



کد محصول
ES1632



آخرین بروزرسانی
۱۲ اردیبهشت ۱۴۰۵

سوالات استخدامی

مهندسی نرم افزار

- ✓ ویژه آزمون های استخدامی
- ✓ نسخه رایگان شامل ۵۰ سوال (تعداد کمتر و تنها برخی دارای پاسخ)
- ✓ برای تهیه نسخه اصلی، با ۲۰۲ سوال به همراه پاسخنامه تشریحی، به سایت ایران عرضه مراجعه نمایید.



لینک های مفید آزمون های استخدامی

خرید این محصول	سوال‌ات رایگان دروس عمومی با پاسخنامه
خرید سوال‌ات استخدامی ۱۰ سال اخیر	خرید درسنامه دروس عمومی
خرید اصل سوال‌ات سال ۱۴۰۴	خرید اصل سوال‌ات سال ۱۴۰۳
خرید سوال‌ات مصاحبه	شبکه های اجتماعی ایران عرضه (فایل های رایگان + تخفیفات هفتگی + اخبار)
(برای مشاهده هر بخش روی آن بزنید )	
<p>آخرین بروزرسانی های محصول:</p> <p>۱۴۰۵/۰۲/۰۲ تالیف مجدد محصول</p>	

۲ سوال ابتدایی این فایل، دارای پاسخنامه تشریحی می باشد. در صورت تمایل به دریافت سوالات بیشتر با جواب تشریحی می توانید این محصول را از سایت ایران عرضه خریداری نمایید.

خرید محصول

❖ سوالات مهندسی نرم افزار تالیف ایران عرضه

۱- کدام گزینه در خصوص وظایف کلاس های کنترل گر در طراحی صحیح است؟ (منبع ایران عرضه)

(۱) ذخیره سازی دائمی داده ها در پایگاه داده

(۲) مدیریت جریان یک واحد کار از ابتدا تا انتها و هماهنگی میان اشیای مختلف

(۳) جایگزینی اشیای مرزی در ارتباط با کاربر

(۴) شناسایی کلاس ها در مرحله تحلیل و قبل از طراحی

❑ پاسخ سایت ایران عرضه: گزینه ۲ ➡ کلاس های کنترل گر یک واحد کار (UML031) را از ابتدا تا انتها مدیریت می کنند. یعنی کلاس های کنترل گر را می توان طوری طراحی کرد که (۱) ایجاد یا پیکربندی اشیای موجودیت، و (۲) معرفی اشیای مرزی به هنگام کسب اطلاعات از اشیای موجودیت، (۳) ارتباطات پیچیده میان مجموعه های اشیای، (۴) اعتبارسنجی داده های ارتباطی میان اشیای یا میان کاربر و برنامه را مدیریت کنند. به طور کلی، کلاس های کنترل گر تا شروع فعالیت طراحی در نظر گرفته نمی شوند.

۲- در مدل سازی تحلیل، گروه بندی عناصر مختلفی مانند use case ها و کلاس های تحلیل که به صورت یک مجموعه در کنار هم قرار می گیرند، چه نام دارد؟

(۱) پکیج طراحی (۲) پکیج پیاده سازی (۳) پکیج تحلیل (۴) کلاس کنترلی

❑ پاسخ سایت ایران عرضه: گزینه ۳ ➡ بخش مهمی از مدل سازی تحلیل، گروه بندی است. یعنی عناصر گوناگون مدل تحلیل (مثل use case ها و کلاس های تحلیل) طوری گروه بندی می شوند که یک پکیج از آن ها تشکیل شود و به آن ها پکیج تحلیل گفته می شود؛ به هر کدام از این پکیج ها یک نام مشخص داده می شود.

۳- کدام یک از مدل های زیر به چگونگی تبدیل اشیای داده ای توسط قابلیت های پردازش و تعامل آن ها با یکدیگر اشاره دارد؟

(۱) مدل سازی رفتار سیستم (۲) مدل سازی مبتنی بر الگو

(۳) مدل سازی جریان کار (۴) مدل سازی رابط کاربر

۴- کدام گزینه درست ترین توصیف از "جدول فعال سازی برنامه ها" در مشخصات کنترل (CSPEC) است؟

