

## سوال چهارم - الف

ج ۱ : هر کدام از این مفاهیم را به اختصار توضیح می دهیم.

• **Array** : ساده ترین نوع ذخیره سازی چند نوع **primitive data** است به این صورت که **jvm**

یک تعداد مشخص و بدون این قابلیت را که تعداد آنها را دستکاری کند را برای آن در نظر میگیرد و به هر کدام آدرسی می دهد که کاملاً پشت سر هم هستند و آن را به همان ترتیب که ما برایش مشخص میکنیم ذخیره سازی می کند.

• **Arraylist** : همانطور که در بالا هم اشاره شد سائز یک **array** معمولی ثابت است و دیگر نمی توان به هیچ وجه نمیتوان به آن یک خانه ی دیگر را اضافه کرد و اگر بخواهیم به **array** مان یک مورد جدید اضافه کنیم باید به صورت دستی یک **array** دیگر بسازیم اما **array list** با این فلسفه عرضه شد که با استفاده از **generic** در **java** این مشکلات را حل کند و دیگر بتوان به راحتی به یک **array** یک مورد اضافه کرد و یا از آن کم کرد و این کار را به این صورت انجام می دهد که در هسته ی خود یک **array** دارد و اگر ما یک خانه ی جدید می خواهیم به آرایه اضافه کنیم یک آرایه ی جدید با سائز بزرگتر می سازد بعد از آن مقدار های خانه های قبلی را در جای درست جدیدشان کپی می کند و بعد از آن مقدار جدید را به انتهای **array** اضافه می کند.

• **Linked list** : این نوع از **data structure** با این صورت کار می کند که خانه های حافظه را به هم به صورت دو طرفه یا یک طرفه به هم وصل می کند و لزوماً به این صورت نیست که آن خانه ها دقیقاً به صورت فیزیکی پشت سر هم باشند به این صورت مشکلات سرعتی که با **array list** داریم را می توانیم برطرف کنیم زیرا اگر در **linked list** بخواهیم یک داده را حذف کنیم دیگر نیازی نیست به داده های دیگر دست بزنیم بر خلاف **linked list** که باید تمام خانه های بعد از آن را **bit by bit shift** کرد.

در انتها اینگونه میتوان نتیجه گرفت فرق این سه نوع **data structure** در فلسفه ی ذخیره سازی آنهاست و مشکلاتی هست که حل می کنند پس اگر سائز را از قبل می دانیم بهتر است که خیلی ساده یک آرایه به آن سائز بسازیم اما اگر سائز را نمی دانیم اما صرفاً میخواهیم داده ها را ذخیره کنیم و حذف کردن یا اضافه کردنی در کار نیست بهتر است از **array list** استفاده کنیم و از طرفی هم اگر میخواهیم بر روی داده ها تغییراتی را اجرا کنیم بهتر است از **linked list** استفاده کنیم.

**ج ۲:** به نظر من بهتر است که از linked list استفاده کنیم ؛ با توجه به اینکه سائز ما از قبل مشخص نیست پس استفاده کردن از array کار درستی نیست و با توجه به اینکه به صورت مرتب میخواهیم به بیماران مورد اضافه یا کم کنیم بهتر از linked list استفاده کنیم .

**ج ۳:** بله می شود زیرا در `arrayList` هیچ چکی برای برابری `item` ها انجام نمی شود و اگر می خواهیم این چک حتما انجام شود باید از `set` استفاده کنیم.

**ج4:** برای جواب به این سوال بهتر است که به شیوه ی ساختن array و ArrayList برویم ؛ هنگامی که ما یک array در java میسازیم jvm یک تعداد مشخصی از خانه ها را کنار هم قرار میدهد و با توجه به نوع ارایه برای آنها default value را قرار میدهد حال وقتی ما length میگیریم در واقع این طول برای ما حساب می شود و اگر مثلاً ارایه ای داریم که ۱۰۰ خانه طول دارد ولی فقط ۵ تا از خانه های آن را پر کرده ایم length آن را ۱۰۰ میگیریم نه ۵

به این نکته هم باید اشاره شود که length یک property از یک object به اسم array در java هست و گذاشتن ( ) در جلوی آن اشتباه هست !

از طرفی دیگر ArrayList در واقع تعداد دقیق item های ذخیره شده را در خودش ذخیره میکند و چون برای linked list طولی در کار نیست پس اگر ۵ تا item به ArrayList اضافه کرده باشیم در واقع size را ۵ میگیریم.