

L I I T E

Kertauskysymykset ja vastaukset

Sivu 1

Luku 1**Kertauskysymykset**

1. Mikä on MSF?

MSF on kokoelma malleja, periaatteita ja käytäntöjä, jotka auttavat organisaatioita tehostamaan kehitystyötä ja teknologian hyväksikäyttöä liiketoimintaongelmien ratkaisemisessa.

2. Mitkä mallit liittyvät kaikkein läheisimmin liiketoimintakeskeisten sovellusten kehittämiseen?

Liiketoimintaan liittyvien sovellusten kehitystyöhön liittyvät läheisimmin Team Model, Process Model, Application Model ja Design Process Model.

3. Mikä on loogisen suunnittelun tärkein näkökulma?

Loogisen suunnittelun näkökulmassa asiaa tarkastellaan projektitiimin kannalta.

4. Kuinka asiakas/palvelin-sovelluksessa voidaan käyttää kolmitasoista sovellusmallia?

Asiakas/palvelin-sovelluksessa voidaan toteuttaa kolme sovelluskerrosta liittämällä liiketoimintakerros joko tietolähdekerrokseen (thin client/fat server malli) tai asiakaskerrokseen (fat client/thin server malli).

5. Mitä hyötyä MFC:stä on ohjelmoijalle?

Näihin etuihin kuuluu mahdollisuus luoda nopeasti ja helposti sovelluksia, kirjastoja ja ohjelmakomponentteja. MFC mahdollistaa nopean kehitystyön kaikilla Windows-ohjelmoinnin osa-alueilla, mukaan lukien käyttöliittymä-, Internet-, komponentti- ja tietokantaohjelmointi — tinkimättä ohjelmoinnin vapaudesta tai joustavuudesta. Lisäksi MFC:hen kuuluu AppWizard, joka on vaiheittain etenevä visuaalinen työkalu, jolla sovelluksen perusrakenteen ja perustoimintojen tekeminen on helppoa.

6. Teet MFC-sovellusta, jonka avulla verkon pääkäyttäjä voi hallita NT-verkon käyttäjätilejä. Mitä kirjastoja täytyy asentaa?

Sinun täytyy asentaa MFC:n kirjastojen Unicodeversiot (joko jaetut tai staattiset, riippuen sovelluksen jakelutavasta). Windows NT:n sisäiset merkkijonot, kuten tilien ja laitteiden nimet, ovat Unicode-muodossa, joten jos sovelluksessa aiotaan käsitellä näitä merkkijonoja, ne täytyy asentaa.

7. Minkälaisia tietoja voidaan käsitellä OLE DB:n avulla?

OLE DB:n avulla voidaan käsitellä kaikenlaista tietoa riippumatta sen tyypistä ja tallennustavasta. OLE DB:n avulla voidaan esimerkiksi käsitellä tietoja, jotka ovat Excel-tilukossa, tekstitiedostossa tai jopa sähköpostipalvelimella kuten Exchange-palvelimella.

Luku2

Kertauskysymykset

1. Mitä projektin asetuksia MFC AppWizard tekee automaattisesti?
AppWizard luo automaattisesti Debug ja Release -konfiguraatiot.
2. Missä voit säätää sovellusikkunan ominaisuuksia AppWizardissa?
Voit muokata sovellusikkunan ominaisuuksia AppWizardin vaiheessa 4 Advanced Options -dialogin Window Styles -välilehdellä.
3. Millaisia ohjeita on saatavissa Visual C++ IDE:ssä?
Visual C++ IDE sisältää MSDN library -ohjeen, joka on ohjelmoijalle välttämätön hakuteos, sisältäen yli gigatavun teknistä ohjelmointitietoa.
4. Mitä asetuksia voit tehdä Project Settings -dialogin C/C++-välilehdellä?
**Project Settings- dialogin C/C++-välilehdellä voit määritellä kääntäjän asetuksia ja kielen ominaisuuksia kuten kutsukäytäntöjä, proses-
sorikohtaisia asetuksia, optimointi- ja esikäsittelijämäärittelyjä.**
5. Mistä syystä saatat joutua käyttämään Visual SourceSafen Get Latest Version -komentoa?
Get Latest Version -ominaisuuden avulla voit varmistaa, että käytössäsi on viimeisin versio kaikista projektin tiedostoista. Get Latest Version -komennon avulla voit myös kopioida projektin lähdekooditiedostot työhakemistoosi, mikäli teet muutoksia, joiden seurauksena koodi tulee yhteensopimattomaksi viimeksi muokattujen modulien kanssa.
6. Missä tilanteissa tarvitaan Visual SourceSafe Merge-toimintoa?
Voit käyttää Merge-toimintoa saman tiedoston eri versioihin tehtyjen muutosten yhdistämiseen yhdeksi tiedostoksi.