(۱) روشی برای نمایش داده های ورودی سیستم

۲) جزییاتی از فرایندهای داخلی که در نتیجه رویدادها فعال می شوند

۳) ساختاری برای توصیف رفتارهای ترتیبی سیستم

۴) لیستی از برنامه ها که به ترتیب و در واکنش به رویدادها اجرا می شوند

۵- کدام یک از گزینه های زیر جزء اصلی ترین عناصر در توصیف مدل خواسته ها محسوب می شود؟

۱) مدل داده ها ۲) use case ۳) مدل رفتاری ۴) مدل مبتنی بر کلاس

۶- اشیای محتوایی در یک سیستم نرم افزاری چگونه شناسایی می شوند؟ (تالیف توسط سایت ایران عرضه)

۱) با تحلیل معماری فنی سیستم و تعیین اجزای وابسته به سرور

۲) با تحلیل الگوریتم های پردازشی و منطق کنترلی سیستم

۳) با مطالعه جزئیات کد منبع و ساختار کلاس ها در مرحله پیاده سازی

۴) از روی use case ها یا با بررسی شرح سناریو و ارجاع های مستقیم و غیرمستقیم به محتوا

۷- در مدل سازی گفت و گذار در برنامه های تحت وب، تمرکز بر کدام یک از موارد زیر است؟

۱) در نظر گرفتن سیاست جابه جایی گروه های کاربری بین عناصر برنامه

۲) تعیین ساختار داده ها و نحوه ذخیره سازی اطلاعات

۳) پیاده سازی جزئیات فنی رابط کاربر در مرحله کدنویسی

۴) طراحی الگوریتم های پردازشی سیستم و منطق تجاری

۸- کدام یک از مراحل طراحی نرم افزار در ابتدا نمای کلی از سیستم را تصویر می کند و فرآیند طراحی نرم افزار چگونه به

تکامل می رسد؟

۱) طراحی در سطح پایین از انتزاع و تکامل از طریق اضافه کردن ویژگی های جدید به سیستم

۲) طراحی در سطح بالای انتزاع و تکامل از طریق پالایش های بعدی در سطوح پایین تر از انتزاع

۳) طراحی جزئیات فنی سیستم و تکامل از طریق تغییرات در ساختار داده ها

۴) طراحی منطق تجاری سیستم و تکامل از طریق کدنویسی و پیاده سازی جزئیات فنی

۹- کدام گزینه درباره نقش پیمانانه بندی در طراحی نرم افزار، صحیح نیست؟

۱) پیمانانه بندی با کاهش پیچیدگی کلی برنامه، درک و مدیریت نرم افزار را آسان تر می کند.

۲) نرم افزار یکپارچه با افزایش مسیرهای کنترلی و ارجاعات، فهم برنامه را برای مهندس نرم افزار دشوار می سازد.

۳) تقسیم طراحی به چندین پیمانانه می تواند به کاهش هزینه ساخت نرم افزار منجر شود.

۴) پیمانانه بندی باعث افزایش تعداد متغیر ها و پیچیدگی کلی سیستم به منظور بهبود کنترل برنامه می شود.

۱۰- طراحی معماری نرم افزار، مشابه نقشه ساختمان هستند که برای ما را فراهم می آورند.

۱) جزئیات کامل پیاده سازی و کدنویسی سیستم

۲) دیدی کلی از ساختار، اجزا و ارتباط آن ها با یکدیگر

۳) الگوریتم های پردازشی و منطق اجرایی برنامه

۴) نحوه مدیریت حافظه و منابع سخت افزاری

۱۱- در طراحی نرم افزار، وقتی از «تصمیم گیری های ساختاری عمده» سخن گفته می شود، منظور از این تصمیم ها چیست و چگونه بر فرآیند طراحی تأثیر می گذارد؟

۱) تصمیم هایی که بر اساس خواسته های کاربری گرفته می شوند و مستقیماً به طراحی رابط کاربری مربوط می شوند.

۲) تصمیم هایی که در سطوح پایین تر از انتزاع گرفته می شوند و به کدنویسی سیستم مربوط می شوند.

۳) تصمیم هایی که بر روابط میان اجزای مختلف سیستم تأثیر دارند و ساختار کلی سیستم را تعیین می کنند.

۴) تصمیم هایی که به انتخاب ابزارها و زبان های برنامه نویسی مربوط می شوند و بر کدنویسی سیستم تأثیر دارند.

۱۲- معماری های جریان داده ها معمولاً در چه شرایطی به کار برده می شوند؟

۱) هنگامی که سیستم نیاز به مدیریت همزمان تعداد زیادی تراکنش مالی با قابلیت rollback داشته باشد.

