اشکان شکیبا (9931030)

سوال اول

الف)

State به معنای حالت، اشاره به فیلد ها (Instance variables) دارد که تعیین کننده مشخصات اشیا هستند.

Behavior به معنای رفتار، اشاره به متد ها دارد که توصیف کننده رفتار های اشیا هستند.

Identity به معنای هویت، مربوط به اشاره گر های اشیا (References) ست که تمایز میان اشیای از نوع (Class) یکسان را ایجاد میکند.

ب)

برای تعامل میان اشیای A و B، دو راه وجود دارد:
I) تعریف اشیایی از نوع (Class) A در میان فیلد های B (و یا برعکس)

II) فراخوانی متد های public شی A در متد ها یا کانستراکتور B (و یا برعکس)

ج)

Overloading به معنای تفاوت در پارامتر های ورودی متد یا کانستراکتور است که این تفاوت میتواند در تعداد، نوع و یا ترتیب پارامتر ها ظاهر شود.

Casting به معنای تغییر نوعی از داده به نوعی دیگر می باشد، مثل تغییر عددی اعشاری به صحیح. بسته به نوع Casting، این عمل میتواند باعث آسیب دیدن بخش هایی از داده مورد نظر شود (برای مثال در تبدیل عدد اعشاری به صحیح، داده های مربوط به ارقام پس از اعشار در این فرآیند از بین میروند).

Modularization به معنای تقسیم برنامه به بخش های کوچکتر (Module) می باشد که تلاش میشود تا بخش ها تا حد ممکن مستقل از یکدیگر باشند تا بتوان فرآیند توسعه بخش های مختلف را به شکلی موازی پیش برد و با کنار هم گذاشتن بخش های توسعه داده شده کوچکتر، به برنامه جامع دست یافت.

Abstraction به معنای انتزاع و یا پنهان کردن جزئیات است که در آن جزئیات مسئله در نظر گرفته نمیشوند و بدون توجه به فرآیند های پیچیده ای که در پشت صحنه انجام میشوند، تنها به بررسی خروجی کار میپردازیم.

د)

Immutable Object، شی ست که مقادیر فیلد های آن تنها در هنگام تشکیل اولیه (کانستراکتور) مقداردهی میشوند و پس از آن قابل تغییر نیستند (به دلیل private بودن فیلد ها و عدم وجود متد setter در کلاس مورد نظر).

به عنوان مثال، اشیای تشکیل شده از کلاس Transaction در سوال پنجم این تمرین، دارای این ویژگی هستند.

سوال دوم

1) نادرست، بعد از تعریف کانستراکتور، کامپایلر این کار را نمیکند.

2) درست

3) نادرست، تنها برای Object ها میتوانیم ArrayList داشته باشیم.

4) درست

5) درست

6) درست

7) نادرست، این تعریف به فیلد ها اختصاص دارد.

8) درست

سوال سوم

الف)

1)

در صورتی که نام دو متد و تعداد، نوع و ترتیب پارامتر های ورودی دو متد یکسان باشند، نمیتوان برای آنها خروجی های متفاوتی در نظر گرفت.

در صورت تفاوت در هر یک از موارد فوق، تعریف متد دوم ممکن خواهد شد.

2)

کانستراکتور هیچ نوع return typeای ندارد و برای مقداردهی فیلد های کلاس مورد نظر هنگام ساخت اشیای جدید از آن کلاس استفاده میشود؛ در حالیکه استفاده از متد میتواند دلایلی همچون انجام

دستوراتی معین، محاسبه و بازگرداندن مقداری خاص و یا دریافت و تنظیم مقادیر فیلد های شی پس از ساخت آن باشد.

3)

بله! برای این کار باید کانستراکتور های مورد نظر در تعداد، نوع و یا ترتیب پارامتر های ورودی تفاوت داشته باشند.

برای مثال کانستراکتور ClassName(int x, char c) میتواند به شکل های ClassName(int x)، ClassName(float x, char c) و یا ClassName(char c, int x)، Overload شود اما امکان این کار به شکل ClassName(int a, char b) وجود ندارد.

ب)

کد اصلاح شده در پوشه Q3B قرار دارد.

همچنین توضیح مراحل مختلف اصلاح برنامه در خود کد و به شکل کامنت قرار داده شده اند.

سوال چهارم

الف)

1)

Array را میتوان ساده ترین و در عین حال سریع ترین نوع کالکشن در جاوا دانست. در Array همه عناصر به ترتیب با index هایی که از صفر شروع میشود دسته بندی شده و به همین شکل در حافظه نیز ذخیره میشوند. با توجه به سادگی ذخیره سازی آن در حافظه، فرآیند دسترسی به اجزای آن سریع تر از دیگر کالکشن ها انجام میشود. از لحاظ سینتکس نیز تفاوت آن با دیگر انواع کالکشن در نحوه دسترسی به آن است که نیازی به متد هایی چون get ندارد و با علامت های [و] نیز میتوان به اجزای آن دسترسی داشت. از محدودیت های Array هم میتوان به دشوار بودن حذف اشیای آن و پویا نبودن تعداد عناصر آن اشاره کرد.

ArrayList راه حلی برای محدودیت های Array است. کلاسی Generic با کارکردی مشابه Array که امکان حذف عناصر کالکشن و افزودن عناصر جدید به آن را به شکلی پویا فراهم می آورد.

اما کالکشن LinkedList ساختار متفاوتی با دو نوع قبلی دارد؛ این کالکشن برخلاف قبلی ها، چینش عناصر را بر پایه index انجام نمیدهد و ارتباط عناصر توسط رفرنس های درون هر عنصر انجام میشود. بنابراین برای دسترسی به عناصر میانی کالکشن باید از روی همه عناصر قبلی پیمایش کرد و از این رو

دسترسی به عناصر در آن کند تر از کالکشن های قبلی ست، اما فرآیند حذف عناصر و یا قرار دادن عنصر جدید در جایگاهی غیر از آخر، در LinkedList ساده تر و سریع تر از دیگر کالکشن ها انجام میشود.

2)

ArrayList، زیرا علاوه بر افزوده شدن عناصر به شکلی پویا (و تغییر تعداد اعضای کالکشن بدون محدودیت)، امکان حذف عناصر از کالکشن را نیز فراهم میکند.

3)

بله، وجود دارد!

4)

متد size در ArrayList تعداد عناصر افزوده شده به آن را نشان می دهد که با افزوده شدن هر عنصر جدید، مقدار آن افزایش و با حذف هر عنصر، مقدار آن کاهش می یابد.

اما length در Array، بیانگر حافظه تعیین شده برای Array است؛ به بیان دیگر، مقدار آن برابر تعداد عناصر قابل ذخیره در Array بوده و از ابتدای تشکیل Array تعیین میشود و بعد از آن تغییری نمیکند.

ب)

برنامه نوشته شده در پوشه Q4B قرار دارد.

سوال پنجم

برنامه نوشته شده در پوشه Q5 قرار دارد.

برای اجرای برنامه کافی است از فایل Run From Here استفاده کنید.

این برنامه در دو نسخه آماده شده است. نسخه اصلی که در پوشه with cls قرار دارد، دارای متدی به نام cls در کلاس Main است که وظیفه پاک کردن صفحه کنسول در هنگام تغییر صفحات منو را دارد؛ اما با توجه به مشکل داشتن این پدیده در برخی نسخه های جاوا، نسخه دیگری در پوشه without cls قرار داده شده که در صورت عدم کارکرد صحیح نسخه اصلی، میتوان از آن استفاده کرد.