

LFK Prozedurale Programmierung

Frage 1 von 40: Abhängig von der Anzahl der Argumente, auf die ein Operator wirkt, unterscheidet man unäre, binäre und ternäre Operatoren.

Welche Aussagen treffen zu?

- 1) Keine der übrigen Aussagen ist richtig.
- 2) Ternäre Operatoren liefern abhängig von einer Fallunterscheidung unterschiedliche Werte.
- 3) Unäre Operatoren sind in der Regel kommutativ.
- 4) Binäre Operatoren können nur bitweise angewendet werden.
- 5) In vielen prozeduralen Programmiersprachen werden für unäre und binäre Operatoren dieselben Symbole verwendet.

Frage 2 von 40: Welche Bestandteile umfasst der Sprachumfang einer prozeduralen Programmiersprache im allgemeinen?

- 1) Variable,
- 2) Operatoren,
- 3) Keine der übrigen.
- 4) Ein- und Ausgabe,
- 5) Kontrollstrukturen,

Frage 3 von 40: Welche Aussagen bezüglich der unbedingten Sprunganweisung ("Goto") treffen zu?

- 1) Die häufige Verwendung dieser Anweisung kennzeichnet guten Programmierstil.
- 2) Die Anweisung sollte nur für besondere Zwecke genutzt werden, zum Beispiel die Behandlung von Sondersituationen.
- 3) Die Anweisung sollte sparsam verwendet werden, weil durch sie der Programmfluss unübersichtlich werden kann.
- 4) Keine von den übrigen.
- 5) Die Anweisung kann beliebig verwendet werden, weil sie auf Prozessebene ebenfalls vorhanden ist.

Frage 4 von 40: In prozeduralen Programmiersprachen können literale Ausdrücke formuliert werden. Welche Aussagen treffen zu?

- 1) Literale Ausdrücke können - genau wie Ausdrücke mit Variablen - erst zur Laufzeit ausgewertet werden. Sie erfordern daher zusätzliche Laufzeit und sollten daher möglichst vermieden werden.
- 2) Literale Ausdrücke sollten möglichst vom Programmierer (beispielsweise unter Verwendung von Taschenrechnern) evaluiert werden, um kürzere Quelltexte zu erhalten.
- 3) Literale Ausdrücke werden während der Übersetzung ausgewertet. Ihre Evaluation erfordert also keine zusätzliche Laufzeit.
- 4) Literale Ausdrücke können die Klarheit der Programmierung erhöhen, indem sie das Entstehen ihrer Werte beschreiben.
- 5) Keine der übrigen Aussagen trifft zu.

Frage 5 von 40: Welche Informationen sollten in Quelltextkommentaren auftreten?

- 1) Der Urheber des Kommentars.
- 2) Angaben zu Beweggründen für die Art und Weise der Programmierung.
- 3) Informationen zu etwaigen Quereffekten.
- 4) Der Erstellungszeitpunkt des Kommentars.
- 5) Keine von den übrigen.

Frage 6 von 40: Die meisten prozeduralen Sprachen kennen Schlüsselwörter. Was gilt in den meisten Sprachen bezüglich dieser Schlüsselwörter?

- 1) Die Verwendung von Schlüsselwörtern als Bezeichnernamen ist erlaubt.
- 2) Der Umfang der Schlüsselwörter ist definiert.
- 3) Keine der übrigen Antworten trifft zu.
- 4) Schlüsselwörter dienen zum Verschlüsseln vertraulicher Informationen.
- 5) Der Programmierer kann eigene Schlüsselwörter ergänzen.

Frage 7 von 40: In der prozeduralen Programmierung findet ein Teil der dynamischen Speicherverwaltung über den Stack statt. Welche Aussagen sind korrekt?