۲) در سیستم هایی که تصمیم گیری ها مبتنی بر رویدادها و وضعیت های داخلی است.

۳) هنگامی که داده های ورودی باید از طریق مجموعه ای از مؤلفه های محاسباتی و پردازشی به داده های خروجی تبدیل شوند.

۴) هنگامی که طراحی باید حول یک پایگاه داده متمرکز باشد و دسترسی های همزمان چندین کاربر با سطوح دسترسی متفاوت را مدیریت کند.

۱۳- کدام یک از مراحل طراحی ساختار برنامه به طور خاص مربوط به گذار از نمودار جریان داده ها (DFD) به ساختار برنامه است؟

۱) تعیین نوع جریان اطلاعات

۲) تعریف سلسله مراتب کنترلی

۳) مشخص کردن مرزهای جریان اطلاعات

۴) نگاشت DFD به یک ساختار برنامه ای

۱۴- کدام یک از ویژگی های زیر از مزایای اصلی پالایش معماری نرم افزار به شمار می رود؟ (منبع سوالات سایت ایران عرضه)

۱) ارزیابی و انتخاب بهترین سبک های معماری برای هر پروژه

۲) بهینه سازی پیچیدگی کدنویسی

۳) استفاده از بیشتر مؤلفه ها برای افزایش انعطاف پذیری سیستم

۴) ایجاد ساختار داده ای پیچیده برای پاسخگویی بهتر به خواسته های اطلاعاتی

۱۵- نقشهای مهمی که مؤلفه های سنتی (پیمانه ها) در معماری نرم افزار ایفا می کنند، کدامند؟

۱) مؤلفه داده، مؤلفه واسط، مؤلفه پردازش

۲) مؤلفه کنترلی، مؤلفه دامنه مسئله، مؤلفه زیرساختی

۳) مؤلفه ورودی، مؤلفه خروجی، مؤلفه ذخیره سازی

۴) مؤلفه ساختاری، مؤلفه منطقی، مؤلفه اجرایی

۱۶- بر اساس اصل هم ارزی استفاده ی مجدد از نسخه ها (REP)، برای مدیریت و کنترل تکامل نسخه های جدید نرم افزار چه راهکاری توصیه می شود؟

۱) گروه بندی کلاس های قابل استفاده مجدد در قالب بسته ها

۲) طراحی مجدد هر کلاس به صورت مستقل در هر نسخه

۳) حذف پشتیبانی از نسخه های قدیمی و تمرکز کامل بر نسخه های جدید

۴) محدود کردن استفاده مجدد فقط به مؤلفه های کنترلی

۱۷- اتصال محتوا در مهندسی نرم افزار در چه حالتی رخ می دهد؟

۱) زمانی که دو مؤلفه از طریق واسط های تعریف شده با یکدیگر تبادل داده می کنند.

۲) زمانی که یک مؤلفه داده ها را برای خواندن از مؤلفه دیگر دریافت می کند.

۳) زمانی که یک مؤلفه به طور مستقیم داده های داخلی مؤلفه دیگر را اصلاح می کند که این کار نقض اصل پنهان سازی اطلاعات است.

۴) زمانی که چند مؤلفه از یک ساختار داده مشترک و عمومی استفاده می کنند.

۱۸- چه زمانی اشیاء محتوایی در برنامه های تحت وب باید به صورت مؤلفه های سازمان دهی شده پیاده سازی شوند؟

۱) وقتی محتوا پویا و روابط پیچیده ای داشته باشد.

۲) وقتی برنامه کوچک و ساده باشد.

۳) وقتی اشیاء محتوایی نیازی به تغییر نداشته باشند.

۴) وقتی محتوا ثابت باشد.

۱۹- مهندسی دامنه چه هدفی دارد؟

۱) تولید نرم افزارهای کاملاً اختصاصی برای امکان استفاده ی مجدد در فرآیند توسعه.

۲) توزیع مؤلفه های نرم افزاری و تمرکز بر تحلیل نیازمندی ها بدون ساخت و انتشار مؤلفه ها.

۳) افزایش وابستگی میان مؤلفه ها برای ساده سازی فرآیند توسعه.

۴) شناسایی، پیاده سازی و توزیع مؤلفه های نرم افزاری برای استفاده در سیستم های موجود و جدید.

