

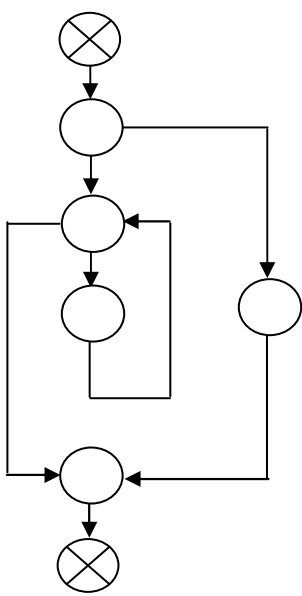
**Frage 1.**

	<b>Welcher der folgenden Testfallentwürfe ist ein Beispiel für die Anwendung des Fehlerangriff-Testverfahren?</b>	
<b>A</b>	Ausführen der Berechnung mit 100 gültigen Datensätzen. Messen der Bearbeitungszeit unter nominaler und unter maximaler Last. Um aussagekräftige Werte zu ermitteln, muss dieser Test 3 mal wiederholt werden.	
<b>B</b>	Beta-Test mit mindestens 24 repräsentativen Benutzern. Dabei muss jeder Benutzer mindestens einmal die Berechnung mit 100 gültigen Datensätzen durchführen.	
<b>C</b>	Ausführen der Berechnung mit 100 gültigen Datensätzen. Dabei soll die Betriebssystemfunktion "Öffnen einer Datei" beim x-ten (konfigurierbar) Aufruf keine Antwort mehr liefern (d.h. bleibt "hängen").	
<b>D</b>	Ausführen der Berechnung mit 100 gültigen Datensätzen. Stichprobenartige Prüfung der Resultate. Es sollen mindestens 12 Ausgabedaten geprüft werden.	

**Frage 2.**

	<b>Welche zwei der folgenden Aussagen bezüglich der Auswahl und Einführung von Testwerkzeugen sind zutreffend?</b>	
<b>A</b>	Bei der Auswahl von Testwerkzeugen sollte zur Risikominimierung immer ein externer Dienstleister beigezogen werden.	
<b>B</b>	Zur Auswahl der Testwerkzeuge wird eine Liste verfügbarer Werkzeuge der entsprechenden Kategorie erstellt.	
<b>C</b>	Eine gute Einführung von Testwerkzeugen führt in der Regel zu einer kurzfristigen Produktivitätssteigerung.	
<b>D</b>	Mit der Werkzeugeinführung entstehen lediglich Kosten für Auswahl und Anschaffung.	
<b>E</b>	Die Einführung von Testwerkzeugen ist häufig mit hohen Investitionskosten und Risiken verbunden und muss daher sorgfältig vorbereitet (Auswahl) und durchgeführt (Pilotprojekt, Information, Schulung, etc.) werden.	

**Frage 3.**

	<p>Ihnen wird der folgende Kontrollflussgraph vorgelegt:</p>  <p><b>Welche der folgenden Aussagen bezüglich der Zahl der mindestens benötigten Testfälle ist richtig?</b></p>	
<b>A</b>	<ul style="list-style-type: none"> <li>• 100% Anweisungsüberdeckung: 1 Testfall</li> <li>• 100% Zweigüberdeckung: 2 Testfälle</li> </ul>	
<b>B</b>	<ul style="list-style-type: none"> <li>• 100% Anweisungsüberdeckung: 2 Testfälle</li> <li>• 100% Entscheidungsüberdeckung: 2 Testfälle</li> </ul>	
<b>C</b>	<ul style="list-style-type: none"> <li>• 100% Anweisungsüberdeckung: 2 Testfälle</li> <li>• 100% Zweigüberdeckung: 3 Testfälle</li> </ul>	
<b>D</b>	<ul style="list-style-type: none"> <li>• 100% Entscheidungsüberdeckung : 3 Testfälle</li> <li>• 100% Zweigüberdeckung: 2 Testfälle</li> </ul>	

**Frage 4.**

	<p>Reviews können zur Evaluation von Alternativen eingesetzt werden.</p> <p><b>Welche der folgenden Reviewarten eignet sich zur Evaluation von technischen Lösungen am besten?</b></p>	
<b>A</b>	Inspektion	
<b>B</b>	Technisches Review	
<b>C</b>	Walkthrough	
<b>D</b>	Informelles Review	