- 1) Beim Aufruf von Unterprogrammen werden die Rücksprungadressen in der Regel auf dem Stack abgelegt.
- 2) In der Regel werden Funktionsargumente auf dem Stack abgelegt.
- 3) Bei rekursiven Prozeduren erfolgt die Verwaltung von Zwischenwerten in der Regel auf dem Stack.
- 4) Keine der übrigen Aussagen ist korrekt.
- 5) Globale Variable werden in der Regel im Stack gespeichert.

Frage 8 von 40: Warum verwendet man in prozeduralen Programmen symbolische Konstante?

- 1) Das ausführbare Programm wird kleiner.
- 2) Veränderung und Anpassung der Konstantenwerte werden erheblich erleichtert.
- 3) Literale können dadurch sprechende Namen erhalten, was die Verständlichkeit des Programms erhöht.
- 4) Aus keinem der übrigen Gründe.
- 5) Die Compilation wird beschleunigt.

Frage 9 von 40: Was gilt bezüglich der Syntax einer prozeduralen Programmiersprache?

- 1) Sie definiert, wie gültige Anweisungen aufgebaut werden.
- 2) Sie umfasst die Lexikalik der Sprache.
- 3) Sie legt den Aufbau von gültigen Programmen fest.
- 4) Sie legt fest, aus welchen Konstrukten Programme bestehen können.
- 5) Keine der übrigen Antworten trifft zu.

Frage 10 von 40: Warum sollten in prozeduralen Programmen Kommentare verwendet werden?

- 1) In Kommentaren kann man dem Compiler Hinweise zur Optimierung geben.
- 2) Durch den erhöhten Quelltextumfang wirkt das Programm mächtiger.
- 3) Aus keinem der übrigen Gründe.
- 4) Die Wartbarkeit wird verbessert.
- 5) Zur Dokumentation.

Frage 11 von 40: Manche prozeduralen Programmiersprachen enthalten ternäre Operatoren. Welche der folgenden Aussagen treffen zu?

- 1) Ternäre Operatoren liefern genau einen von drei möglichen Werten zurück.
- 2) Ternäre Operatoren liefern drei Werte zurück.
- 3) Ternäre Operatoren haben gegenüber unären und binären stets den höheren Vorrang.
- 4) Keine der übrigen Aussagen trifft zu.
- 5) Ternäre Operatoren haben drei Argumente.

Frage 12 von 40: In einigen prozeduralen Programmiersprachen ist eine Funktionsdeklaration möglich. Welche Aussagen sind in diesem Zusammenhang korrekt?

- 1) Die Funktionsdeklaration erlaubt dem Compiler eine formale Prüfung der korrekten Verwendung der Funktion.
- 2) Die Funktionsdeklaration erlaubt dem Compiler eine formale Prüfung der korrekten Verwendung der Funktion.
- 3) Die Funktionsdeklaration beinhaltet eine implizite Funktionsdefinition.
- 4) Keine der übrigen Aussagen ist korrekt.
- 5) Die Funktionsdeklaration legt den Rückgabewert der Funktion fest.

Frage 13 von 40: Im Zuge der Ausgabe verwendet man in den meisten prozeduralen Programmiersprachen Funktionen, die eine Formatierung vornehmen. Welche Aussagen sind richtig?

- 1) Durch die Ausgabeformatierung wird festgelegt, welches Ausgabemedium zu verwenden ist.
- 2) Die Ausgabeformatierung erfolgt in der Regel entsprechend einer Formatbeschreibung.
- 3) Viele prozedurale Programmiersprachen bieten die Mechanismen der Ausgabeformatierung auch als so genannte "interne Ausgabe" an, um formatierte Zeichenketten zu erhalten.
- 4) Formatbeschreibungen müssen immer bereits zur Kompilationszeit festliegen.
- 5) Keine der übrigen Aussagen ist richtig.

Frage 14 von 40: Variable müssen in vielen prozeduralen Programmiersprachen zwar deklariert, nicht aber definiert werden. Welche Auswirkungen kann der Verzicht auf eine Variablendefinition haben?