Luku 3

Kertauskysymykset

1. Kun ohjelmoit sovellusta MFC:tä käyttäen, missä tilanteissa saatat joutua kutsumaan Win32 APIa suoraan?

MFC sovellusta tehtäessä voidaan joutua kutsumaan Win32 API:a suoraan, jos järjestelmän toimintoja täytyy käsitellä hyvin matalalla tasolla. Jos esimerkiksi teet apuohjelman, jonka avulla käsitellään verkon käyttäjien käyttäjätilejä, täytyy käyttää Win32:n verkkotoimintoja, koska MFC-luokat eivät sisällä Windows NT -verkonhallinta-toimintoja.

2. Mitä palveluja CObject-luokka tarjoaa?

CObject-luokka sisältää tallennuksen (tietojen tallentaminen ja lukeminen tiedostoista), luokan suoritusaikaiset tiedot, tuen diagnostiikalle ja virheen etsinnälle (kelpuutus ja vedostus), että yhteensopivuuden kokoomaluokkien kanssa.

3. Miksi käyttäisit MFC:n laajennettuja DLL:ä?

Koska MFC:n laajennetut DLL:t toteuttavat olemassa olevista MFC-luokista johdettuja uudelleenkäytettäviä luokkia, MFC:n laajennettuja DLL:iä voidaan käyttää MFC:n "laajentamiseen".

4. Minkä MFC-luokan jäsenfunktio toteuttaa sanomasilmukan? Mikä funktio on kyseessä?

CWinApp MFC-luokan Run()-jäsenfunktio toteuttaa sovelluksen sanomasilmukan.

5. Minkä tyyppisissä luokissa sanomakartta voidaan toteuttaa?

CCommandTarget luokasta periytyvät luokat voivat sisältää sanomakartan.

6. Nimeä neljä Windows-sanomien luokkaa, joita sanomakartat voivat käsitellä.

Sanomakartan tuntemat neljä Windows-sanomatyyppiä ovat Windows-sanomat, komentosanomat, käyttöliittymän päivityssanomat ja kontrollien ilmoitussanomat.

7. Mikä on dokumenttimalliobjektin tehtävä?

Dokumenttimalliobjekti liittää dokumenttiluokan joukkoon resursseja (kuten valikkoja ja kuvakkeita) kehysikkunaan ja näkymään.

8. Mikä MFC luokka kätkee GDI:n piirtofunktioita?

MFC:n CDC-luokka kapseloi GDI-piirtofunktioita.

9. Kuinka varmistat sen, että sovelluksesi pääikkunassa on ajantasainen kuva sovelluksen tiedoista?

Varmistaaksesi, että sovelluksen pääikkunan esittämät tiedot ovat ajan tasalla, sinun tulisi kutsua dokumentin UpdateAllViews()-jäsenfunktiota aina dokumentin tietojen muuttuessa niin, että niiden visuaalinen asu muuttuu.

Luku4

Kertauskysymykset

1. Mitä kuvakkeita on toimitettava Windows 98/Windows NT logo-vaatimusten mukaisessa sovelluksissa?

Windows 98/Windows NT -logovaatimukset edellyttävät, että sovelluksen mukana toimitetaan standardikuvake (32x32 pikseliä) ja pieni kuvake (16x16 pikseliä) sekä sovellusta että siihen liittyviä dokumentteja varten.

2. Kuinka työkalurivin vihjetekstit toteutetaan?

Vihjeteksti toteutetaan lisäämällä \n -merkintä ja vihjetekstin Toolbar Button Properties dialogissa määriteltävän Prompt-merkkijonon loppuun.

3. Kuinka sovelluksen koodista voidaan muuttaa valikon tekstiä dynaamisesti?

Valikkokomentojen dynaamiseen muuttamiseen käytetään MFC-luokan CCmdUI jäsenfunktiota SetText.

4. Mikä on suositeltavin tapa tilarivin osoittimen päivittämiseen?

Suosittelava tapa tilarivin osoittimien päivittämiseksi on tehdä niistä merkkijonotaulukko ja yhdistää ne merkkijonotunnisteisiin sovelluksen merkkijonotaulukossa. Tilarivin tekstin päivittäminen kannattaa suorittaa käyttämällä ON_UPDATE_COMMAND_UI-merkintää sanomakartassa käyttöliittymän päivityskäsittelijän ja osoittimen merkkijonotunnisteen yhdistämiseen.

5. Kuinka dialogin muokkausruudusta tehdään vain-luku tyyppinen? Miksi näin pitäisi tehdä?

Dialogin muokkausruudusta voidaan tehdä vain-luku-tyyppinen merkitsemällä Edit Properties -dialogin Styles-välilehdeksi Read Only -valintaruutu valituksi. Vain-luku-tyyppistä muokkausruutua voidaan käyttää dynaamisesti muuttuvan tekstin välittämiseen käyttäjälle.

