بگیره (اعداد صحیح) و اگه نمیشد با اضلاعش مثلث ساخت یک اکسپشن از نوع گفته شده رو پرتاب کنه که توش پیام "You can't make triangle!" نوشته شده باشه.

ورودی

ورودی نمونه

```
2 1 3
```

خروجی نمونه

```
You can't make triangle!
```

ماشین‌حسابِ پیش‌رفته

مهرشاد برای درس ریاضی خود نیاز به یک ماشین حساب دارد. با توجه به اینکه مغازه‌های ماشین‌حساب‌فروشی تعطیل هستند از شما می‌خواهد که:

برنامه ماشین‌حسابی بنویسید که دارای اعمال **جمع** و **تفریق** و **تقسیم** و **ضرب** و **جذر** باشد.

ضمنا برای هر عملگر، متدی داشته باشد مثلا یک متد کار جمع را انجام دهد و غیره..

برنامه در ابتدای شروع از کاربر بپرسد کدام‌یک از عملیات‌ها را می‌خواهد انجام دهد و سپس بر اساس آن، متد متناسب را فراخوانی کند و از کاربر عدد بگیرد.

نکته

دقت کنید که می‌دانیم **تقسیم بر صفر** و **جذر عدد منفی** در بازه اعداد حقیقی معنی ندارد پس در متدی که این عملیات‌ها را انجام می‌دهد، استثنا مربوطه را مدیریت کنید.

- برای **تقسیم بر صفر** از **کلاس پیشفرض خود جاوا** استفاده کنید
- برای جذر منفی خودتان **کلاس استثنا ایجاد کنید ،**
- همچنین که اگر کاربر به جای عددها **رشته غیر عددی** وارد کند، باید استثنا مربوطه را پرتاب کنید.
- برنامه شما باید با عدد صحیح کار کند.
- تصحیح این سوال به صورت دستی است.

سودوکو

حتماً با بازیِ سودوکو آشنا هستید. در این سؤال قرار است یک سودوکو را **به کمکِ Exceptionها** پیاده‌سازی کنیم.

در ابتدا جدولِ بازی خالی است. در هر مرحله، یک مختصات + عددِ موردِ نظرِ کاربر را از ورودی دریافت کنید. همچنین در صورتی که دستورِ showChart وارد شده بود، جدول را نمایش دهید.

سه Exception را برای وقتی که که عددِ ورودی قبلاً در سطر، ستون و مربّعِ خانه‌ی انتخاب‌شده وجود داشته باشد تعریف کنید (و از آن‌ها استفاده کنید!).

