



Raik Pöhl

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung
Abschlussprüfung Teil 2



Best.-Nr. 75721
2. Auflage 2024

2. Auflage 2024

Best.-Nr. 75721 · ISBN 978-3-95532-751-4

Alle Rechte liegen beim Verlag bzw. sind der Verwertungsgesellschaft Wort, Untere Weidenstr. 5, 81543 München, Telefon 089 514120, zur treuhänderischen Wahrnehmung überlassen. Damit ist jegliche Verbreitung und Vervielfältigung dieses Werkes – durch welches Medium auch immer – untersagt.



© u-form Verlag | Hermann Ullrich GmbH & Co. KG
Cronenberger Straße 58 | 42651 Solingen
Telefon: 0212 22207-0 | Telefax: 0212 22207-63
Internet: www.u-form.de | E-Mail: uform@u-form.de





Achtung!

Sollte es für diese Lernkarten Korrekturen oder Änderungen geben, kannst du diese herunterladen unter

www.u-form.de/addons/75721-2024.pdf

Ist dieser Link nicht verfügbar, so haben wir noch keine Änderungen, Korrekturen oder Zusatzinfos hinterlegt.

Hinweis: Auf Lernkarten ist der Platz begrenzt und die hier beschriebenen Themen sind kompliziert. Auch wenn mehrheitlich von Kunden, Nutzern oder Administratoren die Rede ist, sind stets alle möglichen Geschlechter gemeint.

Frage

Beschreiben Sie den Unterschied zwischen Zugangskontrolle und Zugriffskontrolle.

Das Konzept der **Zugangskontrolle** (Access Control) bezieht sich auf die Möglichkeit, den physischen oder virtuellen Zugang zu einer Ressource zu kontrollieren. Die Zugangskontrolle bestimmt, wer oder was Zugang zu einer Ressource hat.

Die **Zugriffskontrolle** (Authorization Control) regelt, was ein Benutzer oder System nach dem Zugang machen darf. Nachdem jemand Zugang zu einer Ressource erlangt hat, kann die Zugriffskontrolle festlegen, welche Operationen er auf dieser Ressource ausführen darf. Sie regelt die Befugnisse oder Berechtigungen, die einem Benutzer oder System gewährt werden, nachdem der Zugang gewährt wurde.

Frage

Was ist in Bezug auf die Systemsicherheit unter dem Begriff der „Betriebssystemhärtung“ zu verstehen?

Betriebssystemhärtung bezieht sich auf eine Reihe von Maßnahmen, die darauf abzielen, die Sicherheit eines Betriebssystems zu erhöhen, indem dessen Anfälligkeiten reduziert und potenzielle Angriffspunkte minimiert werden.

Beispiele für wirksame Maßnahmen sind:

- nur die absolut notwendigen Softwarekomponenten und -dienste installieren
- Patch-Management
- Einschränkung von Systemzugriffsrechten
- Aktivieren der verfügbaren Sicherheitsfunktionen
- Nutzung von Firewall und Antivirensoftware
- Verwendung von Verschlüsselung

Frage

Was besagt die 3-2-1-Regel in Bezug auf die Datensicherung?

Die **3-2-1-Regel** stellt sicher, dass die Daten sicher aufbewahrt werden. Sie besagt folgendes:

- Es sollten mindestens **drei** Kopien der Daten aufbewahrt werden. Das schließt die ursprüngliche Datei und zwei Backup-Kopien ein.
- Es sollten mindestens **zwei** verschiedene Speichermedien für die Backups verwendet werden.
- Mindestens **eine** dieser Kopien sollte an einem anderen Ort aufbewahrt werden, um sich gegen lokale Katastrophen wie Feuer, Überschwemmungen oder Einbrüche zu schützen. Diese „Offsite“-Kopie kann entweder in einem physisch entfernten Ort oder in einem Cloud-Speicherdienst aufbewahrt werden.

Frage

Erklären Sie anhand eines Beispiels, warum ein RAID keine Datensicherung ersetzt.