6. Olet tehnyt dialogimallin dialogieditorilla. Kuinka avaat sovelluksessasi tämän mallin mukaisen modaalisen dialogin?

Dialogieditorilla tehdyn mallin mukaisen dialogin avaaminen modaalisesti edellyttää, että ensin tehdään ClassWizardilla malliin perustuva dialogiluokka. Tämän jälkeen koodissa tehdään luokan ilmentymä ja kutsutaan kantaluokan metodia CDialog::DoModal().

Luku 5**Kertauskysymykset**

1. Kuinka **DoDataExchange()**-funktiota kutsutaan? Kuinka se määrittelee tiedonsiirron suunnan?

DoDataExchange()-funktiota kutsuu CWnd::UpdateData()-funktio. DoDataExchange()-funktio välittää tietoja dialogilta sovellukselle ja sovellukselta dialogiin; CDataExchange-luokan jäsenmuuttuja m_bSaveAndValidate osoittaa tiedonsiirron suunnan.

2. Kuinka kontrollin käyttäminen estetään dialogissa?

Dialogin kontrollin käyttäminen estetään kutsumalla funktiota CWnd::EnableWindow() ja käyttämällä argumenttia FALSE.

3. Mitä kahta funktiota käytetään koordinaatiston kohdistamiseen vapaassa kohdistustavassa?

Koordinaatiston kohdistamiseen käytetään vapaassa kohdistustavassa CDC-funktioita SetWindowExt() ja SetViewportExt().

4. Mikä funktio yhdistää piirtotyökalun piirtopintaan?

Piirtotyökalun yhdistää piirtopintaan CDC::SelectObject()-funktio.

5. Minkälaisia säikeitä täytyy luoda CWinThread-objektia käyttämällä?

CWinThread-luokkaa tulisi käyttää aina luotaessa säikeitä, jotka käyttävät MFC:n toimintoja, koska CWinThread-luokka varastoi säikekohtaiset tiedot sisäisesti MFC-ympäristössä.

6. Mikä ero on CCriticalSection-objektilla ja CMutex-objektilla?

Vaikka molemmat objektit ovat synkronointiluokkia, CCriticalSection sallii vain yhden nykyisen prosessin säikeen käsitellä objektia, kun taas CMutex sallii minkä tahansa prosessin yhden säikeen käsitellä objektia.

7. Kuinka ohjeen aihe-tunnisteet määritellään ohjeprojektitiedostossa?

Ohjetunnisteet on sijoitettu .hm-päätteisiin tiedostoihin, jotka ovat ohjeprojektin osana.

8. Kuinka tehdään hyperlinkki ohjeprojektin .rtf-tiedossa?

Hyperlinkki ohjeprojektin .rtf-tiedostossa tehdään kaksoisalleiviivamalla teksti ja sijoittamalla sen perään linkin kohde piilotekstinä. #-alaviittemerkintä osoittaa linkin kohteen. Alaviite sisältää aiheen nimen, joka voi olla ohjetiedoston sisäinen tai jokin ohjeprojektin .hm-tiedostossa määritellyistä tunnisteista.

Luku 6

Kertauskysymykset

1. Kuinka avaat tiedoston tekstiilassa?
Tiedosto avataan tekstiilassa luomalla CStdioFile-objekti ja liittämällä se tekstitiedostoon ja toimittamalla sitten CFile::typeText-lippu Open()-funktioille.
2. Millaisen poikkeuksen nostaa CFile::Open()-funktio?
CFile::Open() **funktio nostaa poikkeuksen CFileException.**
3. Mitä vaiheita luokkaa serialisoitaessa täytyy suorittaa?
 - **Periytetään luokka luokasta CObject tai jostain CObject-luokasta periytetystä luokasta.**
 - **Lisätään luokkaan oletusmuodostin.**
 - **Lisätään luokan määrittelyyn makro DECLARE_SERIAL.**
 - **Lisätään toteutustiedostoon makro IMPLEMENT_SERIAL.**
 - **Ylikuormitetaan CObject::Serialize()-funktio luokassa.**
4. Mitä SerializeElements()-funktion perustoteutus tekee?
SerializeElements()-**funktion perustoteutus kopioi tietoja biteittäin MFC-kokoelman ja arkiston välillä.**
5. Mitä rekisteriavainta tulisi käyttää sovelluksen kaikille tietokonelaitteen käyttäjille yhteisten asetusten tallentamiseen?
Kaikille koneenkäyttäjille yhteisten asetusten tallentamiseen tulisi käyttää avainta HKEY_LOCAL_MACHINE\SOFTWARE.
6. Minne MFC:n profiilinkäsittelyluokat WriteProfileString() ja WriteProfileInt() varastoivat profiiliasetukset?
MFC:n profiilinhallintaluokat WriteProfileString() ja WriteProfileInt() varastoivat profiilasetukset CWinApp::SetRegistryKey()-funktion määrittelemän avaimen alle.

Luku 7**Kertauskysymykset**

1. Kuinka voit yhdistää ADO-sovelluksen ODBC-tietolähteeseen?

Voit yhdistää ADO-sovelluksen ODBC-tietolähteeseen käyttämällä ODBC:n OLE DB:tä.

