

## Fragenkatalog zur Schulaufgabe

### Windows Grundlagen

1 Welche Bedeutung hatte die Einführung grafischer Benutzeroberflächen für die Verbreitung von PCs?

AW: Durch die einfachere Bedienung erreichte Windows den Massenmarkt.

2 Nennen Sie drei Merkmale von Windowsprogrammen

AW:

- ereignisorientiert
- Darstellung in Fenstern
- grafische Benutzeroberfläche

3 Erläutern Sie den Unterschied zwischen einem Programm und einem Prozess

AW: Ein Prozess ist eine Art virtuelle Hülle, die das Programm umgibt (virtueller Adressraum mit 4 GByte Speicher). Das Betriebssystem startet den Hauptthread des Programms. Je nach Implementierung kann dieser weitere Threads erzeugen und ausführen lassen. Pro Programm wird ein Prozess gestartet.

4 Was versteht man unter einer „Instanz eines Programms“?

AW: Wird ein Programm mehrfach geöffnet, so spricht man von mehreren Instanzen des gleichen Programms. Jede laufende Programminstanz läuft in einem eigenen Prozess!

5 Beschreiben Sie die Aufgaben des Windows Betriebssystems, wenn eine Anwendung geschlossen wird

AW:

Betriebssystem entzieht den Threads die Rechenzeit, Threads werden beendet  
Daten und Programmcode werden aus dem Prozess entladen, der Prozess wird geschlossen

6 Weshalb können unterschiedliche Anwendungen in Win32 nicht über Zeiger auf die gleiche Speicheradresse miteinander kommunizieren

AW: Jeder Prozess erhält seinen eigenen virtuellen Adressraum zugewiesen. Da es sich um virtuelle Adressräume handelt, kennen diese sich nicht untereinander. Das Betriebssystem verwaltet die Prozesse (die Abbildung der virtuellen Adressen auf die physikalischen Adressen), daher ist ein Datenaustausch nur über das OS (über die Zwischenablage) möglich.

7 Welche Vorteile bieten multithreadingfähige Systeme?

AW: Praktisch zeitgleiche Ausführung mehrerer Aufgaben. Multithreadingfähige Systeme bieten die Möglichkeit, dass Threads andere Threads erzeugen können. Somit können zeitaufwendige Routinen in einen eigenen Thread ausgelagert werden. Da jeder Thread am Multitasking teilnimmt, kann die Oberfläche einer Anwendung auch während der Abarbeitung einer längeren Routing weiterhin bedient werden.

8 Erläutern Sie den Unterschied zwischen kooperativem und preemptivem Multitasking

AW:

- kooperativ: Das Betriebssystem gibt die Kontrolle über die CPU an die Anwendung ab. Stürzt die Anwendung ab, so passiert dies mit dem gesamten System.
- Preemptiv: Das Betriebssystem behält die Kontrolle über die CPU, gibt lediglich ein Zeitfenster für die CPU-Rechnung an die Anwendung ab. Stürzt die Anwendung ab, kann sie einfach vom Betriebssystem beendet werden (Taskmanager).

9 Warum können „abgestürzte“ DOS-Programme das gesamte Betriebssystem lahmlegen?

AW: DOS arbeitet kooperativ. Das heißt, dass das abgestürzte Programm die Kontrolle über die CPU hatte. Aufgrund des Absturzes kann die Kontrolle nicht mehr abgegeben werden.

10 Beschreiben Sie die Abarbeitung einer Tastatureingabe über die Message Loop

AW:

- Ereignis tritt auf
- Ereignis wird in die System Message **Queue** eingetragen
- System Message **Queue** wird abgearbeitet (Zuweisung des Ereignisses in die Application **Queue**)
- Botschaft wird von der Message Loop empfangen, übersetzt und an Windows weitergeleitet, Windows leitet die Botschaft dann an die Fensterfunktion zur Bearbeitung weiter.

11 Welche Aufgabe erfüllt die Fensterfunktion (WndProc(...))?

AW:

Erhält die Botschaft und prüft in einer Switch-Anweisung, ob diese bearbeitet oder im default Zweig an eine Standard-Methode (DefWindowProc(...)) weitergeleitet wird.

12 Beschreiben Sie den grundsätzlichen Aufbau einer Fensterfunktion unter Win32

AW:

Erhält die Botschaft und prüft in einer Switch-Anweisung, ob diese bearbeitet oder im default Zweig an eine Standard-Methode (DefWindowProc(...)) weitergeleitet wird. Wird zum Beispiel eine WM\_DESTROY- Botschaft empfangen, so wird die PostQuitMessage-Methode aufgerufen und die Fensterfunktion mit einer return- Anweisung beendet.

