بنام او

پروژه ابتدایی (Redis)

مقدمه

این تمرین به منظور آشنایی اولیه دانشجویان با یک سامانه ساده مدیریت پایگاه داده طرح شده. در این تمرین با استفاده از زبان **جاوا**، برنامه‌ای خواهید نوشت که به یک سامانه مدیریت داده به نام Redis متصل می‌شود و با استفاده از آن، قابلیت ذخیره و بازیابی اطلاعات را خواهد داشت. این برنامه در ابتدای کار خود برای اولین بار داده‌های موجود در یک فایل CSV که در پیوست آمده را بارگذاری کرده و در ادامه، از طریق CLI دستورات را با فرمت مشخص شده ورودی گرفته و تغییرات لازم رو روی محتوای پایگاه داده ایجاد کند.

توجه داشته باشید که ساختار ورودی‌ها و خروجی‌ها برنامه **دقیقاً** مطابق آنچه در ادامه گفته خواهد شد باشد. تست‌کیس‌های این برنامه توسط Script اجرا و تست خواهد شد و درصورتی که جواب‌ها دقیقا مطابق آنچه خواسته شده نباشد، Script صحت آن را تایید نمی‌کند و نمره‌ای به برنامه تعلق نمی‌گیرد.

Redis

Redis یک سامانه مدیریت داده است. این سامانه یک مخزن ساختمان داده در حافظه است. به این صورت که داده‌ها را در حافظه اصلی (RAM) ذخیره می‌کند تا بتواند علاوه بر سرعت بالا خواندن و نوشتن، از قابلیت‌های مدیریت داده همانند یک پایگاه داده استفاده کند؛ همانطور که این در خیلی از موارد به عنوان یک پایگاه داده موقت استفاده می‌شود. این گونه پایگاه داده‌ها که داده‌های خود را در حافظه اصلی نگهداری می‌کنند، با توجه به سرعت بالای آن‌ها، در ایجاد cache برای جلوگیری از دسترسی‌های پیاپی به پایگاه داده اصلی برای دریافت داده یکسان استفاده می‌شود.

معماری کلید-مقدار در سامانه‌های مدیریت پایگاه داده، بسیار شبیه به ساختمان داده‌های HashMap، Dictionary و Object به ترتیب در زبان‌های Java، C# (و Python) و JavaScript است. به این صورت که دسترسی به مقادیر از طریق کلید آن‌ها انجام می‌شود و هر مقدار، برای ذخیره شدن در سامانه نیاز به یک کلید دارد که نماینده آن مقدار باشد.

این سامانه یک سامانه متن باز است و کد آن در گیت‌هاب قابل دسترسی است. برای استفاده از این سامانه ابتدا نیاز است Redis را نصب کنید. شما می‌توانید برای نصب Redis روی ویندوز از اینجا و روی لینوکس از اینجا کمک بگیرید. در ادامه برای اتصال برنامه خود به Redis نیاز به یک Client از Redis دارید. لیست

Clientهای Redis برای زبان جاوا را می‌توانید از اینجا دریافت کنید. از اینجا نیز می‌توانید یکی از Clientهای معروف Redis برای جاوا را برای Package Manager مورد استفاده خود دریافت کنید یا کتابخانه را به صورت خام (jar.) دانلود کرده و به پروژه خود اضافه کنید (لینک دانلود بالا صفحه سمت راست قرار دارد). در اینجا روش اضافه کردن کتابخانه ”jar.“ به پروژه در IntellJ قابل مشاهده است.

داده‌ها

داده‌های این پروژه به صورت CSV در پیوست در اختیار شما قرار گرفته است. ستون اول این داده‌های کلید آن‌هاست که باید از آن برای ذخیره سازی داده استفاده کنید و ستون دوم، مقدار مربوط به کلید است. در ابتدای شروع برنامه باید این داده‌ها را از روی فایل خوانده و به پایگاه داده اضافه کنید.