2. Mitä vikaa on seuraavassa SQL-lauseessa?

```
SELECT * FROM authors WHERE au_Iname LIKE M%
```

Annetussa koodissa tulisi M% sijoittaa yksinkertaisten lainausmerkkien (‘’) sisälle niin, että siinä lukee ‘M%’. Huomaa, että yksinkertaisia lainausmerkkejä käytetään aina, kun WHERE-lauseen arvo on merkkijono, mutta ei silloin, kun kyseessä on numeroarvo.

3. Sinun täytyy avata CRecordset-objekti niin, että voit tulostaa kirjoittimella kaikki tietokannan taulussa olevat tietueet. Minkä arvon asetat Open()-funktion ensimmäiseksi parametriksi?

Koska tietokantaa luetaan vain kerran tulostettavia tietoja listattaessa, Open()-funktion ensimmäiseksi argumentiksi tulisi antaa CRecordset::forwardonly. Annetun tehtävän suorittamisessa tulisi käyttää aina nopeinta ja vähiten muistia kuluttavaa kohdistinta.

4. Mitä CRecordset::GetDefaultConnect()-funktio määrittelee?

Tämä funktio määrittelee tietojoukon käyttämän tietolähteen.

5. Mihin ADO-objektiin Errors-kokoelma on yhdistetty? Mitä se pitää sisällään?

Errors-kokoelma liittyy Connection-objektiin, ja se sisältää informaatiota tietolähteen virheistä.

6. Mitä ADO Data-kontrollin ominaisuutta käytetään haettavat tietueet määräävän SQL-komennon määrittelemiseen?

RecordSource-ominaisuus määrittää tietolähteestä haettavat tiedot SQL-komentona.

Luku 8

Kertauskysymykset

1. Mikä on COM-rajapinta?

COM-rajapinta on yhteenkuuluvien metodien muodostama rajapinta, joka identifioidaan GUID:lla, josta käytetään myös nimitystä Interface Identifier (IID).

2. Mikä on GUID, ja mikä on sen rooli COM:ssa?

GUID eli maailmanlaajuisesti yksilöllinen tunnistus, on 128-bittinen numerotunnus, jonka on taattu olevan maailman laajuisesti yksilöllinen ja pysyvän yksilöllisenä aina vuoteen 3400.

3. Miten prosessinsisäinen palvelin, paikallinen palvelin ja etäpalvelin-COM-objektit poikkeavat toisistaan?

DLL:nä toteutettu prosessinsisäinen palvelin toimii samassa prosessissa kuin objektin asiakassovellus. Paikallinen (tai prosessin ulkoinen) palvelin, joka on toteutettu .exe-tiedostona, sijaitsee asiakassovelluksen kanssa samassa tietokoneessa, mutta sen suoritus tapahtuu eri prosessissa. Etäpalvelin sijaitsee eri tietokoneessa kuin sen asiakas.

4. Miten toteutetaan perusetähallinta?

Perusetähallinta toteutetaan määrittelemällä rajapinnat IDL:llä ja sallimalla MIDL-kääntäjän luoda etähallintakoodin.

5. Mikä on tyyppikirjasto ja kuinka sitä käytetään?

Tyyppikirjasto on binäärinen kuvaus, johon kuuluvat rajapinnan ominaisuudet ja metodit, sekä metodin argumentit. Tyyppikirjastoja käytetään tilanteissa, joissa korkean tason kielen kuten Visual Basic 6.0:n täytyy voida käsitellä rajapinnan metodeja suoraan vtablen kautta. Jotta tämä olisi mahdollista, Visual Basic -asiakkaalle täytyy tyyppikirjaston avulla kertoa, minkä tyyppistä tietoa rajapinnan metodit vaativat.

6. Mitkä ovat seuraukset COM-objektin määrittämisestä free-threaded malliseksi?

COM-objektien, jotka tehdään tukemaan free-threading mallia, täytyy olla säieturvallisia ja niiden täytyy sisältää oma synkronointikoodinsa. Poistamalla etähallinnan aiheuttaman pullonkaulan, free-threaded objektit tarjoavat palvelimen puolelta parhaan suorituskyvyn ja siirtokapasiteetin.

7. Mitä asioita tulee ottaa huomioon ActiveX-kontrollia toteuttaessa?

ActiveX-kontrollin toteutuksessa tulee huomioida, että vaikka ActiveX-kontrollien määritelmä vaatii tuen vain IUnknown-rajapinnalle, eri säilöillä on erilaisia vaatimuksia. Jotkin säilöt vaativat kontrollilta tiettyjä rajapintoja, joita toiset eivät vaadi. Jos kontrollin halutaan toimivan tiettyssä säilössä, kontrollin täytyy sisältää säilön vaatimat rajapinnat.