13 Erklären Sie die Bedeutung von Botschaften in ereignisgesteuerten Systemen

AW: Windows allein - OHNE DotNet-Framework - kennt bereits ca. 500 verschiedene Botschaften. Der Programmierer muss nicht alle Botschaften selbst bearbeiten sondern kann sie zur Standardverarbeitung zuführen. Worauf es ankommt ist, dass er im Programm die wirklich interessanten Botschaften abfängt. **Ohne Windows Botschaften wäre ein ereignisgesteuertes System nicht möglich.**

14 Nennen Sie vier Schnittstellen für Windows-Programmierer und deren spezielles Anwendungsbeispiel

AW:

- TAPI: Telefonie - API für den Zugriff auf Modems, Fax, ...
- WinSock: Für Netzwerk Funktionalität

- WinInet: die Windows Internet - API
- RAS: für den Zugriff auf Ressourcen von Außen

15 Welche Bedeutung hat die API für den Programmierer?

AW: Die API (Application Programming Interface) ist eine Schnittstelle zwischen dem Programm und dem Entwickler. Sie bringt eine Reihe von Methoden, Variablen und Typen, Structures, Konstanten und Makros mit, die der Programmierer verwenden kann.

16 Erläutern Sie, welches Wissen ein Windows-Anwendungsentwickler zusätzlich zur Kenntnis der Programmiersprache benötigt

AW: Er muss wissen, wie er die Methoden, Variablen und Typen, Structures, Konstanten und Makros verwenden kann.

17 Erläutern Sie die vier Parameter der WinMain(...) Prozedur

AW:

- HINSTANCE d1: von Windows definierter Datentyp für den Instance-Handle des Fensters
- HINSTANCE d2: Zeiger auf eine weitere laufende Instanz des Fensters unter Win16. Aber Win32 immer NULL (da jede Instanz in einem eigenen Prozess läuft)
- LPSTR d3: Kommandozeilenparameter als String
- int d4: Speichert, wie das Fenster angezeigt werden soll

18 Was versteht man unter einer Windows „Fensterklasse“?

AW: Keine Klasse im objektorientierten Sinn, sondern eine Structure zur Speicherung der Eigenschaften des Fensters und eines Zeigers auf seine Fensterfunktion.

Wichtig ist eine eindeutige Bezeichnung der Fensterklasse zur Registrierung unter Win32!

19 Beschreiben Sie die Inhalte der Windows „Fensterklasse“

AW: Sie speichert alle Eigenschaften eines Fensters und einen Zeiger auf seine Fensterfunktion in einer Structure. (Wie bei 18., eindeutige Bezeichnung zur Registrierung nicht vergessen)

### Das Document-View-Modell

1. Erläutern Sie das Grundkonzept des Document-View-Modells

AW: Die Trennung von Daten (Document) und Darstellung (View) sind das wesentlichste Grundkonzept. Daten können so unabhängig von der Darstellung bearbeitet und in verschiedenen Views dargestellt werden.

2. Beschreiben Sie die Unterschiede zwischen einem einfachen Windows Programm (z.B. Editor) und einer komplexen Windows-Anwendung (z.B. MS Word)

AW: Der Editor stellt den Text einfach nur dar. In Word hingegen kann er beliebig bearbeitet werden. (Fett, kursiv, unterstrichen, eingerückt, in Tabellen, ...)

Editor: SDI (Single Document Interface) nur ein Document kann zur selben Zeit geöffnet und angezeigt werden.

Word: MDI (Multiple ...) ...

3. Erklären Sie mögliche Vorteile des Document-View-Modells

AW: Daten können unabhängig von der Darstellung bearbeitet und in

verschiedenen Views dargestellt werden. Dadurch ergeben sich eine bessere Wartbarkeit, bessere Erweiterbarkeit und eine bessere Wiederverwendbarkeit.

4. Beschreiben Sie die verwendeten Klassen eines Document-View-Programms

AW:

- Die Anwendung (abgeleitet von CWinApp):
  - o Bei jeder MFC-Anwendung benötigt man eine Klasse für die Applikation selbst.
  - o Diese wird von CWinApp abgeleitet
- Das Rahmenfenster der Anwendung (CFrameWnd) mit Menü, Toolbar
- Die Dokumentenklasse (CDocument) für die Daten
- Die Ansichtsklasse (CView)

5. Erläutern Sie die Aufgaben der Klassen CWinApp, CFrameWnd, CDocument und CView

AW:

CWinApp:

Beinhaltet die Funktionalität zum Erstellen des Prozesses bzw. Hauptthreads. Erzeugt alle anderen Komponenten in der Anwendung. Diese Klasse empfängt alle Ereignisnachrichten und übergibt diese an die Klassen CFrameWnd und CView.