۲۰- کتابخانه استفاده مجدد در چه مواردی کاربرد دارد؟ (منبع فروشگاه اینترنتی ایران عرضه)

۱) نگهداری موقت کدهای خام و ناقص در طول فرآیند توسعه، بدون پشتیبانی از طبقه بندی، مستندسازی و امکان بازیابی هدفمند مؤلفه ها

۲) مستندسازی نیازمندی ها و طراحی های اولیه ی نرم افزار به صورت ایستا، بدون فراهم کردن سازوکار ذخیره سازی و استفاده ی مجدد از مؤلفه ها و محصولات کاری

۳) ذخیره سازی، طبقه بندی و بازیابی مؤلفه های نرم افزاری و محصولات کاری قابل استفاده ی مجدد مانند الگوها، چارچوب ها، کد، موارد آزمون و راهنماها در یک مخزن نرم افزاری

۴) اجرای مستقیم مؤلفه های نرم افزاری بدون نیاز به الگوی طبقه بندی، بانک اطلاعاتی و ابزارهای ارجاع و بازیابی

۲۱- چه رابطه ای میان بار حافظه کاربر و احتمال خطای او در تعامل با سیستم وجود دارد؟

۱) هرچه کاربر مجبور به حفظ جزئیات بیشتری باشد، احتمال خطای او کاهش می یابد.

۲) هرچه کاربر مجبور به حفظ جزئیات بیشتری باشد، احتمال خطای او افزایش می یابد.

۳) بار حافظه کاربر تأثیری بر احتمال خطای او ندارد.

۴) بار حافظه کاربر به طور مستقیم با دقت عملکرد او ارتباط دارد.

۲۲- تحلیل جریان کاری به چه چیزی اشاره دارد؟

۱) شناسایی اشیای رابط کاربر و ویژگی های آن ها

۲) بررسی وظایف یک کاربر منفرد در تعامل با سیستم

۳) بررسی روند انجام یک فرایند توسط چند کاربر با نقش های متفاوت

۴) پیاده سازی فنی و کدنویسی فرایندهای سیستم

۲۳- کدام دو مورد از ویژگی های مهم پاسخ سیستم هستند؟

۱) دقت و قابلیت اطمینان

۲) سرعت پردازش و ظرفیت سیستم

۳) یکنواختی رابط کاربر و سادگی طراحی

۴) طول پاسخ و تغییرپذیری زمان پاسخ

۲۴- شناسایی اشیای واسط کاربر در طراحی واسط برنامه ی تحت وب مستلزم چه فعالیتی است؟

۱) تعیین معماری کلی سیستم، شناسایی کلاس های سفارشی و نحوه ی ارتباط مؤلفه ها

۲) طراحی پایگاه داده و مشخص کردن ساختار ذخیره سازی اطلاعات

۳) جستجو در کتابخانه ی اشیای موجود برای یافتن کلاس های قابل استفاده ی مجدد واسط و شناسایی کلاس های سفارشی

مورد نیاز

۴) پیاده سازی منطق پردازش و کنترل جریان برنامه

۲۵- هدف از ارزیابی نمونه ی اولیه عملیاتی واسط کاربر چیست؟ - طراحی شده توسط ایران عرضه -

۱) شناسایی خطاهای فنی در کدنویسی سیستم.

۲) بررسی میزان تطابق واسط کاربر با استانداردهای برنامه نویسی.

۳) اطمینان از این که واسط کاربر نیازهای کاربران را برآورده می سازد یا خیر.

(۴) افزایش پیچیدگی واسط کاربر برای بهبود کارایی سیستم.

۲۶- کدام گزینه در خصوص وظایف کلاس های کنترل گر در طراحی صحیح است؟

(۱) ذخیره سازی دائمی داده ها در پایگاه داده

(۲) مدیریت جریان یک واحد کار از ابتدا تا انتها و هماهنگی میان اشیای مختلف

(۳) جایگزینی اشیای مرزی در ارتباط با کاربر

(۴) شناسایی کلاس ها در مرحله تحلیل و قبل از طراحی

۲۷- در مدل سازی تحلیل، گروه بندی عناصر مختلفی مانند use case ها و کلاس های تحلیل که به صورت یک مجموعه در