Luku 9

Kertauskysymykset

1. Kuvaile mallikantaluokan **CComObjectRootEx** sisältämät ominaisuudet.

Malliluokka CComObjectRootEx sisältää IUnknown-rajapinnan metodien QueryInterface(), AddRef() ja Release() perustoteutukset. Kun luokka periytetään tästä luokasta, asiakas voi QueryInterface()-rajapinnan avulla hankkia osoittimen mihin tahansa COM-objektin tukemaan rajapintaan. CComObjectRootEx sisältää myös IUnknown metodien AddRef() ja Release() toteutukset, jotka huolehtivat viittauksien laskemisesta.

2. Kuvaile mallikantaluokan **CComCoClass** sisältämät ominaisuudet.

Malliluokka CComCoClass sisältää palvelinluokan ilmentymiä luovan luokkatehtaan perustoteutuksen.

3. Mikä on COM-kartta ja kuinka sitä käytetään?

COM-kartta on luokan määrittelyn ominaisuus, joka sisältää luettelon COM-objektin tukemista rajapinnoista. COM-karttaan lisätään merkintöjä käyttämällä COM_INTERFACE_ENTRY-makroa. Aina asiakas-sovelluksen suorittaessa QueryInterface()-funktion, objekti etsii tästä kartasta vastaavan rajapinnan GUID:n. Jos vastaava tunniste löytyy, siihen liittyvä rajapinnan osoitin palautetaan asiakkaalle.

4. Mikä on objektikartta ja kuinka sitä käytetään?

Objektikartta on taulukko, jossa säilytetään tietoja, joita kehys tarvitsee luotaessa objekteja luokkatehdasta käyttäen, asiakkaan ja kantaobjektin välistä kommunikaatiota toteutettaessa, luokan objektien elinikää hallittaessa ja lisättäessä ja poistettaessa objektin määrittelyjä järjestelmärekisteriin. Objektikarttaan lisätään merkintöjä käyttämällä OBJECT_ENTRY-makroa.

5. Kuinka avainsanoja **interface**, **coclass** ja **library** käytetään IDL:ssä, ja kuinka ne liittyvät toisiinsa?

IDL-tiedostossa jokainen COM-objektimäärittely identifioidaan omassa coclass-osassaan ja jokainen rajapinta omassa interface-osassaan. Palvelimen tyyppikirjasto määritellään library-osassa. Yhteistyössä toimivien luokkien määrittelyt, jotka julkistetaan tyyppikirjaston kautta, kootaan library-määrittelyyn.

Luku10

Kertauskysymykset

1. Millainen luokka edustaa COM-rajapintaa C++:ssa?

COM-rajapintoja edustavat C++-koodissa abstraktit luokat.

2. Kuinka `_com_ptr_t`-objekti auttaa COM-palvelimen elinajan kontrolloimisessa?

`_com_ptr_t`-objekti auttaa COM-palvelimen elinajan kontrolloimisessa kutsumalla kapseloidun rajapinnan `IUnknown::AddRef()` ja `IUnknown::Release()` metodeja puolestasi. `AddRef()`-metodia kutsutaan automaattisesti, kun `_com_ptr_t`-objekti luodaan kopioimalla olemassaoleva rajapinnan osoitin, ja `Release()`-metodia kutsutaan automaattisesti, kun `_com_ptr_t`-objektin käyttö loppuu.

3. Mitkä ovat COM:n sisällyttämisen hyvät ja huonot puolet?

Sisällyttämisestä saatava etu on se, että ulomman objektin rajapinta voi kontrolloida sisemmän objektin rajapinnan käyttämistä ja voi määrätä, mitä sisemmän objektin palveluita voidaan käyttää. Sisällyttämisen huono puoli on metodikutsujen uudelleenohjaamisesta aiheutuva pieni lisäkuormitus.

4. Mikä on `CoCreateInstance()`-funktion toisen parametrin merkitys?

Kun ulompi objekti luo koostetun sisemmän objektin, se välittää `CoCreateInstance()`-funktion toista argumenttia käyttämällä `IUnknown`-osoittimensa (kontrolloiva `unknown`) sisemmän objektin luokkatehtäälle. Jos tämä osoite ei ole `NULL`, sisempi objekti tietää, että se on koostettu ja ohjaa ulkoisilta asiakkailta tulevat `IUnknown`-metodin kutsut kontrolloivalle `unknown`:lle.

Luku 11**Kertauskysymykset**

1. Kuinka ominaisuuksien tallentaminen tehdään MFC ActiveX-kontrollissa?

MFC:n ActiveX-kontrollien ominaisuuksien tallentaminen toteutetaan käyttämällä MFC:n ominaisuuksien siirtofunktioita, joiden avulla eri tyyppien serialisointi on mahdollista. Näiden funktioiden nimet alkavat etuliitteellä PX_. PX_-funktiot on sijoitettu kontrolliluokan DoPropExchange()-funktioon.

