اشکان شکیبا (9931030)

سوال اول:

الف)

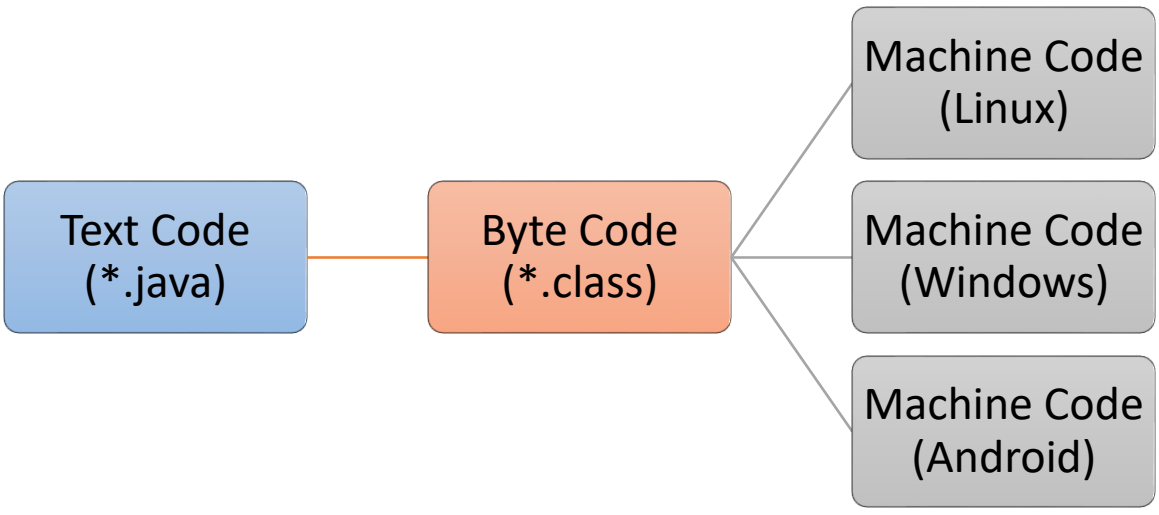
JVM که در واقع مخفف Java Virtual Machine است، وظیفه اجرای برنامه های نوشته شده به زبان جاوا را بر عهده دارد. JVM برای سیستم عامل های گوناگون نسخه های مختلفی دارد که بایت کد کامپایل شده (فایل های با پسوند class) را با توجه به ویژگی های سیستم عامل و به شکل مفسری (interpreter) به اجرا در می آورد. وجود JVM، عاملی ست که منجر به مستقل از پلتفرم بودن (Cross-platform) برنامه های نوشته شده به زبان جاوا میشود.

JRE یا Java Runtime Environment، محیط اجرای برنامه های جاواست که در برگیرنده JVM، و تعدادی از مهم ترین کتابخانه های این زبان است و بستر اجرای برنامه های جاوا را فراهم میکند.

JDK که مخفف Java Development Kit است، علاوه بر محتویات JRE، شامل ابزار هایی چون کامپایلر این زبان و فایل های راهنماست و ابزار مورد نیاز برای توسعه برنامه های به زبان جاواست.

ب)

هر برنامه جاوا ابتدا در قالب تعدادی فایل تکست با فرمت java توسط برنامه نویس توسعه داده میشود. سپس این فایل ها کامپایل شده و به شکل فایل هایی باینری با فرمت class در می آیند. پس از آن، JVM مورد نصب بر روی سیستم عامل، فایل های مرحله قبل را به شکل فایل های قابل اجرا توسط رایانه مورد نظر در آورده و در انتها آن را اجرا میکند



ج)

این متغیر ها، نماینده بخش هایی از حافظه هستند که داده های نسبتا ساده ای را در خود نگه میدارند و برای هر یک از انواع این متغیر ها، کلیدواژه ای در زبان جاوا تعیین شده است؛ اصلی ترین تفاوت این نوع از متغیر ها با سایر متغیر ها، این است که این نوع از متغیر ها، شی (object) محسوب نمیشوند.

از اصلی ترین متغیر های ابتدایی، میتوان به int برای اعداد صحیح، char برای کاراکتر های نوشتاری و double برای اعداد اعشاری اشاره کرد.

د)

در برنامه نویسی ساخت یافته (Structured Programming)، مسیری که باید توسط برنامه طی شود به گام های کوچکتر شکسته میشود و برنامه نویس می کوشد تا با حل کردن این مسائل کوچکتر و طراحی آنها به شکل توابع، بخش های گوناگون برنامه را آماده و سپس با کنار هم قرار دادن این رویه ها، مسیر کلی برنامه را تعیین کند.