Ein RAID schützt in den folgenden Fällen nicht vor einem Datenverlust:

- Virenbefall eines Systems
- versehentlich gelöschte oder veränderte Daten
- physikalische Beschädigung durch Feuer, Wasser oder Vandalismus

Frage

Digitale Signaturen stützen sich auf Hashverfahren und basieren auf der asymmetrischen Kryptografie.

Erklären Sie diesen Zusammenhang.

Bei **digitalen Signaturen** wird die asymmetrische Kryptografie umgekehrt angewandt. Die Signatur wird vom Sender einer Nachricht mit Hilfe seines privaten Schlüssels erzeugt. Vom Empfänger kann die Korrektheit einer Signatur über den öffentlichen Schlüssel des Senders überprüft werden.

Um die Effizienz des Signaturverfahrens zu steigern, wird die digitale Signatur nur auf dem Hashwert einer Nachricht erzeugt.

Frage

Was versteht man unter einem Code Review?

Ein **Code Review** ist eine systematische Untersuchung von ⁶⁹ Programmcode und wird den statischen Testverfahren zugeordnet. Hierbei kann der Code unter anderem auf folgende Aspekte geprüft werden:

- Funktionalität
- Lesbarkeit und Verständlichkeit
- Wartbarkeit
- Best Practices und Standards

Nachteile:

- Code Reviews erfordern Zeit und Ressourcen
- Code Reviews können subjektiv sein, da die Meinungen der Reviewer variieren können
- der Programmcode wird nicht ausgeführt

Frage

Welche Schritte sind notwendig, um Datenstrukturen in einem ER-Diagramm zu modellieren?

Ein **Entity-Relationship-Diagramm** (ER-Diagramm) wird zur Modellierung von Datenbanken wie folgt verwendet:

1. Identifizieren der Entitäten: Das sind die Hauptobjekte oder Konzepte in der Datenbank.
2. Bestimmen der Attribute: Das sind die Eigenschaften oder Merkmale der Entitäten.
3. Identifizieren der Beziehungen: Das sind die Verbindungen und Interaktionen zwischen Entitäten.
4. Bestimmen der Kardinalität: Das definiert, wie viele Instanzen einer Entität mit einer Instanz einer anderen Entität in Beziehung stehen können.

Frage

Wie funktionieren Trigger in Datenbankumgebungen und wie können sie zur automatischen Ausführung von Aktionen verwendet werden?

Ein **Trigger** in einer Datenbank ist eine Art von Stored Procedure, die automatisch in Reaktion auf bestimmte Ereignisse in einer spezifischen Tabelle oder Ansicht in der Datenbank ausgeführt wird. Die Ereignisse können das Einfügen, Aktualisieren oder Löschen von Daten sein.

Ein Trigger wird immer mit einer Tabelle verknüpft und aktiviert sich, wenn eine angegebene Datenmanipulationsoperation (DML) ausgeführt wird.

Frage

Was ist der Unterschied zwischen einem Compiler und einem Interpreter?

Ein **Compiler** ist ein Programm, das den gesamten Quellcode einer Programmiersprache analysiert und in einen ausführbaren Maschinencode übersetzt. Der übersetzte Code kann später unabhängig vom Compiler ausgeführt werden.

Ein **Interpreter** ist ein Programm, das den Quellcode einer Programmiersprache Zeile für Zeile ausführt. Im Gegensatz zum Compiler, der den gesamten Code vor der Ausführung übersetzt, übersetzt der Interpreter den Code zur Laufzeit. Der Interpreter analysiert und führt jede Anweisung einzeln aus, wodurch eine direkte Interpretation des Codes ermöglicht wird.

Frage

Wozu dient die DTD (Dokumenttyp-Definition) bei der Verwendung von XML-Dateien?

Die **Dokumenttyp-Definition** (DTD) ist eine Möglichkeit, die Struktur und den Inhalt von XML-Dokumenten zu definieren. Sie dient dazu, Regeln und Beschränkungen für die Elemente, Attribute und deren Beziehungen in einem XML-Dokument festzulegen.

Die DTD stellt eine formale Beschreibung bereit, die es ermöglicht, die Gültigkeit und Konformität eines XML-Dokuments zu überprüfen.