2. Mikä on DDP-funktion tehtävä ja missä niitä käytetään?

DDP-funktioiden tehtävä on siirtää tietoa ominaisuussivujen ja kontrollin ominaisuuksien välillä. Näitä funktioita käytetään ominaisuusluokan DoDataExchange()-funktiossa.

3. Mikä on yhdistyspiste?

Yhdistyspiste on rajapinta, joka auttaa lähderajapintojen yhdistämisessä vastaavaan asiakasobjektiin. Sink-rajapinnan osoitin välitetään yhdistyspistemekanismin kautta lähdeobjektille. Tämän osoittimen avulla lähde voi käsitellä metodiensa kohteessa olevia toteutuksia.

4. Kuinka CComControl-luokasta periytetty objekti käsittelee Windowsin sanomia?

CComControl-luokan objekti käsittelee Windows-sanomat sanomakartan avulla.

5. Missä tilanteissa MFC soveltuu parhaiten käytettäväksi ActiveX-kontrollin tekemiseen?

MFC yksinkertaistaa merkittävästi ActiveX-kontrollin luomista. Käyttämällä MFC:n ActiveX ControlWizardia voidaan helposti luoda täysin toimivia ActiveX-kontrolleja. ATL:ää käyttäen tehdyt ActiveX-kontrollit ovat pienempiä kuin vastaavat MFC:llä tehdyt kontrollit, ja oikein ohjelmoituina ne ovat suorituskykyisempiä. Käytä MFC:tä, kun ohjelmoinnin nopeus on oleellisempi kuin kontrollin suorituskyky ja koko.

Luku 12

Kertauskysymykset

1. Mikä on suositeltavin tapa selaintyyllisen sovelluksen tekemiseen?
Suosittelava tapa Web-selain –tyylisen sovelluksen tekemiseen on käyttää MFC AppWizardia ja määritellä näkymä luokaksi CHtmlView.
2. Kuinka ATL HTML -kontrollissa määritellään kontrollin esittämän HTML-sivun lähdekoodissa käytettävissä olevat metodit?
ATL HTML-kontrollissa määritellään metodit, joihin kontrollin näyttämän dokumentin HTML-lähdekoodilla on pääsy lisäämällä metodit UI-sanomarakenteeseen, joka on luotu ATL HTML-kontrollin projektiin. Tätä UI-sanomarakentaa käytetään C++-koodin ja HTML UI:n väliseen liikennöintiin. Kontrollin esittämällä HTML-sivulla oleva skripti voi käsitellä kontrollin ominaisuuksia ja metodeja, jotka on määritelty tälle rajapinnalle DHTML window.external-objektin kautta.
3. Kuinka Microsoft Scriptlet Component käsittelee skriptletin mukautetut tapahtumat?
Scriptlet Component ohjaa kaikki mukautetut skriptlet-tapahtumat yhteen tapahtumaan onscriptletevent. Tähän tapahtumaan kuuluu kaksi parametria. Ensimmäinen parametri määrittää tapahtuman nimen, toista parametria voidaan käyttää tapahtumiin liittyvien tietojen välittämiseen.
4. Kuinka voit tallentaa Web-palvelimelta haettuun ActiveX-dokumenttiin tehdyt muutokset?
Voit tallentaa ActiveX-dokumenttiin tekemäsi muutokset käyttämällä File-valikon Save As -toimintoa, jonka kautta voit tallentaa uuden version omalle kiintolevyillesi. ActiveX-dokumenttipalvelin ei anna tallentaa dokumenttiin tehtyjä muutoksia Internet-palvelimelle.
5. Kuinka määritellään, mitkä ActiveX-palvelinsovelluksen valikkokomennoista nykyvät ActiveX-dokumenttisäilössä, kun se lataa ActiveX-dokumentin?
Kun ActiveX-dokumenttisäilö lataa ActiveX-dokumentin, se näyttää ActiveX-dokumenttipalvelimen IDR_SRVR_INPLACE-valikossa määritellyt toiminnot. Muokkaamalla IDR_SRVR_INPLACE-valikkoa (ja työkaluriviä) saat sopivat komennot näkyviin.
6. Kuinka ISAPI-palvelinlaajennuksen oletustoiminto määritellään?
ISAPI palvelinlaajennuksen oletusfunktio määritellään käyttämällä DEFAULT_PARSE_COMMAND-makroa ISAPI laajennusluokan PARSE MAP:ssa.
7. Kuinka ISAPI-palvelinlaajennuksen ja ISAPI-suodattimen lataaminen eroavat?
Vaikka sekä ISAPI-palvelinlaajennukset että suodattimet toteutetaan DLL:inä, ISAPI palvelinlaajennus DLL:t ladataan IIS-prosessiin asiakkaan kutsuessa niitä ensimmäistä kertaa ja ne pysyvät käytössä, kunnes WWW-palvelu suljetaan. ISAPI-suodattimet ladataan, kun WWW-palvelu käynnistetään.