- 1) Keine der übrigen Auswirkungen kann auftreten.
- 2) Wird der Variableninhalt ohne vorherige Definition verwendet, dann kann ihr Wert unbestimmt sein, was sich auf das Resultat der Verwendung auswirken kann.
- 3) Eine nicht definierte Variable ist unbekannt und kann daher nicht benutzt werden.
- 4) Wenn eine Variable nicht definiert wird, dann wird für sie auch kein Speicherplatz reserviert.
- 5) Der Verzicht auf Definition ist unproblematisch, wenn zur Laufzeit die erste Verwendung der Variablen in einer Zuweisung besteht.

Frage 15 von 40: Welche Aufgaben hat ein Compiler?

- 1) Übersetzen des Quelltextes in Maschinencode.
- 2) Keine von den übrigen.
- 3) Prüfen auf formal korrekte Syntax des Quelltextes.
- 4) Abfangen von Laufzeitfehlern.
- 5) Übersetzen des Quelltextes in andere Hochsprachen.

Frage 16 von 40: Welche der folgenden Systeme können zur Versionsverwaltung der Quelltexte von prozeduralen Programmen verwendet werden?

- 1) Subversion.
- 2) Keines von den übrigen.
- 3) CVS
- 4) Revolte.
- 5) Anarchie.

Frage 17 von 40: Welche Aufgaben hat ein Profiler?

- 1) Laufzeitanteile einzelner Unterprogramme ermitteln.
- 2) Häufigkeiten von Funktionsaufrufen zur Laufzeit bestimmen.
- 3) Keine von den übrigen.
- 4) Stilprägnanz des Quelltextes verbessern.
- 5) Bösen Code aufspüren.

Frage 18 von 40: Welche Wirkung haben optimierende Compiler?

- a) Sie erzeugen schnellere Programme.
- b) Sie bewirken, dass die erzeugten Programme vom Betriebssystem bevorzugt bearbeitet werden.
- c) Sie erzeugen in der Regel kleinere Programme.
- d) Sie benötigen in der Regel eine längere Zeit zur Compilation.
- e) Keine von den übrigen.

Frage 19 von 40: Prozedurale Programmiersprachen bieten Funktionen und Operatoren an. Welche Gemeinsamkeiten besitzen sie?

- 1) Funktionen und Operatoren können unter bestimmten Umständen ihre Argumente manipulieren.
- 2) Funktionen und Operatoren können Seiteneffekte aufweisen.
- 3) Operatoren und Funktionen besitzen keine Gemeinsamkeiten.
- 4) Die Argumente beziehungsweise Operanden müssen stets in Klammern geschrieben werden.
- 5) Funktionen und Operatoren liefern einen Wert.

Frage 20 von 40: Welche Aufgaben haben Quelltextformatierer (Beautifier wie z.B. indent) bei der Anwendung auf prozedurale Programme?

- 1) Passende Schriftarten für Schlüsselwörter, Bezeichner usw. wählen.
- 2) Lesbarkeit verbessern
- 3) Einheitliches Layout von Einrückung und Syntaxelementen einhalten.
- 4) Keine von den übrigen.
- 5) Programmierer beim Entwickeln ihres individuellen Programmierstils unterstützen.

Frage 21 von 40: Datenstrukturen - welche Aussagen treffen zu?

- 1) Keine von den übrigen.
- 2) Komplexere Datenstrukturen werden aus elementaren Typen aufgebaut.
- 3) Auf Elemente von Feldern kann über Indizes zugegriffen werden, deren Werte zur Laufzeit als Ausdrücke berechenbar sind.
- 4) Datenstrukturen sind unübersichtlich und ihre Verwendung erschwert die Lesbarkeit von Programmen.
- 5) Häufig verwendete Datenstrukturen sind Felder oder Arrays.

Frage 22 von 40: Die Compiler von prozeduralen Programmiersprachen arbeiten in mehreren Stufen. Welche Aussagen sind in diesem Zusammenhang korrekt?