**Frage 5.**

	<p>Es wird Ihnen das folgende Programm vorgelegt:</p> <pre>begin   a=3;   b=2*a;   c=b+100;    if (b &gt; 4) then     if ((b &gt; 8) OR (c &gt; 120)) then       ...     endif   else     ...   endif end</pre> <p><b>Wie gross ist die zyklomatische Komplexität dieses Programmes?</b></p>	
<b>A</b>	3	
<b>B</b>	4	
<b>C</b>	5	
<b>D</b>	6	

**Frage 6.**

	<p>Während eines Code-Reviews stossen Sie auf die folgende Funktion:</p> <pre> int searchForSubString(const char *buf,                       const char *subString) {     int i=0;     int j=0;     bool done=false;     int ret_val=-1;      while(buf[i] &amp;&amp; !done){         if(buf[i]==subString[0]){             j=1;             while(subString[j] &amp;&amp; !done){                 if(buf[i+j] != subString[j]){                     done=true;                 }                 j++;             }         } /* endif */         if(subString[j]==0){             done=true;         }         i++;     }      if(subString[j]==0){         ret_val=i+j;     }      return (ret_val); }                 </pre> <p><b>Wie gross ist die maximale Verschachtelungstiefe dieser Funktion?</b></p>	
<b>A</b>	3	
<b>B</b>	4	
<b>C</b>	5	
<b>D</b>	6	

**Frage 7.**

	<p>Ihnen wird die folgende Methode vorgelegt:</p> <pre> public boolean insert (Comparable c) {     Node parent = null;     Node child = root;      while (child != null) {         parent = child;         int cmp = c.compareTo (child.getElement());         if (cmp == 0) {             return false;         } else {             if (cmp &lt; 0) {                 child = child.getLeft ();             } else {                 child = child.getRight ();             }         }     }      if (parent == null) {         root = new Node (c);     } else {         if (c.compareTo (parent.getElement ()) &lt; 0) {             parent.setLeft (new Node (c));         } else {             parent.setRight (new Node (c));         }     }      return true; }                 </pre> <p>(Gehen Sie davon aus, dass die while-Schleife und die nachfolgenden if-Abfragen unabhängig voneinander sind und entsprechend "passende" Testdaten gefunden werden können.)</p> <p><b>Wieviele Testfälle sind mindestens nötig, um 100% Entscheidungsüberdeckung zu erreichen?</b></p>	
<b>A</b>	Es werden 2 Testfälle benötigt	
<b>B</b>	Es werden 3 Testfälle benötigt	
<b>C</b>	Es werden 4 Testfälle benötigt	
<b>D</b>	Es werden 5 Testfälle benötigt	

**Frage 8.**

	<p>Sie sind verantwortlich für die Komponententestfälle einer neuen Applikation. Dabei stoßen Sie auf die folgende Definition für eine noch zu entwickelnde Funktion:</p> <pre>                 FUNCTION CheckDiscount (FLOAT Amount,  NATURAL Discount): BOOLEAN;                 // Eingabeparameter:                 // -- Amount:   Wert der gekauften Ware                 //                (0 &lt; Amount)                 // -- Discount: Gewünschter Rabatt                 //                (0 &lt;= Discount &lt;= 100)                 //                 // Rückgabewert:                 // -- TRUE, falls Rabatt berechtigt ist,                 // -- FALSE in allen anderen Fällen.             </pre> <p><b>Welche <u>zwei</u> der folgenden Testentwurfverfahren sind zum Testen der Funktionalität dieses Codes am besten geeignet?</b></p>	
<b>A</b>	Paarweises Testen	
<b>B</b>	Äquivalenzklassentest / Äquivalenzbildung	
<b>C</b>	Zweigtest / Zweigüberdeckungstest	
<b>D</b>	Grenzwertanalyse	
<b>E</b>	Bedingungstest	

**Frage 9.**

	<p><b>Welche der folgenden Aussagen bezüglich der Ausgangskriterien (auch Testendekriterien genannt) und deren Abhängigkeit ist richtig?</b></p>	
<b>A</b>	100% (einfache) Bedingungsüberdeckung impliziert 100% Anweisungsüberdeckung.	
<b>B</b>	100% Entscheidungsüberdeckung impliziert 100% minimal bestimmende Mehrfachbedingungsüberdeckung (auch definierte Bedingungsüberdeckung genannt).	
<b>C</b>	100% Mehrfachbedingungsüberdeckung impliziert 100% Anweisungsüberdeckung.	
<b>D</b>	100% Pfadüberdeckung impliziert 100% Mehrfachbedingungsüberdeckung.	