CFrameWnd:

Diese Klasse bildet den Fensterrahmen. Sie nimmt Menüs, Symbolleisten, Bildlaufleisten und alle anderen sichtbaren Objekte auf, die mit dem Rahmen verbunden sind. Sie bestimmt wie viel vom Dokument zu einem bestimmten Zeitpunkt zu sehen ist.

CDocument:

Diese Klasse nimmt ihr Dokument auf. In dieser Klasse werden die erforderlichen Datenstrukturen, um die Daten des Dokuments zu speichern und zu verwalten, erstellt. Sie empfängt die Eingaben von CView und übergibt Anzeigeinformationen an diese. Sie ist für das Speichern und Abrufen der Dokumentdaten in bzw. aus Dateien verantwortlich.

CView:

Diese Klasse ist die visuelle Repräsentation des Dokuments. Sie reicht Eingabeinformationen an die Klasse CDocument weiter und empfängt Anzeigeinformationen aus dieser. Diese Klasse hat mehrere abgeleitete Klassen, die man als Basisklassen für die Ansichtsklassen verwenden kann.

6. Beschreiben Sie vier von CView abgeleitete Klassendefinitionen

AW:

CEditView:

Liefert Funktionalität eines Eingabefelds. Mit dieser Klasse lassen sich einfache Texteditoren implementieren.

CFormView:

Basisklasse für Ansichten, die Steuerelemente enthalten. Diese wird für formularbasierte Dokumente in Anwendungen verwendet.

CHtmlView:

Liefert Funktionalität eines Webbrowsers. Diese Ansicht behandelt direkt die URL-Navigation, Hyperlinks usw. Sie verwaltet eine Verlaufsliste für das Durchsuchen in Rückwärts-

und Vorwärtsrichtung.

CListView:

Stellt Funktionalität von Listen bereit.

7. Welchen Zweck hat die CSingleDocTemplate Klasse in der InitInstance Methode der Applikation?

AW: Erstellt eine Dokumentvorlage zur Verbindung zwischen Dokument, View, Rahmenfenster und Hauptressource.

8. Auf welche Weise kann die View auf Daten des Documents zugreifen?

AW: über die Methode CDocument\* GetDocument() const. Sie liefert einen Zeiger auf das Document-Objekt zurück.

9. Beschreiben Sie, wie Menüeingaben in VC++ verarbeitet werden

AW: ON\_COMMAND Botschaft wird in den MessageMaps in folgender Reihenfolge abgearbeitet:

- View
- Doc
- DocTemplate
- Frame
- Application

10. Weshalb versendet Windows eine COMMAND- Botschaft bei einem Menüereignis?

AW: Weil nicht genau erkannt werden kann, welche Klasse die Botschaft empfangen soll. Botschaften können z.B. die View oder das Rahmenfenster betreffen.

11. Erläutern Sie die Bedeutung der WM\_PAINT Botschaft und der zugehörigen Methode OnPaint() der View

AW: Bei jedem Neuzeichnen - z.B. durch Verschieben des Fensters - wird die WM\_PAINT Botschaft versandt. Die empfangende Methode muss die OnPaint() Methode der View implementieren **und den Fensterinhalt neu zeichnen.**

12. Wozu benötigt man den Device Context in der OnPaint(...) Methode

AW: Zur hardwareunabhängigen Ausgabe. Über den Device Context kommuniziert das Programm mit dem Betriebssystem. Dieses sendet die Botschaft dann über den Treiber an die entsprechende Hardware.

13. Nennen Sie Methoden des Device Contextes

AW:

- TextOut(...)
- Rectangle(...)
- Ellipse(...)

### Serialisierung in VC++

1 Erläutern Sie den Begriff Serialisierung

AW: Das Speichern (Serialisierung) und Wiederherstellen (Deserialisierung) von Daten und Objekten zusammen nennt man Serialisierung.

2 Erklären Sie die an der Serialisierung beteiligten MFCKlassen

AW:

- CArchive: Agiert als Ein-/Ausgabestream für ein CFile-Objekt
- CFile: Klasse für Dateien – managed das Speichern der Daten

3 Beschreiben Sie die MFC Makros zur Serialisierung

AW: IMPLEMENT\_SERIAL(CBeispiel, CObject, 1):

Makro zur Implementierung der Serialisierung. CBeispiel ist die aufzurufende Klasse, CObject die Basisklasse und der letzte Parameter entspricht der Versionsnummer als unsigned int.