ورودی و خروجی‌ها

برنامه بعد از آن که اجرا شد باید منتظر ورودی کاربر باشد و جواب آن را چاپ کند و بعد از چاپ جواب، آماده دریافت ورودی بعدی باشد.

کاربر ورودی خود را به صورت یک خط دستور نوشته و دکمه Enter را می‌فشارد. سپس برنامه خروجی(ها)ی مطلوب را در ادامه چاپ می‌کند (هر جواب در یک خط) و در نهایت بعد از آخرین جواب، یک خط خالی چاپ شده و برنامه آماده گرفتن ورودی بعدی می‌شود.

این برنامه به عنوان ورودی 4 دستور اصلی زیر را می‌پذیرد:

- Create
- Fetch
- Update
- Delete

هر دستور یک نتیجه عملکرد دارد که می‌تواند true یا false باشد که در خط اول جواب آن چاپ می‌شود.

در ادامه به شرح هر کدام از دستورها می‌پردازیم.

دستور Create

این دستور وظیفه ایجاد یک داده جدید در پایگاه داده را دارد. به این صورت که این دستور به عنوان ورودی 2 متغیر دریافت کرده (یکی کلید و دیگری مقدار) و مقدار داده شده را با کلید داده شده، به عنوان داده‌ی جدید به پایگاه داده اضافه می‌کند.

توجه داشته باشید که در صورتی که کلید داده شده برای ایجاد یک داده در پایگاه داده وجود داشت، این دستور نباید انجام شود و نباید تغییری روی پایگاه داده ایجاد شود. در غیر این صورت داده جدید با کلید مورد نظر ایجاد شده و نتیجه کار مثبت خواهد بود.

create <key> <value>

دستور بالا ساختار یک دستور Create را نشان می‌دهد. برای مثال:

create a 1

create b 2

دستور Create به عنوان نتیجه، موفق بودن یا نبودن عملیات مورد نظر نمایش می‌دهد. برای مثال:

```
create a 1
true

create b 2
true

create a 3
false
```

به این صورت که در دستور اول یک داده با کلید a ساخته می‌شود و نتیجه مثبت است؛ در دستور دوم داده با کلید b ساخته می‌شود و نتیجه مثبت است؛ در دستور سوم از آنجا که کلید a قبلا در پایگاه داده وجود داشته، دستور انجام نمی‌شود و نتیجه منفی است. توجه داشته باشید خروجی این دستور باید دقیقا به همین صورت باشد.

دستور Fetch

این دستور وظیفه برگرداندن مقدار یک داده در پایگاه داده را دارد. به این صورت که کلید یک داده را به عنوان ورودی دریافت کرده و مقدار داده با آن کلید را باز می‌گرداند.

توجه داشته باشید که درصورتی که کلید داده شده به دستور، با هیچ کدام از کلیدهای داده‌های پایگاه داده تطابق نداشته باشد، نتیجه این دستور منفی و در صورت تطابق، نتیجه مثبت است.

fetch <key>

دستور بالا ساختار یک دستور Fetch را نشان می‌دهد. برای مثال:

fetch a

fetch b

دستورهای بالا به ترتیب مقادیر موجود در داده‌هایی با کلیدهای a و b را برمی‌گرداند.

این دستور به عنوان خروجی در خط اول، نتیجه موفق بودن یا نبودن عملیات را نشان می‌دهد سپس مقدار داده، با کلید مشخص شده را (در صورت وجود) نمایش می‌دهد. برای مثال:

```
create a 1234
true

create b 5678
true

fetch a
true
1234

fetch b
true
5678

fetch c
false
```

طبق تصویر بالا، ابتدا مقدار 1234 را با کلید a و مقدار 5678 را با کلید b وارد پایگاه داده می‌کنیم. سپس دستور گرفتن مقدار داده با کلید a را اجرا می‌کنیم؛ در جواب ابتدا نتیجه موفقیت این دستور نمایش داده می‌شود و سپس مقدار آن. دستور چهارم برای گرفتن مقدار داده با کلید b هم به همین صورت. در دستور پنجم، تلاش کردیم تا مقدار داده با کلیدی که وجود ندارد را بگیریم که نتیجه موفقیت این دستور منفی بوده و جوابی برنمی‌گرداند.