**Frage 10.**

	<p>Sie werden als Technical Test Analyst beigezogen, um ein Projekt, das mit Qualitätsschwierigkeiten kämpft, zu unterstützen. Beim Studium des Testkonzepts entdecken Sie, dass auf der Komponententeststufe ausschliesslich strukturorientiertes Testen durchgeführt wird.</p> <p><b>Welche der folgenden Aussagen beschreibt die damit verbundene Problematik am zutreffendsten?</b></p>	
<b>A</b>	Testfälle die mittels strukturorientierten Testentwurfsverfahren entwickelt wurden sind, je nach zu testendem Code, meist sehr komplex.	
<b>B</b>	Testfälle die mittels strukturorientierten Testentwurfsverfahren entwickelt wurden benötigen spezielle Werkzeuge zur Ausführung. Dies bedeutet zusätzlichen Aufwand.	
<b>C</b>	Testfälle die mittels strukturorientierten Testentwurfsverfahren entwickelt wurden müssen bei jeder Codeänderung angepasst werden.	
<b>D</b>	Testfälle die mittels strukturorientierten Testentwurfsverfahren entwickelt wurden, erlauben das Finden von funktionalem Fehlverhalten nicht.	

**Frage 11.**

	<p>Gemäss der Anforderungen einer Applikation müssen Daten vor Verlust geschützt werden. Sichergestellt wird diese Anforderung durch einen Backup-Prozess und eine Online-Spiegelung der Datenbank. Zusätzlich werden die Transaktionen aufgezeichnet.</p> <p><b>Welcher der folgenden Testfallentwürfe ist der geeignetste für den Test dieser Anforderung?</b></p>	
<b>A</b>	<p>Durch eine logische Bombe (simuliert für den Test) werden die Daten korrupt, dann wird das Backup eingespielt und anschliessend die Transaktionen nachgefahren.                  Erwartet: Datenbank intakt, kein Datenverlust oder verlorene Daten bekannt (z.B. alle Aufzeichnung der letzten 3 Stunden).</p>	
<b>B</b>	<p>Durch eine logische Bombe (simuliert für den Test) werden die Daten korrupt, dann wird das Backup zurückgespielt.                  Erwartet: Datenbank intakt, kein Datenverlust, oder verlorene Daten bekannt (z.B. alle Aufzeichnung der letzten 3 Stunden).</p>	
<b>C</b>	<p>Durch eine logische Bombe (simuliert für den Test) werden die Daten korrupt, dann wird das Backup zurückgespielt und wenn notwendig mit der Spiegelung abgeglichen.                  Erwartet: Datenbank intakt, kein Datenverlust, oder verlorene Daten bekannt (z.B. alle Aufzeichnung der letzten 3 Stunden.)</p>	
<b>D</b>	<p>Durch eine logische Bombe (simuliert für den Test) werden die Daten korrupt, dann werden die Daten der gespiegelten Datenbank zurückgespielt.                  Erwartet: Datenbank intakt, kein Datenverlust, oder verlorene Daten bekannt (z.B. alle Aufzeichnung der letzten 3 Stunden).</p>	