- 1) Während der syntaktischen Analyse wird der syntaxgerechte Aufbau des Quelltextes überprüft.
- 2) Optional kann eine Optimierungsstufe vorgesehen werden.
- 3) Im Zuge der lexikalischen Analyse werden die sprachlichen Elemente des Quelltextes (Token) identifiziert und klassifiziert.
- 4) Keine der übrigen Aussagen ist korrekt.
- 5) Das ausführbare Programm entsteht durch das Generieren des Bytecodes.

Frage 23 von 40: Einige prozedurale Programmiersprachen erlauben die implizite Deklaration von Variablen. Was bedeutet dies?

- 1) Die Verwendung impliziter Deklaration verhindert die formale Prüfbarkeit der korrekten Variablenverwendung.
- 2) Der Typ dieser Variablen bleibt unbestimmt.
- 3) Diese Variablen werden erst zur Laufzeit initialisiert.
- 4) Keine der übrigen Aussagen trifft zu.
- 5) Diese Variablen werden im Zuge ihrer ersten Verwendung im Quelltext angelegt.

Frage 24 von 40: In welcher Reihenfolge werden Programmierwerkzeuge genutzt?

- 1) Linker, Compiler, Editor.
- 2) Linker, Debugger, Compiler.
- 3) In keiner von den übrigen Reihenfolgen.
- 4) Editor, Debugger, Linker.
- 5) Editor, Compiler, Linker.

Frage 25 von 40: In Quelltexten prozeduraler Programme dürfen nur zulässige Zeichen verwendet werden. Welche Aussagen treffen in diesem Zusammenhang zu?

- 1) Leerzeichen können immer nach Belieben hinzugefügt oder fortgelassen werden.
- 2) Im Quelltext können manchmal auch unsichtbare Zeichen (zum Beispiel das Zeilenende) bedeutsam sein.
- 3) In literalen Zeichenketten und in Kommentaren sind in der Regel weit mehr Zeichen erlaubt als im übrigen Quelltext.
- 4) Die Bedeutung eines Zeichens ist vom Kontext abhängig. So kann beispielsweise das Zeichen "-" den Subtraktionsoperator oder aber den Negationsoperator bezeichnen.
- 5) Keine der übrigen Aussagen trifft zu.

Frage 26 von 40: In prozeduralen Programmen sind häufig Typwandlungen (type casts) erforderlich. Welche Aussagen treffen zu?

- 1) Man unterscheidet explizite und implizite Typwandlungen.
- 2) Datenwerte können bei Typwandlungen unter Umständen eine erhebliche Wertveränderung erfahren.
- 3) Bei Typwandlungen kann es zu Genauigkeitsverlusten kommen.
- 4) Keine der übrigen Aussagen trifft zu.
- 5) Typwandlungen verursachen Laufzeitfehler.

Frage 27 von 40: Welche Aussagen über die Auswertungsreihenfolge von Ausdrücken in prozeduralen Programmiersprachen treffen zu?

- 1) Wenn in einer Programmiersprache die Punkt-vor-Strich-Regel gilt, ergibt sich dies aus der Präzedenz der arithmetischen Operatoren.
- 2) Die Auswertungsreihenfolge hängt von Präzedenz und Assoziativität ab.
- 3) Keine der übrigen Aussagen trifft zu.
- 4) Wenn in einer Programmiersprache die Punkt-vor-Strich-Regel gilt, dann ist für eine Summe von zwei Produkten die zeitliche Abfolge der Auswertung der Produkte in der Regel nicht vorhersagbar.
- 5) Ausdrücke werden grundsätzlich von links nach rechts ausgewertet.

Frage 28 von 40: Was gilt bezüglich der Phonetik einer prozeduralen Programmiersprache?

- 1) Keine der übrigen Antworten trifft zu.
- 2) Sie legt fest, bei welchem maximalen Hintergrundgeräusch prozedurale Programme der jeweiligen Sprache entwickelt werden können.
- 3) Sie legt die Lautstärke des Piepers fest, mit dem auf Fehler hingewiesen wird.
- 4) Sie legt die Aussprache der Wörter der Sprache fest.
- 5) Eine Phonetik ist für keine prozedurale Sprache festgelegt.