کنار هم قرار می گیرند، چه نام دارد؟

(۱) پکیج طراحی (۲) پکیج پیاده سازی (۳) پکیج تحلیل (۴) کلاس کنترلی

۲۸- کدام یک از مدل های زیر به چگونگی تبدیل اشیای داده ای توسط قابلیت های پردازش و تعامل آن ها با یکدیگر اشاره

دارد؟

(۱) مدل سازی رفتار سیستم (۲) مدل سازی مبتنی بر الگو

(۳) مدل سازی جریان کار (۴) مدل سازی رابط کاربر

۲۹- کدام گزینه درست ترین توصیف از "جدول فعال سازی برنامه ها" در مشخصات کنترل (CSPEC) است؟

(۱) روشی برای نمایش داده های ورودی سیستم

(۲) جزییاتی از فرایندهای داخلی که در نتیجه رویدادها فعال می شوند

(۳) ساختاری برای توصیف رفتارهای ترتیبی سیستم

(۴) لیستی از برنامه ها که به ترتیب و در واکنش به رویدادها اجرا می شوند

۳۰- کدام یک از گزینه های زیر جزء اصلی ترین عناصر در توصیف مدل خواسته ها محسوب می شود؟

(۱) مدل داده ها (۲) use case (۳) مدل رفتاری (۴) مدل مبتنی بر کلاس

۳۱- اشیای محتوایی در یک سیستم نرم افزاری چگونه شناسایی می شوند؟

(۱) با تحلیل معماری فنی سیستم و تعیین اجزای وابسته به سرور

(۲) با تحلیل الگوریتم های پردازشی و منطق کنترلی سیستم

(۳) با مطالعه جزئیات کد منبع و ساختار کلاس ها در مرحله پیاده سازی

(۴) از روی use case ها یا با بررسی شرح سناریو و ارجاع های مستقیم و غیرمستقیم به محتوا

۳۲- در مدل سازی گفت وگذار در برنامه های تحت وب، تمرکز بر کدام یک از موارد زیر است؟

(۱) در نظر گرفتن سیاست جابه جایی گروه های کاربری بین عناصر برنامه

(۲) تعیین ساختار داده ها و نحوه ذخیره سازی اطلاعات

۳) پیاده سازی جزئیات فنی رابط کاربر در مرحله کدنویسی

۴) طراحی الگوریتم های پردازشی سیستم و منطق تجاری

۳۳- کدام یک از مراحل طراحی نرم افزار در ابتدا نمای کلی از سیستم را تصویر می کند و فرآیند طراحی نرم افزار چگونه به تکامل می رسد؟

۱) طراحی در سطح پایین از انتزاع و تکامل از طریق اضافه کردن ویژگی های جدید به سیستم

۲) طراحی در سطح بالای انتزاع و تکامل از طریق پالایش های بعدی در سطوح پایین تر از انتزاع

۳) طراحی جزئیات فنی سیستم و تکامل از طریق تغییرات در ساختار داده ها

۴) طراحی منطق تجاری سیستم و تکامل از طریق کدنویسی و پیاده سازی جزئیات فنی

۳۴- کدام گزینه درباره نقش پیمانه بندی در طراحی نرم افزار، صحیح نیست؟ (منتشر کننده سوالات ایران عرضه)

۱) پیمانه بندی با کاهش پیچیدگی کلی برنامه، درک و مدیریت نرم افزار را آسان تر می کند.

۲) نرم افزار یکپارچه با افزایش مسیرهای کنترلی و ارجاعات، فهم برنامه را برای مهندس نرم افزار دشوار می سازد.

۳) تقسیم طراحی به چندین پیمانه می تواند به کاهش هزینه ساخت نرم افزار منجر شود.

۴) پیمانه بندی باعث افزایش تعداد متغیرها و پیچیدگی کلی سیستم به منظور بهبود کنترل برنامه می شود.

۳۵- طراحی معماری نرم افزار، مشابه نقشه ساختمان هستند که برای ما را فراهم می آورند.

۱) جزئیات کامل پیاده سازی و کدنویسی سیستم

۲) دیدی کلی از ساختار، اجزا و ارتباط آن ها با یکدیگر

۳) الگوریتم های پردازشی و منطق اجرایی برنامه

۴) نحوه مدیریت حافظه و منابع سخت افزاری

۳۶- در طراحی نرم افزار، وقتی از «تصمیم گیری های ساختاری عمده» سخن گفته می شود، منظور از این تصمیم ها چیست

و چگونه بر فرآیند طراحی تأثیر می گذارد؟

۱) تصمیم هایی که بر اساس خواسته های کاربری گرفته می شوند و مستقیماً به طراحی رابط کاربری مربوط می شوند.

