بازگشت جان‌ویک

جان ویک به تازگی شروع به پذیرش قراردادهای جدید کرده. برای اینکار نرم افزاری روی تلفنش نصب کرده که مشتریان از طریق آن قرارداد های خود را برای او می‌فرستند. هر قرارداد به فرم یک شی در جاوا وجود دارد. برای انتقال آن از طریق اینترنت باید آن را به یک رشته تبدیل کنیم و سپس در تلفن جان ویک این رشته دوباره به یک رشته جاوا تبدیل می‌شود.

حال شما باید دو تابع serialize و deserialize را پیاده‌سازی کنید که این اعمال را انجام دهد. تابع serialize به شکل زیر است که یک شی دریافت کرده و یک رشته را خروجی می‌دهد.

```
1 | public static String serialize(Object obj)
```

تابع deserialize در ورودی خود رشته و نام کلاسی که این شی به آن تعلق دارد را می‌گیرد و باید کلاس ساخته شده را بازگرداند.

```
1 | public static Object deserialize(String content, String fullClassName)
```

توجه کنید که یک شی ممکن است datatype های اصلی (مثل int و string و double) و همچنین آرایه (ArrayList) و حتی یک شی دیگر را در خود داشته باشد. البته تضمین می‌شود که شی‌های تو در تو تنها تا یک لایه در هم قرار میگیرند.

همچنین باید مواردی مثل ارث بردن را نیز مد نظر قرار دهید.

رشته را به فرم json و در یک خط (بدون space و nextline) بسازید. همچنین داخل رشته فیلدها به ترتیب حروف الفبا آمده باشند.

چیزی که باید آپلود کنید

یک فایل زیپ آپلود کنید با محتوای یک فایل به اسم `YourConvertor.java` که باید در آن فایل زیر را تکمیل کنید:

```java
public class YourConvertor{
    public Object deserialize(String input, String className) {
        //TODO Implement this method
        return null;
    }

    public String serialize(Object input) {
        //TODO Implement this method
        return null;
    }
}
```

فرار از کانتیننتال

جان ویک متوجه شده است که کانتیننتال دیگر جای امنی برای ماندن نیست و در تلاش است هرچه زودتر وسایلش را جمع کند و بزند به چاک! اما برای این کار نیاز دارد وسایلش را تا جای ممکن فشرده‌سازی کند تا در هنگام فرار برای او مشکل ایجاد نکنند.

او تصمیم گرفت که تنها وسایل ضروری را با خود ببرد که به شکل Object هایی با تعدادی فیلد primitive (شامل اعداد صحیح، اعشاری و رشته‌ها) و بدون متد هستند. برای فشرده‌سازی این Object ها باید هرکدام را به صورت یک رشته در بیاورد که نام هر فیلد و مقدار آن به صورت *key:value* آورده شده است. برای مثال اگر آبجکت Obj شامل یک متغیر int با نام anInt و مقدار *36* و یک رشته با نام aString و مقدار *such is life* باشد، فشرده شده آن به صورت زیر میشود:

```
"anInt:36,aString:such is life"
```

توجه کنید فیلدها در رشته ساخته شده با علامت , و بدون هیچ فاصله ای از هم جدا می شوند. همچنین **فیلدها باید به ترتیب الفبایی در رشته خروجی سورت شده باشند**.

جان ویک برای این که در صورت لو رفتن، دشمنانش نتوانند وسایلش را رمزگشایی کنند بالای تعدادی از فیلدها از انوتیشن `@rename(key = "some name")` استفاده کرده است. در صورتی که بالای هر فیلدی این را دیدید، کافی است در رشته خروجی نام آن فیلد را به عبارتی که داخل key آمده است تغییر دهید. اگر هم key وجود نداشت و انوتیشن خالی بود نیازی به تغییر نام نیست. برای درک بهتر این بخش توصیه میشود یونیت تست آخر را بررسی کنید.

چیزی که باید آپلود کنید

باید فایل زیپی آپلود کنید که دو فایل `Rename.java` و `Serializer.java` را داشته باشد. محتوای فایل `Rename.java` باید به صورت زیر باشد:

```java
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
```

```java
import java.lang.annotation.Target;

@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
public @interface Rename {
    public String name() default "";
}
```

محتوای فایل `Serializer.java` هم باید به صورت زیر باشد:

```java

public class Serializer {
    public String getSerializedString() throws IllegalAccessException {
        // TODO: FILL ME!
    }

    public Object getObject() {
        // TODO: FILL ME!
    }

    public void setObject(Object object) {
        // TODO: FILL ME!
    }
}
```