Frage 29 von 40: In der Anwendungsentwicklung müssen prozedural erstellte Programme vielfach Betriebssystemfunktionen in Anspruch nehmen. Welche gehören dazu?

- 1) Funktionen der dynamischen Speicherverwaltung,
- 2) Funktionen der Ein- und Ausgabe,
- 3) keine der übrigen.
- 4) Funktionen zur Interaktion mit anderen Programmen,
- 5) Funktionen zur Bestimmung von Typgrößen,

Frage 30 von 40: In vielen prozeduralen Programmiersprachen ist es möglich, Prototypen von Funktionen zu deklarieren. Welche Aussagen sind richtig?

- 1) Prototypen deklarieren lediglich den Aufruf der Funktion.
- 2) Die Typprüfung der Argumente findet zur Laufzeit statt.
- 3) Prototypen erlauben es dem Compiler, Anzahl und Typ der Argumente zu überprüfen.
- 4) Der Prototyp einer Funktion ist eine vorläufige Fassung des Quelltextes.
- 5) Keine der übrigen Aussagen ist richtig.

Frage 31 von 40: Bei vielen prozeduralen Programmiersprachen gestaltet sich die Behandlung von Feldern (Arrays) problematisch, wenn deren Größe erst zur Laufzeit bekannt ist. Welche Probleme treten auf?

- 1) Keines der übrigen Probleme tritt auf.
- 2) Die Verwendung von Programmen, die Felder erst zur Laufzeit anlegen, erfordert den zusätzlichen Einbau von Speicher in den Rechner.
- 3) Die Bereitstellung einer passenden Feldgröße zur Laufzeit erfolgt durch das Betriebssystem, dessen Eigenschaften berücksichtigt werden müssen. Dies verringert die Portabilität des Programms.
- 4) Die Feldgröße kann in manchen Fällen zur Übersetzungszeit abgeschätzt werden. In der Regel ist die Felddimensionierung dann zu groß. Dies führt dies zur Speicherplatzverschwendung während der Laufzeit.
- 5) Oft kann die Feldgröße nur ungefähr geschätzt werden. So kann das Feld zur Laufzeit zu klein sein. Das Problem ist dann ohne Anpassung des Quelltextes und neuerliche Übersetzung nicht lösbar.

Frage 32 von 40: Die meisten prozeduralen Programmiersprachen bieten außer einfachen Datentypen auch aggregierte Datentypen an. Welche der folgenden Aussagen ist korrekt?

- 1) Aggregierte Datentypen werden aus einfachen Datentypen aufgebaut.
- 2) Komponenten aggregierter Datentypen können selbst aggregierte Datentypen sein.
- 3) Aggregierte Datentypen müssen immer aus Komponenten identischen Datentyps aufgebaut sein.
- 4) Keine der übrigen Aussagen ist korrekt.
- 5) Der Speicherbedarf eines aggregierten Datentyps ergibt sich aus der Anzahl der Komponenten und deren jeweiligem Typgröße sowie etwaigem zusätzlichem Alignment.

Frage 33 von 40: Was gilt für Typen von Variablen und Literalen?

- 1) Der exakte Speicherbedarf für einen Typ ist grundsätzlich durch den Standard der Programmiersprache festgelegt.
- 2) In den meisten Programmiersprachen existieren elementare Typen wie Ganzzahlige und Fließpunkt.
- 3) Typwandlung kann mit Informationsverlust behaftet sein.
- 4) Alle übrigen Aussagen sind falsch.
- 5) Bei der Verarbeitung von Werten unterschiedlicher Typen muss gegebenenfalls eine implizite oder explizite Typwandlung erfolgen.

Frage 34 von 40: In welchen Situationen ist die Verwendung prozeduraler Programmierung nachteilig?

