

Apache commons یک پروژه ای از آپاچی می باشد که بر روی کدهای تکراری که برنامه نویس می نویسد تمرکز کرده است تا از تکرار این کارها جلوگیری و به نحوی سرعت برنامه نویس را افزایش دهد.

Apache commons از سه قسمت تشکیل شده است:

Proper commons : یک ریپازیتوری (مخزن) از قسمت هایی از کار برنامه نویس می باشد که مدام در حال تکرار آن ها می باشد.

sandbox commons : یک فضای کار برای توسعه دهندگان مولفه های جاوا.

Dormant commons : یک ریپازیتوری از مولفه های جاوا که در حال حاضر غیرفعال اند.

Proper commons به یک هدف اصلی اختصاص یافته است: ساخت و نگهداری مولفه های قابل استفاده ی مجدد در جاوا. Proper common فضایی برای همکاری و اشتراک گذاری می باشد تا همه ی توسعه دهندگان جامعه ی آپاچی بتوانند با یکدیگر روی پروژه ای که توسط آپاچی به اشتراک گذاشته شده است کار کنند. این توسعه دهندگان تلاش میکنند که مولفه های آنها حداقل وابستگی به سایر کتابخانه ها را داشته باشد. در نتیجه این مولفه ها میتوانند به راحتی توسط کاربر پیاده سازی شوند و در نتیجه کاربران آپاچی به راحتی و بدون هیچ نگرانی درباره ی تغییر آنها در آینده میتوانند اجزای این common را پیاده سازی کنند.

sandbox commons: پروژه های این commons حاوی فضاهای کاری اند که برای همه ی commit کنندگان قابل مشاهده است.

گوگل guava مجموعه ای از کتابخانه های openSource می باشد که توسط گوگل توسعه یافته است. این کتابخانه ها به کم شدن ارور در هنگام کدنویسی کمک میکنند. این کتابخانه ها ابزار مفیدی برای کالکشن ها ، ذخیره سازی، پشتیبانی از primitive ها ، annotation گذاری های رایج و... می باشند.

چند تا از دلایلی که، guava نسبت به سایر کتابخانه ها در جاوا برتر دانسته می شود:

- کاهش یافتن مقدار کدهایی که باید نگهداری شوند.
- بسیاری از کلاس های ابزاری مانند iterable ها، lists، sets، maps، multisets، multimap و tables ... که بسیار زیاد برای برنامه نویس قابل استفاده می باشد.
- پیاده سازی متدهای Object مانند hashCode() و toString() ساده سازی شده است.
- کتابخانه guava بهینه سازی شده است.
- مجموعه های قدرتمندی از ابزارهای collection فراهم میکند که در java.util.Collection موجود نیست.

چند نمونه از متدهای کلاس String

- متد concat رشته ای را به عنوان ورودی از کاربر دریافت می کند و آن را به انتهای رشته ی فعلی الحاق می کند. سپس نتیجه را در یک رشته ی جدید return میکند.
- متد toUpperCase همه ی عنصرهای رشته را به حروف بزرگ لاتین تبدیل میکند و همچنین متد toLowerCase رشته ای جدید با حروف کوچک لاتین، return می کند.
- متد compareTo یک رشته را به عنوان ورودی از کاربر میگیرد و اگر با رشته ی کنونی برابر بود مقدار true و در غیر اینصورت مقدار false برمی گرداند.
- متد replaceAll دو رشته را از کاربر میگیرد و هر کجا در رشته ی کنونی رشته ی اولی تکرار شده باشد، آنرا با رشته ی دومی جایگزین می کند.
- متد split رشته را به عنوان ورودی می گیرد و هرکجا که آن رشته در رشته ی کنونی واقع شده باشد؛ رشته را تکه میکند و در نهایت آرایه ای از رشته ها را return میکند.
- متد substring دو متد overload شده دارد. در متد اولی یک index به آن می دهیم و رشته ی جدیدی با شروع از index تا انتهای رشته کنونی را برای ما return میکند. در متد دومی دو index به آن میدهیم و یک رشته ی جدید از index1 تا (index2-1) را برای ما return می کند.
- متد trim تمام whiteSpace های قبل و بعد از رشته را ignore میکند و یک رشته ی جدید بدون whiteSpace در قبل و یا بعد از رشته، return میکند.

منابع:

[www.commons.apache.org](http://www.commons.apache.org)

[www.geeksforgeeks.com](http://www.geeksforgeeks.com)

