



دانشگاه بوعلی سینا

درس برنامه سازی پیشرفته

پیاده سازی Key-Value Store

زمستان ۹۷

شرح کلی

معمولا در طراحی یک نرم افزار لازم است داده هایی را به صورت دائمی ذخیره کنیم. برای ذخیره سازی داده های یک برنامه از پایگاه داده یا Database استفاده می شود. پایگاه داده دارای انواع مختلفی است که هریک دارای کاربرد و روش استفاده ی بخصوصی هستند. دسته ای از Database ها مانند Redis روشی ساده برای ذخیره سازی داده ها در اختیار کاربر قرار می دهند. در این دیتابیس ها هر داده دارای یک کلید یا Key و یک مقدار متناظر با آن کلید یعنی Value است. به این نوع پایگاه داده Key-Value Store هم می گویند. کاربر با استفاده از دستور set مقداری را به یک کلید نسبت می دهد. همچنین با استفاده از دستور get می تواند مقدار یک کلید را دریافت کند. برای واضح تر شدن موضوع به مثال زیر دقت کنید:

```
> SET (course) (Advanced Programming)
```

```
> SET (university) (BASU)
```

در این مثال کاربر به کلید های course و university به ترتیب مقادیر Advanced Programming و BASU را نسبت می دهد. به صورت زیر می توان مقدار کلید های ذخیره شده را بدست آورد. خروجی هر دستور بعد از آن نمایش داده شده است:

```
> GET (course)
```

```
Advanced Programming
```

```
> GET (university)
```

```
BASU
```

هدف از این تمرین پیاده سازی یک Key-Value Store بسیار ساده است. جزئیات این برنامه در ادامه آمده است.

جزئیات تمرین

در این تمرین قصد داریم یک Key-Value Store بسیار ساده طراحی کنیم. در این برنامه کاربر قادر است تعدادی دلخواه از این نوع پایگاه داده ایجاد و آن ها را مدیریت کند. روال کلی برنامه بدین صورت است که بعد از باز شدن برنامه کاربر بایستی دستوراتی را برای اجرا وارد کند. نماد > نشان دهنده ی این است که برنامه منتظر دریافت دستورات از کاربر است. تا هنگام پایان برنامه دستورات کاربر دریافت و اجرا می شوند. پایان برنامه توسط کاربر با دستور EXIT مشخص می شود. در این برنامه کاربر قادر به کار با چندین پایگاه است. برای مشخص کردن پایگاه داده مورد نظر از دستور USE استفاده می شود. اگر پایگاه داده ای با نام مورد نظر وجود نداشت یکی ساخته می شود و برنامه وارد حالت کار با آن پایگاه می شود. نام پایگاهی که کاربر در حال کار با آن است قبل از > نمایش داده می شود. برای مثال در برنامه زیر کاربر ابتدا پایگاه داده ای به نام courses می سازد.

```
> USE courses
```

```
courses>
```

با استفاده از دستور SET مقدار جدیدی به پایگاه داده ای که در حال کار بر روی آن هستیم افزوده می شود. این دستور به ترتیب یک کلید و مقدار متناظر با آن را دریافت می کند. اگر کلید موردنظر در پایگاه داده وجود نداشته باشد آن را اضافه و در غیر اینصورت مقدار آن را بروزرسانی می کند. مثلاً چند مقدار به دیتابیس courses اضافه می کنیم:

```
courses> SET (c1) (Basic Programming)
courses> SET (c2) (Advanced Programming)
courses> SET (c3) (Data Structures)
courses> SET (c2) (Artificial Intelligence)
```

با استفاده از دستور GET می توان مقدار متناظر با یک کلید را بدست آورد. این دستور کلید مورد نظر را به عنوان ورودی دریافت می کند. با اجرای این دستور مقدار کلید موردنظر نمایش داده می شود. در صورتی که چنین کلیدی وجود نداشته باشد پیغامی مبنی بر عدم وجود کلید نمایش داده می شود. در ادامه مثال قبلی:

```
courses> GET (c2)
Artificial Intelligence
courses> GET (c5)
Error: (c5) was not found.
```

دستور EXIST وجود یک کلید را چک می کند. به این صورت که کلید را به عنوان ورودی پذیرفته و در صورتی که این کلید در پایگاه داده وجود داشت مقدار 1 و در غیر اینصورت مقدار 0 را نمایش می دهد.

```
courses> EXIST (c2)
1
courses> EXIST (c5)
0
```

با استفاده از دستور DELETE می توان یک کلید را حذف کرد برای مثال:

```
courses> DELETE (c2)
courses> EXIST (c2)
0
```

با دستور EXIT برنامه بسته می شود.

نکات مهم

- دستورات بایستی دقیقاً در قالب مشخص شده از کاربر دریافت شوند.
- در صورتی که کاربر قبل از انتخاب یک پایگاه داده از دستورات GET, SET, EXIST, DELETE استفاده کند، بایستی پیامی به کاربر نمایش داده شود تا یک پایگاه داده انتخاب کند.
- در فایل تمرین باید پوشه ای به نام storage وجود داشته باشد. هر پایگاه داده در یک فایل جداگانه به همان نام ذخیره می شود.
- کاربر می تواند در هر زمان پایگاه داده ای که در حال کار بر روی آن است را تغییر دهد.
- کدنویسی خود را به صورت Modular انجام دهید.
- تمرین باید دارای Comment های مناسب باشد.

نکات تحویل

- ذخیره سازی داده ها باید به صورت دائمی باشد. به این صورت که با بستن و دوباره بازکردن برنامه بتوان به پایگاه داده های قبلی و مقادیر آنها دسترسی داشت.
- تمرین علاوه بر آپلود در سامانه Quera به صورت حضوری تحویل می شود.
- انجام تمرین به صورت انفرادی است. همفکری در انجام تمرین پذیرفته است اما در تحویل حضوری دانشجو بایستی قادر به پیاده سازی امکانات دیگر هم باشد.
- صرفاً فایل های Source و Header خود را تحویل دهید.