دستور Update

این دستور مقدار یک داده که از قبل در پایگاه داده وجود دارد را تغییر می‌دهد. به این صورت که کلید یک داده و مقدار جدید آن را دریافت می‌کند و مقدار داده با کلید داده‌شده را به مقدار داده‌شده تغییر می‌دهد.

توجه داشته باشید که درصورتی که کلید داده شده به دستور، با هیچ کدام از کلیدهای داده‌های پایگاه داده تطابق نداشته باشد، نتیجه این دستور منفی و در صورت تطابق، نتیجه مثبت است.

update <key> <value>

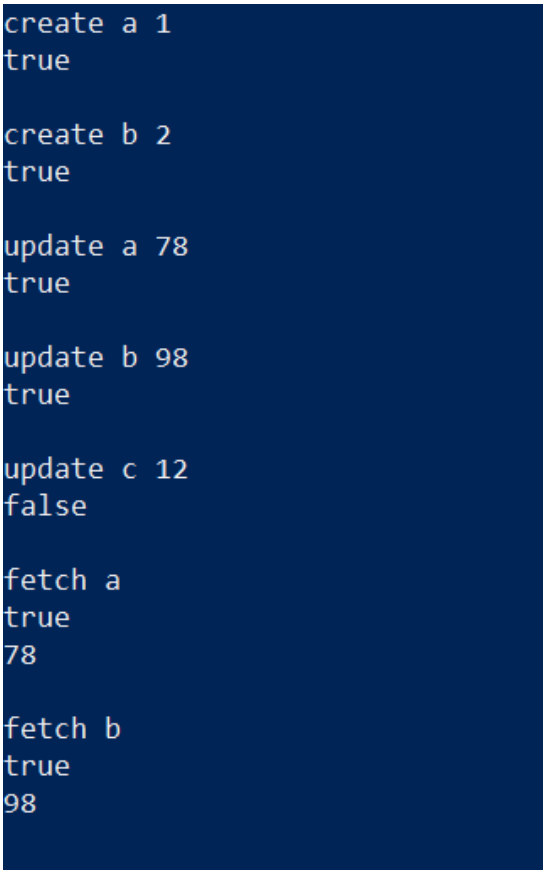
دستور بالا ساختار یک دستورUpdate را نشان می‌دهد. برای مثال:

update a 5

update b 12

دستورهای بالا به ترتیب مقدار داده‌ای را که کلید آن a است (و از قبل وجود دارد) را به 5 و مقدار داده‌ای را که کلید آن b است (و از قبل وجود دارد) را به 12 تغییر می‌دهد.

دستور Update به عنوان نتیجه، موفق بودن یا نبودن عملیات مورد نظر را نمایش می‌دهد. برای مثال:



طبق تصویر بالا، ابتدا داده‌ها با کلیدهای a و b را ایجاد می‌کنیم. سپس در دستور 3 و 4 آن‌ها را تغییر می‌دهیم. در دستور 5 سعی می‌کنیم مقدار داده با کلید c که وجود ندارد را تغییر دهیم؛ بنابراین این دستور با خطا مواجه می‌شود. در دستور 6 و 7 مقادیر داده‌ها با کلیدهای a و b را چاپ می‌کنیم و می‌بینیم این مقادیر نسبت به مقادیر اولیه‌ای که به آن‌ها دادیم متفاوت است.

دستور Delete

این دستور یک داده را از پایگاه داده پاک می‌کند. به این صورت که کلید یک داده را به صورت ورودی دریافت کرده و داده با کلید داده شده را پاک می‌کند.