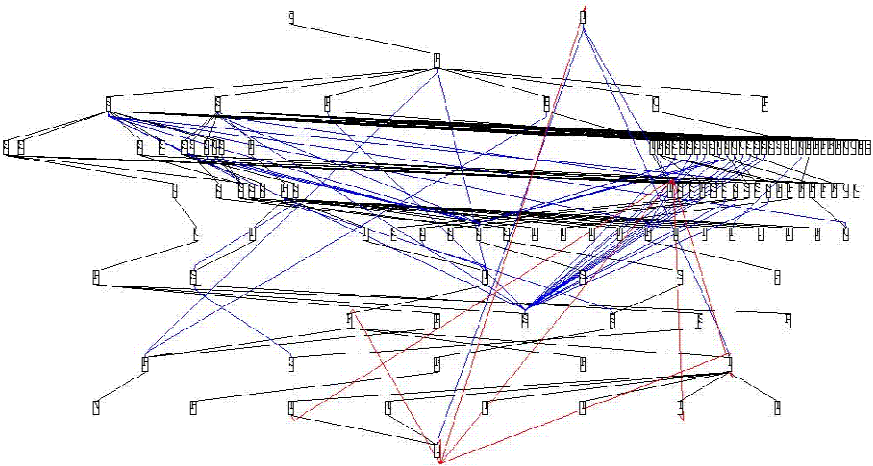
**Frage 12.**

	<p>Ihre Firma ist Lieferant eines Programmes für die Altersversicherung. Der Kunde verlangt die Durchführung eines Mehrfachbedingungstests (auch Mehrfachbedingungs-Überdeckungstest genannt). Während des Codereview stossen Sie auf folgende Quellcodezeilen:</p> <pre>         ...         IF ((Alter &gt;= 65) OR            ((Alter = 64) AND (Geschlecht = 'w')))            ) THEN         ...         ENDIF         ...     </pre> <p><b>Wieviele Testfälle sind nötig, um 100% Mehrfachbedingungsüberdeckung zu erreichen?</b></p>	
<b>A</b>	2 Testfälle werden benötigt	
<b>B</b>	4 Testfälle werden benötigt	
<b>C</b>	6 Testfälle werden benötigt	
<b>D</b>	8 Testfälle werden benötigt	

**Frage 13.**

	<p><b>Welche der folgenden Aussagen bezüglich dynamischer Analysewerkzeuge ist richtig?</b></p>	
<b>A</b>	Dynamische Analysewerkzeuge werden zur Überprüfung der geforderten Performanz der Applikation eingesetzt.	
<b>B</b>	Dynamische Analysewerkzeuge können auch einen Beitrag beim Debugging funktionaler Fehlerwirkungen leisten.	
<b>C</b>	Instrumentierung ist ein typisches Beispiel der dynamischen Analyse.	
<b>D</b>	Der Einsatz dynamischer Analysewerkzeuge hat nur dann einen Sinn, wenn nicht-funktionale Anforderungen definiert sind.	

**Frage 14.**

	<p>Bei der Untersuchung der Softwarearchitektur wurde der nachfolgende Aufrufgraph erstellt.</p>  <p><b>Welche der folgenden Aussagen enthält eine korrekte Schlussfolgerung bezüglich des gezeigten Aufrufgraphen?</b></p>	
<p><b>A</b></p>	<p>Es handelt sich um einen kleinen Aufrufgraphen einer Applikation. Solche Aufrufgraphen können in der Praxis noch viel breiter, tiefer und umfangreicher sein.</p>	
<p><b>B</b></p>	<p>Aufgrund der grossen Zahl von Knoten im Aufrufgraphen kann keine Schlussfolgerung gezogen werden.</p>	
<p><b>C</b></p>	<p>Der Aufrufgraph ist zu gross, zu komplex und daher nicht testbar. Ein "Redesign" sollte in Betracht gezogen werden.</p>	
<p><b>D</b></p>	<p>Die Tiefe und die Breite des Aufrufgraphen können als akzeptabel bezeichnet werden. Hingegen gibt es viele schichtübergreifende Aufrufe, was entweder auf eine nicht ausgereifte Architektur, oder aber auf viele nachträgliche Änderungen schliessen lässt.</p>	

**Frage 15.**

	<b>Welche <u>zwei</u> der folgenden Programmstücke enthalten eine Datenflussanomalie?</b>	
<b>A</b>	<pre>integer a, b, c;  a = 11; a = a + 1; c = 2 * a + b; b = a + c;</pre>	
<b>B</b>	<pre>integer a, b, c;  a = 17; b = a + 1; c = 2 * a + b; b = a + c;</pre>	
<b>C</b>	<pre>integer a, b, c;  a = 21; b = 8 + a; c = 2 * a + b * b; b = a + c;</pre>	
<b>D</b>	<pre>integer a, b, c;  a = -124; b = a + 1; b = 2 * 4a; c = a + b;</pre>	
<b>E</b>	<pre>integer a, b, c;  a = 73; b = a + 1; c = 2 * a + b; a = b + c;</pre>	