۲) تصمیم هایی که در سطوح پایین تر از انتزاع گرفته می شوند و به کدنویسی سیستم مربوط می شوند.

۳) تصمیم هایی که بر روابط میان اجزای مختلف سیستم تأثیر دارند و ساختار کلی سیستم را تعیین می کنند.

۴) تصمیم هایی که به انتخاب ابزارها و زبان های برنامه نویسی مربوط می شوند و بر کدنویسی سیستم تأثیر دارند.

۳۷- معماری های جریان داده ها معمولاً در چه شرایطی به کار برده می شوند؟

۱) هنگامی که سیستم نیاز به مدیریت همزمان تعداد زیادی تراکنش مالی با قابلیت rollback داشته باشد.

۲) در سیستم هایی که تصمیم گیری ها مبتنی بر رویدادها و وضعیت های داخلی است.

۳) هنگامی که داده های ورودی باید از طریق مجموعه ای از مؤلفه های محاسباتی و پردازشی به داده های خروجی تبدیل شوند.

۴) هنگامی که طراحی باید حول یک پایگاه داده متمرکز باشد و دسترسی های همزمان چندین کاربر با سطوح دسترسی متفاوت را مدیریت کند.

۳۸- کدام یک از مراحل طراحی ساختار برنامه به طور خاص مربوط به گذار از نمودار جریان داده ها (DFD) به ساختار برنامه است؟

- ۱) تعیین نوع جریان اطلاعات
 - ۲) تعریف سلسله مراتب کنترلی
 - ۳) مشخص کردن مرزهای جریان اطلاعات
 - ۴) نگاشت DFD به یک ساختار برنامه ای
- ۳۹- کدام یک از ویژگی های زیر از مزایای اصلی پالایش معماری نرم افزار به شمار می رود؟

- ۱) ارزیابی و انتخاب بهترین سبک های معماری برای هر پروژه
 - ۲) بهینه سازی پیچیدگی کدنویسی
 - ۳) استفاده از بیشتر مؤلفه ها برای افزایش انعطاف پذیری سیستم
 - ۴) ایجاد ساختار داده ای پیچیده برای پاسخگویی بهتر به خواسته های اطلاعاتی
- ۴۰- نقشهای مهمی که مؤلفه های سنتی (پیمانه ها) در معماری نرم افزار ایفا می کنند، کدامند؟

- ۱) مؤلفه داده، مؤلفه واسط، مؤلفه پردازش
 - ۲) مؤلفه کنترلی، مؤلفه دامنه مسئله، مؤلفه زیرساختی
 - ۳) مؤلفه ورودی، مؤلفه خروجی، مؤلفه ذخیره سازی
 - ۴) مؤلفه ساختاری، مؤلفه منطقی، مؤلفه اجرایی
- ۴۱- بر اساس اصل هم ارزی استفاده ی مجدد از نسخه ها (REP)، برای مدیریت و کنترل تکامل نسخه های جدید نرم افزار چه راهکاری توصیه می شود؟ - ناشر ایران عرضه -

- ۱) گروه بندی کلاس های قابل استفاده مجدد در قالب بسته ها
- ۲) طراحی مجدد هر کلاس به صورت مستقل در هر نسخه
- ۳) حذف پشتیبانی از نسخه های قدیمی و تمرکز کامل بر نسخه های جدید
- ۴) محدود کردن استفاده مجدد فقط به مؤلفه های کنترلی

۴۲- اتصال محتوا در مهندسی نرم افزار در چه حالتی رخ می دهد؟

- ۱) زمانی که دو مؤلفه از طریق واسط های تعریف شده با یکدیگر تبادل داده می کنند.
- ۲) زمانی که یک مؤلفه داده ها را برای خواندن از مؤلفه دیگر دریافت می کند.

۳) زمانی که یک مؤلفه به طور مستقیم داده های داخلی مؤلفه دیگر را اصلاح می کند که این کار نقض اصل پنهان سازی اطلاعات است.

۴) زمانی که چند مؤلفه از یک ساختار داده مشترک و عمومی استفاده می کنند.

۴۳- چه زمانی اشیاء محتوایی در برنامه های تحت وب باید به صورت مؤلفه های سازمان دهی شده پیاده سازی شوند؟

۱) وقتی محتوا پویا و روابط پیچیده ای داشته باشد.