- 1) Bei verteilter Softwareentwicklung.
- 2) Bei großen Softwareprojekten.
- 3) Bei der Erstellung von Betriebssystemen.
- 4) In keiner von den übrigen.
- 5) Bei komplexer Ereignisverarbeitung.

Frage 35 von 40: Viele prozedurale Programmiersprachen bieten die Verwendung kopfgesteuerter Schleifen an. Welche Aussagen treffen zu?

- 1) Der Schleifenkörper wird mindestens einmal durchlaufen.
- 2) Kopfgesteuerte Schleifen können Endlosschleifen sein.
- 3) Keine der übrigen Aussagen trifft zu.
- 4) Der Test erfolgt vor dem Eintritt in den Schleifenkörper.
- 5) Kopfgesteuerte Schleifen liefern immer ein korrektes Verarbeitungsergebnis.

Frage 36 von 40: Der Begriff der "Deklaration" bezeichnet ein grundlegendes Konzept in der prozeduralen Programmierung. Welche Aussagen treffen zu?

- 1) Eine Deklaration ist eine spezielle Willensbekundung, die der Programmierer in einem gesonderten Kommentar abgibt.
- 2) Eine Deklaration legt den konkreten Wert des Bezeichneten fest.
- 3) Mittels Deklaration wird dem Benutzer die Anwendung des Programms erläutert.
- 4) Keine der übrigen Aussagen ist korrekt.
- 5) Eine Deklaration entspricht einer Vereinbarung zur Festlegung von Namen und deren Typ.

Frage 37 von 40: Die meisten prozeduralen Programmiersprachen erlauben die binäre Ein- und Ausgabe von Daten. Was ist zu beachten, wenn diese Daten von Programmen verarbeitet werden sollen, die in verschiedenen Programmiersprachen erstellt sind und möglicherweise auf unterschiedlichen Plattformen laufen?

- 1) Die interne Kodierung binärer Daten (zum Beispiel ihre Endianess) ist von der Plattform abhängig.
- 2) Typgrößen und Kodierung von Binärdaten sind international durch das Völkerrecht festgelegt. Deshalb sollten Daten bevorzugt im Binärformat ein- und ausgegeben werden.
- 3) Die Typgröße im Sinne des Speicherplatzes ist abhängig von der Programmiersprache.
- 4) Die Ein- und Ausgabe von Daten im Binärformat erfolgt in der Regel schneller als im Textformat.
- 5) Keiner der übrigen Sachverhalte ist gesondert zu beachten.

Frage 38 von 40: In vielen prozeduralen Programmiersprachen können Felder (Arrays) gebildet werden, deren Komponenten über ganzzahlige Indizes angesprochen werden. Welche Aussagen dazu sind richtig?

- 1) Eine Bereichsüberprüfung des Index auf gültige Feldgrenzen findet zur Übersetzungszeit statt.
- 2) Ein Feld enthält immer auch die Information über seine eigenen Dimensionen.
- 3) Keine der übrigen Aussagen ist richtig.
- 4) Der Index kann der Wert eines Ausdrucks sein und damit seinerseits Feldelemente enthalten.
- 5) Die Komponenten eines Feldes liegen geordnet im Speicher.

Frage 39 von 40: Welche Aussagen bezüglich Rekursion sind zutreffend?

- 1) Rekursiv programmierte Prozeduren weisen oft relativ lange Laufzeiten auf.
- 2) Rekursion lässt sich immer auf Iteration zurückführen.
- 3) Rekursive Prozeduren lassen sich oft elegant und einfach programmieren.
- 4) Die übrigen Aussagen sind sämtlich falsch.
- 5) Rekursion bewirkt häufig relativ großen Speicherbedarf.

Frage 40 von 40: Welche der folgenden Begriffe bezeichnen Sprachelemente, die in prozeduralen Programmen auftreten können?

- 1) Einweisung,
- 2) Keiner der übrigen.
- 3) Anweisung,
- 4) Überweisung,
- 5) Zuweisung,