سوال سوم:

در جاوا استرینگ و کاراکتر جدا از هم هستند و مستقیما میتوان کلاس جدایی به اسم استرینگ داشت ولی درC برای تعریف استرینگ باید یک آرایه از جنس کاراکتر تعریف کرد و از آن استفاده کرد.

هم چنین در جاوا میتوان به کمک + استرینگ ها رو به یک دیگر متصل کرد که در C این قابلیت را نداشتیم.

برخی از توابع کلاسString :

1- متد (char charAt(int index))

متود charAt یک index به عنوان ورودی میگیرد و کاراکتر متناظر با آن را در رشته (String) می دهد. یعنی اگر برای مثال رشته ی مورد نظر ما "java" باشد در صورتی که عدد 0 را به عنوان ورودی دهیم اولین کاراکتر که'j' میباشد را برمیگرداند.

نکات ضروری این است که بدانیم شماره گذاری کاراکتر های یک رشته از سمت چپ و از 0 شروع میشود و هم چنین اینکه space ها هم یک کاراکتر حساب شده و شماره دارند.

سینتکس این متود public char charAt(int index) میباشد که طبق آن متوجه میشویم که خروجی از جنس char (کاراکتر) میباشد و ورودی هم از جنس int بوده و index مربوط به کاراکتری است که میخواهیم به دست بیاریم.

در صورتی که index ارسال شده در رشته وجود نداشته باشد خطای IndexOutOfBoundsException رخ میدهد.

2- متد(String replace(CharSequence old, CharSequence new))

این متد برای جایگزین کردن بخشی از یک رشته (String) یا با رشته (String)دیگر یا جایگزین کردن یک کاراکتر با کاراکتر مورد نظر در یک رشته استفاده می شود.

برای مثال اگر رشته مورد نظر str= "java" باشد با دستور(str.replace('j','p' کاراکترهای مذکور جابه جا میشوند و اگر رشته مورد نظر str= "Hello world" باشد با دستور(str.replace('Hello','Hi' رشته ی "Hi" جایگزین رشته "Hello" میشود.

بنابراین ورودی های ما میتوانند رشته یا کاراکتر باشند و خروجی ما تغییر یافته رشته ی اولیه با توجه به ورودی ها میباشد.

همچنین مقدار جدید را با هرچندتا مقدار قدیمی موجود در رشته اصلی جایگزین میکند.

3- متد (boolean equals(Object anther))

این متد دو رشته از جنس آبجکت را بر اساس محتوا بایکدیگر مقایسه می نماید. اگر حتی یک کاراکتر از رشته ی مورد نظر با کاراکتر موجود در رشته ی ورودی منطبق نباشد، مقدار بولی false را برمی گرداند. اگر تمامی کاراکترهای دو رشته با هم منطبق باشند، مقدار true را برمی گرداند.

بنابراین اگر دو object را بخواهیم مقایسه کنیم ورودی جنس آبجکت دیگر است که با رشته ی مورد نظر از نظر محتوا مقایسه می شود.برای خروجی نیز چنانچه کاراکترهای دو رشته با هم منطبق و برابر باشند، مقدار true و در غیر این صورت false را برمی گرداند.

هم چنین باید دقت شود که این تابع حروف بزرگ و کوچک را معادل در نظر نمیگیرد.

4- متد (String[] split(String regex , int limit))

متد split() برای تقسیم یک String به زیر رشته های آن بر اساس جداکننده داده شده استفاده می شود. درواقع وقتی قصد داریم یک String را بر اساس یک علامت خاص قطعه قطعه یا تجزیه کنیم از این متد استفاده می کنیم. هم چنین میتوان تعداد این تقسیمات را نیز کنترل کرد و بگوییم پس از تتجزیه کردن به چندتای اول نیاز داریم.

پس درواقع ورودی های آن به این ترتیب است. اول یک رشته میگیرد و براساس آن تقسم بندی را انجام میدهد. مثلا اگر شته ی اولیه ما "j a v a" باشد و رشته ی " " (یا همون space) را به عنوان ورودی جدا کننده دهیم ، رشته ی اصلی را به 4 قسمت تقسیم میکند. اگر ورودی دوم تابع را ندهیم به صورت default هر 4 قسمت را به عنوان خروجی برمیگرداند ولی اگر ورودی دوم که از جنسint میباشد را بدهیم درواقع تعداد آن ها را نیز تعیین کردیم ومثلا اگر بگوییم 3 ، بعد از تقسیم بندی 3 تای اول را به عنوان خروجی برمیگرداند.

بنابراین خروجی ما نیز این قسمت های ایجاد شده بع از تقسیم بندی است که از جنس stringمیباشد.

برای ذخیره خروجی های حاصل میتوان یک آرایه از جنس String تعریف کرد و قسمت های ایجاد شده را درآن ریخت.

5- متد (String concat(String str))

این متد رشته str را که در ورودی گرفته است به انتهای String (رشته ای) که این متد را صدا زده است وصل می کند.مثلا اگر یک String با نام s داشته باشیم"String s="java و اگر بخواهیم رشته ی "code" را به آن اضافه کنیم میتوان به این صورت عمل کرد:

```
s=s.concat("code");
```

همانطور که دیده میشود یک تساوی تعریف کردیم که سمت چپ آن همان رشته s میباشد. درواقع میگوییم پس از اتصال دو رشته حاصل را در همان متغیر رشته اول ذخیره کنید.
بنابراین ورودی این تابع یک رشته است و خروجی آن نیز یک رشته ی دیگر است.

6- متد (static String format(String format, Object ... args))

متد format در کلاس String همانند متد printf در جاوا یا همان printf موجود در زبان C میباشد به این صورت که یک رشته و تعدادی ورودی object میگیرد و از آنجایی که ورودی اول همان ورودی رشته ای فرمت بندی شده است ورودی های دیگر روی این ورودی تاثیر گذاشته و در جای مناسب با شکل دلخواه قرار میگیرند و یک خروجی رشته ای با فرمت دلخواه را برمیگرداند. در واقع در ورودی اول میتوان یک رشته داشت که در قسمت های مختلف آن مشخص کننده هایی(مثل:%d یا%f و...) قرار میگیرند و حتی برای آن ها میتوان طول میدان و... تعریف کرد و سپس در ورودی های بعدی مقدار دهی میشوند و همانطور که در ساختار این متد دیده مشود پس از ورودی اول یک ورودی از نوع بینهایت object تعریف میشود.

برای مثال در قطعه کد زیر خروجی میشود: ghade ali =175cm

```
String output = String.format("ghade %s = %dcm", "ali", 175);
System.out.println(output);
```

7- متد (int indexOf(int ch))

متد indexOf () برای پیدا کردن ایندکس یک کاراکتر خاص یا زیر رشته ای از یک رشته مشخص استفاده میشود.

باید به این نکته دقت کرد که این متد، ایندکس اولین کاراکتر یا زیر رشته ای که از چپ به راست در رشته پیدا میکند را برمیگرداند. همچنین اگر آن کاراکتر در رشته موجود نباشد مقدار 1- را برمیگرداند.

بنابراین یک مقدار کاراکتر یا رشته در ورودی میگیرد و زمانی که اولین وقوع آن را بیابد یک مقدار صحیح که تعیین کننده ایندکس آن میباشد را برمیگرداند