۲) وقتی برنامه کوچک و ساده باشد.

۳) وقتی اشیاء محتوایی نیازی به تغییر نداشته باشند.

۴) وقتی محتوا ثابت باشد.

۴۴- مهندسی دامنه چه هدفی دارد؟

۱) تولید نرم افزارهای کاملاً اختصاصی برای امکان استفاده ی مجدد در فرآیند توسعه.

۲) توزیع مؤلفه های نرم افزاری و تمرکز بر تحلیل نیازمندی ها بدون ساخت و انتشار مؤلفه ها.

۳) افزایش وابستگی میان مؤلفه ها برای ساده سازی فرآیند توسعه.

۴) شناسایی، پیاده سازی و توزیع مؤلفه های نرم افزاری برای استفاده در سیستم های موجود و جدید.

۴۵- کتابخانه استفاده مجدد در چه مواردی کاربرد دارد؟

۱) نگهداری موقت کدهای خام و ناقص در طول فرآیند توسعه، بدون پشتیبانی از طبقه بندی، مستندسازی و امکان بازیابی هدمند مؤلفه ها

۲) مستندسازی نیازمندی ها و طراحی های اولیه ی نرم افزار به صورت ایستا، بدون فراهم کردن سازوکار ذخیره سازی و استفاده ی مجدد از مؤلفه ها و محصولات کاری

۳) ذخیره سازی، طبقه بندی و بازیابی مؤلفه های نرم افزاری و محصولات کاری قابل استفاده ی مجدد مانند الگوها، چارچوب ها، کد، موارد آزمون و راهنماها در یک مخزن نرم افزاری

۴) اجرای مستقیم مؤلفه های نرم افزاری بدون نیاز به الگوی طبقه بندی، بانک اطلاعاتی و ابزارهای ارجاع و بازیابی

۴۶- چه رابطه ای میان بار حافظه کاربر و احتمال خطای او در تعامل با سیستم وجود دارد؟

۱) هرچه کاربر مجبور به حفظ جزئیات بیشتری باشد، احتمال خطای او کاهش می یابد.

۲) هرچه کاربر مجبور به حفظ جزئیات بیشتری باشد، احتمال خطای او افزایش می یابد.

۳) بار حافظه کاربر تأثیری بر احتمال خطای او ندارد.

۴) بار حافظه کاربر به طور مستقیم با دقت عملکرد او ارتباط دارد.

۴۷- تحلیل جریان کاری به چه چیزی اشاره دارد؟ {ایران عرضه}

۱) شناسایی اشیای رابط کاربر و ویژگی های آن ها

(۲) بررسی وظایف یک کاربر منفرد در تعامل با سیستم

(۳) بررسی روند انجام یک فرایند توسط چند کاربر با نقش های متفاوت

(۴) پیاده سازی فنی و کدنویسی فرایندهای سیستم

۴۸- کدام دو مورد از ویژگی های مهم پاسخ سیستم هستند؟

(۱) دقت و قابلیت اطمینان (۲) سرعت پردازش و ظرفیت سیستم

(۳) یکنواختی رابط کاربر و سادگی طراحی (۴) طول پاسخ و تغییرپذیری زمان پاسخ

۴۹- شناسایی اشیای واسط کاربر در طراحی واسط برنامه ی تحت وب مستلزم چه فعالیتی است؟

(۱) تعیین معماری کلی سیستم، شناسایی کلاس های سفارشی و نحوه ی ارتباط مؤلفه ها

(۲) طراحی پایگاه داده و مشخص کردن ساختار ذخیره سازی اطلاعات

(۳) جستجو در کتابخانه ی اشیای موجود برای یافتن کلاس های قابل استفاده ی مجدد واسط و شناسایی کلاس های سفارشی

مورد نیاز

(۴) پیاده سازی منطق پردازش و کنترل جریان برنامه

۵۰- هدف از ارزیابی نمونه ی اولیه عملیاتی واسط کاربر چیست؟

(۱) شناسایی خطاهای فنی در کدنویسی سیستم.

(۲) بررسی میزان تطابق واسط کاربر با استانداردهای برنامه نویسی.

(۳) اطمینان از این که واسط کاربر نیازهای کاربران را برآورده می سازد یا خیر.

(۴) افزایش پیچیدگی واسط کاربر برای بهبود کارایی سیستم.