توجه داشته باشید که درصورتی که کلید داده شده به دستور، با هیچ کدام از کلیدهای داده‌های پایگاه داده تطابق نداشته باشد، نتیجه این دستور منفی و در صورت تطابق، نتیجه مثبت است.

delete <key>

دستور بالا ساختار یک دستور Delete را نشان می‌دهد. برای مثال:

delete a

delete b

دستورهای بالا به ترتیب داده‌ها با کلیدهای a و b را از پایگاه داده پاک می‌کند.

دستور Delete به عنوان نتیجه، موفق بودن یا نبودن عملیات مورد نظر را نمایش می‌دهد. برای مثال:

```
create a 5
true

create b 17
true

fetch a
true
5

fetch b
true
17

delete a
true

delete c
false

fetch a
false

fetch b
true
17
```

همانطور که در تصویر مشخص است، در 2 دستور اول دو داده ایجاد کردیم. در دستور 3 و 4 مقادیر داده‌ها را از پایگاه داده گرفتیم. در دستور 5 داده با کلید a را پاک کردیم و در دستور 6 داده‌ای را پاک کردیم که کلید آن در پایگاه داده وجود ندارد پس با خطا مواجه می‌شویم. در دستور 7 سعی کردیم مقدار داده با کلید a را بگیریم اما از آنجا که قبلا آن را پاک کرده‌ایم، با خطا مواجه می‌شویم و در دستور آخر هم مقدار داده با کلید b را گرفتیم.

ساختار کد

برنامه شما باید دارای یک کلاس با نام Main باشد که به صورت public تعریف شده و تابع شروع برنامه شما (main) در آن قرار داشته باشد. توجه داشته باشید که در صورت عدم رعایت این نکات، برنامه‌ای که کد شما را تست می‌کند نمی‌تواند کد شما را اجرا کند و در نتیجه نمره‌ای به شما تعلق نمی‌گیرد.

نکات تمرین

- نام دستورات با حروف کوچک به عنوان ورودی داده می‌شود. مثلا *fetch a* درست و *Fetch a* نادرست است.
- استفاده از Redis برای این تمرین اجباری است و کد شما می‌بایست داده‌ها را در این پایگاه داده ذخیره کند.
- این تمرین باید به صورت انفرادی انجام شود و کدها برای تست تقلب توسط سیستم Moss چک خواهند شد.
- کدهای نوشته شده توسط شما به صورت دستی نیز بررسی خواهند شد. پس حتما تمام نکات اصلی از جمله استفاده از Redis را رعایت کنید.
- کدهایی که توسط شما آپلود می‌شود باید در یک فایل zip (zip != rar) باشد که داخل آن محتوای پوشه‌ی پروژه شما به صورت کامل باشد. این محتوا شامل داده‌های CSV داده‌شده و کتابخانه‌های اضافه شده به پروژه نیز می‌باشد. با گفتاری دیگر، پوشه آپلود شده توسط شما باید برای اجرای پروژه کافی باشد. این را با فرض نصب بودن Redis روی ماشین اجرا کننده کد شما در نظر بگیرید.
- نام فایلی که توسط شما آپلود می‌شود باید در قالب proj0_<student_number>.zip باشد. مانند: proj0_9531095.zip
- همانطور گفته شد، کد شما توسط برنامه اجرا و تست خواهد شد. هر گونه عدم تطابق نام فایل آپلود شده، نام کلاس Main، نحوه ورودی گرفتن و خروجی دادن و هر گونه تغییری که پیش‌تر به آن‌ها اشاره شد، منجر به عدم تشخیص یا تشخیص اشتباه کد شما شده و ممکن است نمره شما را کم کند.
- در صورت داشتن هر سوال یا هر گونه ابهام، با ایمیل samimd.77@gmail.com یا اکانت تلگرام @SamiMD ارتباط برقرار کنید.