از معروف ترین زبان های شامل این پارادایم می توان به C و Pascal اشاره کرد.

در برنامه نویسی شی گرا (Object Oriented Programming)، داده ها و دستورات ما، همگی در قالب کلاس ها و اشیا نوشته میشوند؛ به طوری که هر یک از عناصر مورد استفاده در برنامه، باید به شکل یک شی تعریف شود که داده های مربوط به آن به شکل فیلد ها (متغیر هایی که خود میتوانند شی های دیگری باشند) و فرایند های مربوط به آن در قالب متد ها تعیین میشوند.

از معروف ترین زبان های شامل این پارادایم می توان به C++، Java و C# اشاره کرد.

در برنامه نویسی تابعی (Functional Programming)، برنامه به شکل چند تابع کوچک تر نوشته می شود که این توابع همواره به ازای ورودی یکسان، خروجی یکسانی داشته و همچنین، مقادیر متغیر های خارج از خود را تغییر نمی دهند. در اینجا نیز مشابه پارادایم ساخت یافته، با کنار هم قرار دادن این توابع مستقل از هم، برنامه شکل میگیرد.

از معروف ترین زبان های شامل این پارادایم می توان به JavaScript و Scala اشاره کرد.

ه)

Class: یک طرح برای اشیاست. میتوان هر Class را به عنوان طرح و نمونه اولیه هر نوع از اشیا در نظر گرفت که شامل ویژگی های آنها (در قالب فیلد) و فرآیند های مربوط به آنها (در قالب متد) است.

Object: با تعریف Class تنها مشخصات اشیا (Object ها) تعیین میشود؛ اما لازم است تا با ایجاد Object ها، داده های منحصر به فردی از Class مورد نظر را پیاده سازی و برنامه را کامل تر کنیم.

Constructor: که به عنوان بخشی از Class تعریف میشود، زمانی استفاده میشود که قصد ساختن Objectی از Class مورد نظر را داشته باشیم و دو وظیفه را انجام میدهد، اختصاص بخشی از حافظه به Object مورد نظر و مقدار دهی به فیلد های آن. بسته به نوع مقدار دهی، Constructor میتواند ورودی هایی نیز داشته باشد.

Method: فرآیند های مربوط به هر Class، در قالب توابعی در همان Class تعریف میشوند که بسته به نوع فرآیند میتوانند ورودی ها و خروجی های گوناگونی داشته باشند.

Parameter: داده هایی هستند که توسط متد ها پردازش میشوند تا فرآیند های مورد نظر انجام شده و مقادیر خروجی مد نظر محاسبه شوند. Parameter ها میتوانند از انواع گوناگونی از داده ها باشند و هر متد میتواند هر تعدادی از آنها را داشته باشد.

Instance: متغیر هایی هستند که داخل هر Class و خارج از Method ها تعریف میشوند (فیلد ها). این متغیر ها همراه با Object مورد نظر به وجود می آیند و همراه با آن از بین می روند. همچنین در تمام Class و برای همه Method های آن Class، قابل دسترسی هستند.

و)

اصلی ترین تفاوت تابع و متد در محل تعریف آنهاست؛ متد چیزی تقریبا مشابه تابع است با این تفاوت که در بدنه یک کلاس تعریف میشود و فرآیندی را بر روی اشیای ساخته شده از آن کلاس انجام میدهد و به این ترتیب، متد ها ارتباط اجزای داخلی کلاس را با محیط بیرونی آن فراهم میکنند؛ در حالی که توابع مستقل از اشیا و کلاس ها هستند.

تفاوت دیگر در نحوه فراخوانی آنهاست؛ به طوری که تابع را میتوان تنها با نام آن فراخوانی کرد اما برای متد، باید ابتدا یکی از اشاره گر های به کلاس حاوی متد را ذکر کنیم.

یکی دیگر از تفاوت های آنها، در داده های ورودی آنهاست؛ به طوری که داده های ورودی توابع باید به عنوان آرگومان های تابع وارد آن شوند، در حالی که در متد، علاوه بر آرگومان های ورودی، میتوان از متغیر های موجود در کلاس شامل متد نیز استفاده کرد.

به طور کلی میتوان گفت توابع مستقل از اشیا و دیگر عناصر برنامه هستند اما متد ها وابسته به کلاسی هستند که در آن تعریف شده اند.

سوال دوم:

1) نادرست 2) درست 3) نادرست 4) نادرست 5) نادرست

سوال سوم:

A) methods – fields

B) class – objects

C) classes

D) java

E) javac

F) java

G) class

H) bytecodes