Luku 13

Kertauskysymykset

1. Mitä tarkoittaa Structured Exception Handling?

Structured Exception Handling on Windows-käyttöjärjestelmän palvelu, jonka tehtävä on paikantaa ja kutsua virheen tapahtuessa käsittelijän koodia. Sovelluksen koodi sijoitetaan __try-lohkoon ja virheen käsittelykoodi sijoitetaan __except-lohkoon.

2. Mikä on HRESULT?

COM-ohjelmoinnissa HRESULT on määritelty tyyppi, joka on 32-bit-tinen kokonaisluku ja sisältää virhekoodin. COM-komponenien kuten ActiveX-kontrollien ulkoiset funktiot (tai metodit) palauttavat HRESULT koodin. HRESULT kokonaisluku sisältää bittikentät vakuus, laitteisto ja koodi. HRESULT-arvot ovat etumerkillisiä; nolla tai positiivinen arvo ilmaisevat onnistumista, kun taas negatiivinen HRESULT osoittaa epäonnistumista ja aiheuttaa yleensä poikkeuksen.

3. Nimeä kaksi tapaa, joilla COM-palvelin välittää tietoja virheestä asiakkaalle.

COM-palvelin voi ilmoittaa virheestä palauttamalla HRESULT-koodin tai laukaisemalla Error-tapahtuman.

4. Kuinka sovelluksen debug-versio eroaa julkaisukäännöksestä?

Debug- ja julkaisukäännökset eroavat kahdessa kohdassa. Ensinnäkin ohjelman debug-käännös sisältää kääntäjän objektikoodiin sijoittaman informaation, jota julkaisukäännös ei sisällä. Toiseksi debug-käännöksen koodia ei ole optimoitu, millä varmistetaan käännetyin koodin mahdollisimman tarkka vastaavuus alkuperäisen lähdekoodin kanssa. Julkaisukäännös on lähes aina optimoitu.

5. Kuvaile MFC:n ASSERT, VERIFY ja DEBUG_NEW -makroja.

ASSERT ja VERIFY ovat hyvin samankaltaisia, eroten vain yhdessä suhteessa. Molemmille makroille annetaan parametriksi lause ja ne aiheuttavat poikkeuksen, jos lauseen tulos on FALSE. Ainoa ero on, että ASSERT ei toimi julkaisukäännöksessä, kun taas VERIFY toimii molemmissa käännöksissä.

DEBUG_NEW korvaa new-operaattorin kutsut tarjoten samat palvelut, mutta samalla tallentaen tiedoston nimen ja rivinumeron, jossa new-komento esiintyy. Tämä voi auttaa löytämään muistivuodot, jotka syntyvät tilanteissa, joissa new-operaattoria vastaavaa delete-komentoa ei ole.

6. Kuinka debuggeri saa ajettavan ohjelman keskeyttämään toimintansa?

Voit asettaa debuggausta varten keskeytyskohtia määrättyihin kohtiin sovelluksessa. Kun testattava ohjelma tulee keskeytyskohtaan, ohjelman suoritus keskeytyy ja kontrolli siirtyy debuggerille. Keskeytyskohtia on kahdenlaisia: paikkaan perustuvia ja tietoihin perustuvia. Visual

C++:ssa ovat käytettävissä myös ehdollinen- ja sanomakeskeytyskohta, mutta ne ovat vain paikkaan perustuvan keskeytyskohdan erikoistapauksia.

7. Mitä apuohjelmia Visual C++:ssa on ActiveX-kontrollien testaamista ja ajamista varten?

Visual C++ sisältää ActiveX Control Test Containerin, joka on yleiskäyttöinen säilöohjelma, johon voidaan sijoittaa mikä tahansa ActiveX-kontrolli. Test Container voi kutsua metodeja, asettaa ja lukea ominaisuuksia, aktivoida kontrollin ominaisuusikkunan ja tarkkailla tapahtumia.

8. Mitä tarkoittaa riippuvuus?

Riippuvuudella viitataan suoritettavaan moduliin, tavallisesti DLL- tai ActiveX-kontrolliin, jota ilman sovellusta ei voida suorittaa.

Riippuvuussuhteessa olevilla osilla voi olla myös omia riippuvuuksia. Jos MFC-sovellus ei esimerkiksi käytä suoraan C run-time-kirjastoa; sovellus on kuitenkin siitä riippuvainen, koska MFC itse vaatii C run-time-kirjaston.

9. Mitä tietoja sisältyy Spy++:n näyttämiin neljään listaan?

Spy++ esittää tietoja avoimista ikkunoista, prosesseista ja säikeistä. Se voi myös kirjata sanomat, jotka Windows lähettää tai vastaanottaa.

10. Mitä tarkoittaa regressiotestaus?

Regressiotestauksessa sovellus testataan uudelleen samoilla testeillä, joilla se on jo testattu. Regressiotestausta käytetään, kun sovellukseen on tehty muutoksia ja halutaan varmistaa, että muutokset eivät ole aiheuttaneet uusia bugeja tai odottamattomia toimintoja.

