

پاسخ سوالات اول تا سوم

سوال اول:

شی ها نمونه هایی از دسته بندی های تعریف شده (کلاس ها) هستند که مفاهیم دنیای واقعی را در برنامه نویسی نمایش می دهند. هر یک، مقادیر مشخصی را در فیلدهای خود ذخیره می کنند. منظور از حالت یک شی، توصیف این مقادیر است. علاوه بر آن هر شی می تواند کار های مشخصی را انجام دهد یا تحت تاثیر عواملی قرار بگیرد، که نشان دهنده ی رفتار آن شی است و توسط method ها پیاده سازی می شود. هر شی نیز فارغ از مقادیری که در فیلدهای خود نگه داری می کند، آدرس مشخصی دارد. این آدرس هویت یک شی محسوب شده و آن را از دیگر اشیا متمایز می سازد، هم چنین توانایی تعامل با شی های دیگر را برای شی مورد نظر فراهم می کند.

مثلا "علی" یک شی در کلاس "انسان" است. نام، سن، قد، وزن و ویژگی های مشخصی دارد که حالت او را توصیف میکنند. مانند هر انسانی راه می رود، می خورد، کار می کند، خوشحال می شود، و رفتار مشخصی دارد و مهم ترین چیز در مورد "علی" آن است که علی رغم شباهت اسمی و ظاهری با علی های دیگر، اثر انگشت متفاوتی دارد که هویت او را منحصر به فرد می کند.

سوال دوم:

در برنامه نویسی شی گرا هر برنامه به قسمت های کوچک تری به نام شی تقسیم می شود که هم اطلاعات را نگه داری می کنند و هم با توجه به نوع خود، توابعی را اجرا می کنند. این نوع برنامه نویسی شباهت زیادی به دنیای واقعی دارد.

در برنامه نویسی ساخت یافته، کدها و دستورات با ساختار مشخصی پشت سر هم قرار می گیرند و هر برنامه به قسمت های کوچک تری به نام تابع تقسیم می شود.

مزایای برنامه نویسی شی گرا نسبت به ساخت یافته:

امنیت بیشتر و کنترل دسترسی

سادگی اضافه کردن اطلاعات و توابع

قابلیت استفاده ی مجدد از کدها

کارایی بیشتر در پروژه های مختلف

معایب برنامه نویسی شی گرا نسبت به ساخت یافته:

حجم بیشتر و سرعت کمتر برنامه

مناسب نبودن برای حل برخی مسائل خاص

سوال سوم:

در زبان برنامه نویسی C رشته ها در واقع آرایه ای از char ها هستند که یک کاراکتر null نیز در انتهای خود نگهداری می کنند.

در جاوا رشته ها آرایه نیستند، بلکه instance هایی از کلاس String هستند که داده ها را به صورت char نشان می دهند. اما پیاده سازی آنها با پیاده سازی آرایه ها در جاوا متفاوت است.

توضیح توابع:

char charAt(int index)

به عنوان ورودی یک اندیس می گیرد و کارکتر مربوط به این اندیس را برمیگرداند.

`String replace(CharSequence old, CharSequence new)`

به عنوان ورودی دو دسته کارکتر می گیرد، دسته ی دوم را در رشته جایگزین دسته ی اول می کند و رشته ی حاصل را برمیگرداند.

`boolean equals(Object another)`

یک شی را به عنوان ورودی می گیرد و در صورتی که رشته ی مورد نظر دارای حالت یکسانی با آن شی باشد، `true` و در غیر این صورت `false` برمیگرداند.

`String[] split(String regex, int limit)`

رشته ی ورودی را از رشته ی اصلی حذف و قسمت های باقی مانده را به چند رشته می شکند و این رشته ها را در قالب یک آرایه برمیگرداند. در صورت نیاز می توان تعداد رشته های آرایه ی خروجی را به صورت یک آرگومان ورودی مشخص کرد.

`String concat(String str)`

رشته ی ورودی را به رشته ی اصلی اضافه می کند.

`* static String format(String format, Object... args)`

رشته و شی ورودی را چاپ می کند

`int indexOf(char ch)`

به عنوان ورودی کارکتری را گرفته و اندیس اولین باری که آن کارکتر در رشته ظاهر شده را برمیگرداند