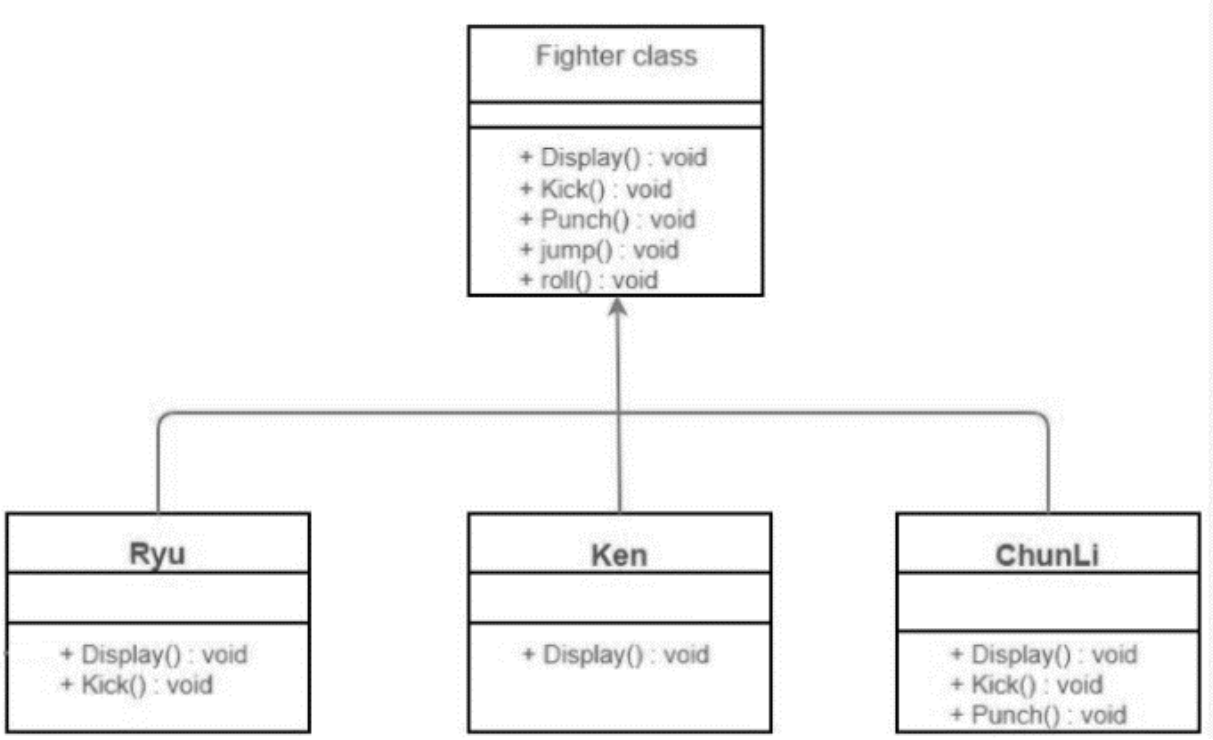
همچنین Exceptionهای دیگر را در موقعیّت‌های دیگر (مثلِ محدوده‌ی عدد، محدوده‌ی جدول، ورودیِ اشتباه و..) با Exceptionهای مربوطه مدیریّت کنید.

هر وقت بازی تمام شد، با نشان دادنِ پیامِ مناسب از برنامه خارج شوید.

دیزاین پترن

تارا برای اینکه از روزهای قرنطینه‌ی خود بهتر استفاده کند، درحال ساخت یک بازی به نام **Black Warrior** است. در نسخه ی ابتدایی بازی ، یک جنگنده چهار حرکت **kick** و **punch** و **jump** و **roll** را دارد. تارا ابتدا از ارث بری استفاده کرده و یک کلاس **Fighter** دارد که شخصیت‌های دیگر از ان ارث بری می کنند.

هر حرکت می تواند در کلاس یکی از شخصیت های بازی مجدد **override** شود و عملکرد خاص‌تری داشته باشد.



بعد از مدتی، تارا متوجه می شود که برخی شخصیت‌ها نمیتوانند **بلند بپرند** ، یا برخی قابلیت **roll** را ندارند ، و برای **kick** سه حالت ضعیف، متوسط و شدید وجود دارد. این تغییرات موجب تولید **کد تکراری فراوان** میشوند. لذا مجبور است کد خود را کاملا تغییر دهد و از یک **دیزاین پترن** مناسب (استراتژی) برای آن استفاده کند.

کاری که شما برای کمک به تارا باید انجام دهید به شرح زیر است:

کلاس اصلی Fighter را مجدد طراحی و ویژگی یا متد های آن را در اینترفیس یا کلاس های دیگر به گونه ای پیاده‌سازی کنید که مشکلات زیر رفع شوند :

- همانطور که گفته شد برای **kick**، **سه** حالت وجود دارد.
- یک شخصیت یا میتواند بلند بپرد، یا کوتاه.
- همه ی شخصیت ها شامل متد **display** و **punch** میشوند.
- ممکن است شخصیتی قابلیت **roll** را داشته باشد یا نداشته باشد.

توجه کنید تمامی متدهایی که تعریف میکنید باید **void** باشند، و بدنه‌ی آن‌ها فقط شامل دستور چاپ نام کلاس آن متد باشد؛ مثلا اگر اینترفیس Roll را دو کلاس CantRoll و CanRoll پیاده سازی میکنند برای کلاس CantRoll داریم :

```
1 void roll(){
2     System.out.println("cantRoll");
3 }
```

توجه کنید نیازی به پیاده سازی شخصیت ها نیست.

testsocket