Luku 14

Kertauskysymykset

1. Mitkä ovat jakelun kaksi vaihetta?

Ensin on kaikki tarvittavat tiedostot siirrettävä käyttäjän kiintolevylle. Toisessa vaiheessa täytyy isäntäjärjestelmä konfiguroida niin, että se tunnistaa ja pystyy ajamaan asennetun sovelluksen oikein.

2. Mitä ovat .cab ja rekisteritiedostot?

Tiedostotarkentimesta .cab tunnistettavat tiedostot pitävät sisällään tiedostokokoelman pakatussa muodossa. Rekisteri tai .reg-tiedostot sisältävät skriptejä, joiden avulla järjestelmärekisterin asetuksia voidaan muuttaa käyttämällä apuohjelmaa RegSvr32.

3. Mitä InstallShield-työkalulla tehdään?

InstallShield luo määrätyn Visual C++ projektin asennuspaketin, joka sisältää Setup-ohjelman, .cab-tiedostot ja joukon sekalaisia informaatiotiedostoja. Kun asennuspaketti käynnistetään toisessa tietokoneessa, se asentaa sovelluksen ja konfiguroi tietokoneen automaattisesti niin, että ohjelma voidaan suorittaa.

4. Pitäkö ActiveX-kontrollit rekisteröidä järjestelmään ennen kuin ne toimivat? Jos pitää, mitä esimerkkejä voit antaa kontrollin rekisteröintitiedoista?

ActiveX-kontrollit täytyy rekisteröidä järjestelmään ennen niiden käyttämistä. Tavallisesti tämä tapahtuma ei näy käyttäjälle, koska Web-selain tai asennusohjelma tekee sen automaattisesti. Rekisteriin tulee merkitä kontrollin luokan tunniste ja sen sijainti kiintolevyllä.

5. Nimeä kolme sovelluksen jakelutapaa.

Sovellus voidaan jakaa asennusmedialla, kuten levykkeillä tai CD-ROM:lla, sekä lähiverkon tai Internetin kautta.

6. Mikä on ATL:n käytöstä saatava suurin etu verrattuna MFC:hen ActiveX-kontrolleja tehtäessä?

Tärkein ATL:n käytöstä ActiveX-ohjelmoinnissa saatava etu on se, että ATL:n avulla voidaan tuottaa kontrolleja, jotka eivät ole riippuvaisia ulkopuolisista ohjelman osista. Oikein kirjoitettu ATL:llä tuotettu ActiveX-kontrolli ei tarvitse MFC eikä C run-time-kirjastoa, jolloin riittää, että ainoastaan kontrolli ladataan Web-sivulta. MFC:tä käyttämällä luotuja ActiveX-kontrolleja voidaan käyttää vain, jos MFC42.dll ja MSVCRT.dll ovat järjestelmässä saatavilla.

7. Mitkä ovat kaksi ActiveX-kontrollien lisensoinnissa käytettyä tapaa?

Lisensointi voidaan hoitaa useilla eri menetelmillä, mutta yleensä käytetään erityisessä tiedossa olevaa merkintään tai järjestelmärekisteriin tehtyä merkintään.

8. Olet tehnyt lisensoitava ActiveX-kontrollin ja olet myynyt sen kopion ohjelmistoyhtiölle. Pitääkö sinun antaa heille myös kopio kontrollin lisenssistä? Miksi näin?

Kyllä, tuotteen mukana tulisi toimittaa kopio sen lisenssistä. ActiveX kontrolleja myydään muille ohjelmoijille, jotka puolestaan käyttävät niitä tehdessään loppukäyttäjille suunnattuja sovelluksia tai Web-sivuja. Lisenssikoodi estää ohjelmoijia käyttämästä ActiveX-kontrolleja ilman asianmukaista lisenssiä. Kun ohjelmoija lisää ActiveX-kontrollin sovellusprojektiinsa, kontrolli varmistaa lisenssin olemassaolon; kontrollia voidaan käyttää vain, jos lisenssi on kunnossa.

9. Nimeä kaksi Internet-sivuilla käytettäväksi suunnitellun ActiveX-kontrollin ominaisuutta.

ActiveX-kontrolleista tulisi tehdä pienikokoisia, ja jos käytössä tarvitaan suurta tietomäärää, se pitäisi voida ladata asynkronisesti. Asynkroninen lataus antaa kontrollille mahdollisuuden aloittaa toimintaa käyttäjän kanssa nopeasti samalla, kun ladattavia tietoja puskuroidaan taustalla.

10. Mikä on Zero Administration for Windows?

Zero Administration for Windows (ZAW) on Microsoftin ehdotus suunnitelmaksi Windows-pohjaisen verkon ylläpidon kustannusten ja työvoimatarpeen pienentämiseksi.