۱. تفاوت lambda با کلاس داخلی را توضیح دهید.

Inner class غالبا یک کلاس و یا یک اینترفیس بدون نام می‌باشد، در حالی که Lambda یک متد بی‌نام است. در Lambda تنها با یک اینترفیس تک متده مواجه هستیم که با نام functional interface میشناسیم، در صورتی که Inner class میتواند حاوی مقادیر دلخواهی از متدها باشد.

Lambda اجازه‌ی تعریف متغیرهای نمونه را نمیدهد اما Inner class این اجازه را میدهد.

از Lambda نمیتوان شئ ساخت اما این امکان در مورد Inner class وجود دارد.

در Lambda هنگام کامپایل کد، کلاس جدا(فایل جدا) ساخته نمیشود در حالی که در Inner class این اتفاق رخ میدهد.

در آخر Lambda بهترین گزینه برای هندل کردن اینترفیس و Inner class بهترین گزینه برای متدهای چندگانه است.

۲. تفاوت process و thread چیست؟

هنگام اجرای برنامه، اگر برنامه multi thread باشد، آن threadها درون یک process واحد قرار میگیرند، در حالی که هر process از process دیگر جداست(فضای مشترک آدرس دهی ندارد.) اما threadها فضای مشترک آدرس دهی دارند(thread های یک برنامه درون یک process واحد قرار دارند.).

۳. type erasure را توضیح دهید.

هرگاه یک متد جنریک تعریف کنیم به صورت unbounded در نظر گرفته شده و با کلاس object جایگزین می‌شود (چرا که هر نوعی باشد فرزندی از object خواهد بود). اگر از کلیدواژه‌ی extends برای مشخص نمودن حد بالایی (upper bounds) استفاده شود type آن متد یا کلاس با حد بالایی جایگزین می‌گردد. به این فرآیند که با حذف تایپ و جایگذاری آن با یک نوع صریح همراه است type erasure گفته می‌شود.

۴. کاربرد تگ dependency management در maven را توضیح دهید.

هنگامی که از ساختار ماژولار maven استفاده میکنیم، تمام dependency های ماژول فرزند از ماژول پدر ارث برده میشود. اگر ماژول پدر از dependency هایی استفاده کرده که با ورژن های خاصی از دیگر dependency ها سازگاری داشته باشد آنها را در تگ dependency management اضافه میکند. بدین شکل ماژول فرزند بدون مشخص نمودن version میتواند آنها را به پروژه‌ی خود اضافه کند.

۷.

A
D
E
F
C