Das Makro DECLARE\_SERIAL übernimmt als einziges Argument den Klassennamen und fügt automatisch bestimmte Standardfunktionen und Operatordeklarationen in die Klasse ein.

### Drucken in VC++

1 Erläutern Sie die Aufgabe des Gerätekontextes beim Drucken unter Visual C++

AW: Zur hardwareunabhängigen Ausgabe. Über den Device Context kommuniziert das Programm mit dem Betriebssystem. Dieses sendet die Botschaft dann über den Treiber an die entsprechende Hardware.

2 Erklären Sie, weshalb ein Programm keine Druckvorschau erzeugen kann, wenn kein Drucker installiert ist

AW: Es kann nicht ermittelt werden, wie der Drucker mit dem Format umgeht. Der Device Context arbeitet mit dem Betriebssystem zusammen, das die installierten Treiber nutzt. Ohne Device Context kann das Programm nicht mit dem Betriebssystem kommunizieren.

3 Erläutern Sie den prinzipiellen Druckvorgang unter Visual C++

AW: Die Methode OnPreparePrinting(...) wird aufgerufen. Mit OnBeginPrinting(...) werden die Zeichengeräte (Schriften, Stifte, Pinsel, ...) erstellt. Solange nicht alle Seiten des Druckauftrags abgearbeitet wurden, wird OnPrint(...) durchlaufen, wodurch die eigentliche Druckausgabe gezeichnet wird. Die Aufräumarbeiten nach dem Drucken erfolgen in OnEndPrinting(...).

4 Welchem Zweck dient das PrintInfo- Objekt beim Drucken?

AW: Über das PrintInfo-Objekt vom Typ CPrintInfo können Druckeinstellungen, wie z.B. die zu druckenden Seiten (von welcher bis zu welcher Seite soll gedruckt werden) vorgenommen bzw. ausgelesen werden

5 Beschreiben Sie die Vorgehensweise zur Ermittlung des bedruckbaren Bereichs

AW: nachdem der Map Mode gesetzt wurde, erstellt man eine CRect-Variable und das PrintInfo-Objekt. Mit DPTOLP werden die absoluten Koordinaten ermittelt und die width- bzw. height- Methoden liefern Breite und Höhe des bedruckbaren Bereichs.

6 Auf welche Weise initialisiert man neue Schriftarten im Ausdruck?

AW: Mit CreateFont(...) wird ein CFont-Objekt erstellt. Über die SelectObject-Methode wird eine neue Schriftart erzeugt und die alte gespeichert.

**Zum Schluss wieder die alte Schriftart setzen!**

### Grundlagen des .NET Frameworks

1 Erklären Sie den Zweck der Common Language Runtime

- 2 Wozu dient die Garbage Collection?
- 3 Beschreiben Sie den Weg eines Quellcodes bis zur Ausführung in einem Programm
- 4 Erläutern Sie den Zweck der MSIL hinsichtlich der Programmiersprachen übergreifenden Entwicklung von Software!
- 5 Erklären Sie den Unterschied zwischen Klassenbezeichnungen und Namensraum
- 6 Wie kann auf eine Klasse in einem bestimmten Namensraum zugegriffen werden?
- 7 Welche Klasse dient als Basis aller .NET Klassen?
- 8 Nennen Sie zwei Arten von Datentypen unter .NET
- 9 Erklären Sie den Unterschied zwischen Reference- und Value Typen
- 10 Was versteht man in .NET unter Boxing und Unboxing?
- 11 Beschreiben Sie die drei Arten von Assemblies
- 12 Wozu dient der Global Assembly Cache?
- 13 Welche Informationen sind in einer Assembly gespeichert?
- 14 Was versteht man unter einer Side By Side Installation?

#### ADO.NET

1. Worin unterscheiden sich connected und disconnected Modell in ADO.NET?
2. Erklären Sie die Begriffe DataProvider und DataSet aus ADO.NET
3. Welche Nachteile hat die Verwendung der OLE DB und ODBC .NET - Data Provider?
4. Erläutern Sie die Vor- und Nachteile des Connected und Disconnected Modells
5. Nennen Sie Beispiele zum Einsatz des Connected und Disconnected Modells
6. Beschreiben Sie die Aufgaben des Connection Objekts, Command Objekts, Parameter Objekts und des DataReader Objekts
7. Welche Einschränkungen bestehen beim Benutzen eines DataReaders?
8. Beschreiben Sie die Zusammenarbeit von DataAdapter, DataSet und DataTables im disconnected Modell
9. Nennen Sie die vier Command - Objekte eines DataAdapters und erläutern Sie deren Aufgabe
10. Welche Collections und Objkte sind in einem DataSet enthalten?