**Frage 16.**

	<p>Sie sollen Ihrem Vorgesetzten die Datenflussanalyse anhand des folgenden Codebeispiels erläutern:</p> <pre>BEGIN Max (a: INTEGER, b: INTEGER)      c: INTEGER     IF (a &gt; b) THEN         c := a;     ELSE         c := b;     END IF      RETURN (c) END</pre> <p><b>Welche der folgenden Antworten enthält alle in diesem Codebeispiel möglichen Zugriffssequenzen (Manipulationssequenzen) für die vorhandenen Datenobjekte?</b></p>	
<b>A</b>	Parameter a: drrdu und dru Parameter b: dru und drru Variable c: udru und udru	
<b>B</b>	Parameter a: drru und dru Parameter b: dru und drru Variable c: udru und udru	
<b>C</b>	Parameter a: drru und dru Parameter b: dru und drru Variable c: udu und udru	
<b>D</b>	Parameter a: drru und dru Parameter b: dru und drru Variable c: udru und udru	

**Frage 17.**

	Ein neues Analysegerät soll unter anderem den folgenden nicht-funktionalen Anforderungen genügen: <ul style="list-style-type: none"> <li>• Eine Analyse darf nicht länger als 32ms dauern.</li> <li>• Es müssen Serien von 24'000 Analysen ununterbrochen verarbeitet werden können.</li> <li>• Das Gerät soll auf drei Plattformen angeboten werden.</li> </ul> <p><b>Welche <u>drei</u> der folgenden Software Qualitätsmerkmale bzw. Teilmerkmale gemäss ISO 9126 sind für dieses neue Produkt von Bedeutung?</b></p>	
<b>A</b>	Zuverlässigkeit	
<b>B</b>	Wiederherstellbarkeit	
<b>C</b>	Übertragbarkeit	
<b>D</b>	Wartbarkeit	
<b>E</b>	Interoperabilität	
<b>F</b>	Benutzbarkeit	
<b>G</b>	Zeitverhalten	

**Frage 18.**

	<p><b>Für den Test einer Funktion sind, je nach gefordertem Überdeckungsgrad, mehr oder weniger Testfälle nötig.</b></p> <p><b>Welche <u>zwei</u> der folgenden Aussagen bezüglich jeweils 100%iger Überdeckungen können zutreffen?</b></p>	
<b>A</b>	Zur Anweisungüberdeckung werden 14, zur Zweigüberdeckung 15 Testfälle benötigt.	
<b>B</b>	Zur minimal bestimmenden Mehrfachbedingungsüberdeckung (auch definierte Bedingungsüberdeckung) werden 12, zur Anweisungsabdeckung 15 Testfälle benötigt.	
<b>C</b>	Zur Pfadabdeckung werden 19485, zur Anweisungüberdeckung 443 und zur Zweigüberdeckung 442 Testfälle benötigt.	
<b>D</b>	Zur Anweisungüberdeckung und zur Zweigüberdeckung werden 35 Testfälle benötigt.	
<b>E</b>	Zur Zweigüberdeckung werden 36, zur minimal bestimmenden Mehrfachbedingungsüberdeckung (auch definierte Bedingungsüberdeckung) 24 Testfälle benötigt.	

**Frage 19.**

	<b>Welche der folgenden Aussagen bezüglich fehlerbasierten Testentwurfsverfahren ist richtig?</b>	
<b>A</b>	Fehlerbasierte Testentwurfsverfahren gelangen dann zu Einsatz, wenn Fehler nachgetestet werden sollen.	
<b>B</b>	Fehlerbasierte Testentwurfsverfahren basieren auf der Erfahrung mit einer bestehenden Version des Testobjektes.	
<b>C</b>	Fehlerbasierte Testentwurfsverfahren haben den Nachteil, dass dabei keine Testprotokolle erstellt werden.	
<b>D</b>	Fehlerbasierte Testentwurfsverfahren können auf einer hierarchischen Liste von Fehlerarten (Fehlertaxonomie) basieren.	

**Frage 20.**

	Während eines Projekts tritt eine grössere Diskussion zum Thema Testautomatisierung auf.  <b>Welche <u>zwei</u> der folgenden Aussagen sind richtig?</b>	
<b>A</b>	Automatisierung der Testdurchführung bringt während des Projekts manchmal keine Vorteile, da sich das Testobjekt noch mehrmals ändert.	
<b>B</b>	Testautomatisierung ist sehr teuer und lohnt sich daher nur bei grossen Systemen.	
<b>C</b>	Testautomatisierung erlaubt es, auf die systematische Testfallermittlung zu verzichten.	
<b>D</b>	Testautomatisierung ist grundsätzlich weniger aufwendig als manuelles Testen.	
<b>E</b>	Testautomatisierung kann unter Umständen durch den Testprozess bedingt werden (z.B. zur Messung von Überdeckungsgraden).	