

Tentamen i

PDA-applikationer med .NET

(fristående kurs)

lördag 17 december 2005 kl 14:00 - 19:00 i L003

Hjälpmedel:	Vilket tryckt eller skrivet material som helst, bara det är på papper: böcker, utskrifter, egna anteckningar. Inga datorer. Som alternativ till ordbok, om man behöver en sån, kan man dock använda en så kallad "C-pen", som är ett elektroniskt lexikon i form av en penna.
Poängkrav:	Maximal poäng är 34. För betyget G krävs 17 poäng. För betyget VG krävs 25 poäng.
Resultat och lösningar:	Meddelas på kursens hemsida senast lördag 7 januari 2006.
Visning:	Tid för visningen meddelas senare. Efter visningen kan tentor hämtas på expeditionen.
Examinator och jourhavande:	Thomas Padron-McCarthy, telefon 070-7347013.

- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara formuleringar kommer att misstolkas.
 - Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
 - Lös bara en uppgift per blad. Skriv bara på en sida av papperet. Använd inte röd skrift.
 - Antaganden utöver de som står i uppgifterna måste anges. Gjorda antaganden får inte förändra den givna uppgiften.
-

LYCKA TILL!

Uppgift 1 (10 p)

Här är en Pocket PC-applikation skriven i C# för .NET Compact Framework. Man skriver ett ord i rutan, och sen trycker man på knappen. Då talar programmet om vad det var för ord man skrev, på det sätt som bilden visar.



a) (1p)

I övre högra hörnet på handdatorskärmen finns det inte ett kryss, utan det står **ok**. Vad betyder det?

b) (9p)

Skriv hela programmet. (Jag vill alltså att du skriver hela den källkodsfil som man normalt får genom att först köra Designern, och sen lägga till några kodsnuttar här och där. Koden behöver inte se likadan ut

som den som Designern genererar, men den ska fungera likadant. Programmet ska innehålla koden för att placera ut kontrollerna, men de behöver inte ha samma storlekar och placeringar som på bilden.)

Uppgift 2 (2 p)

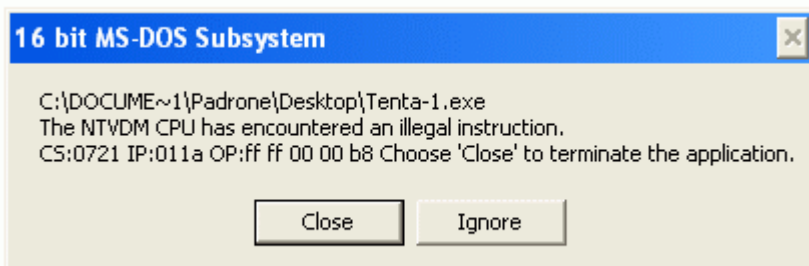
Det färdiga, körbara programmet i uppgiften ovan finns på en fil som heter **Tenta-1.exe**. Det programmet är ju avsett att köras på en handdator, men jag tar den filen och försöker starta den på min vanliga Windows-dator som jag har på skrivbordet.

Vilken av de tre bilderna a-c visar vad som händer, och varför inträffar just detta?

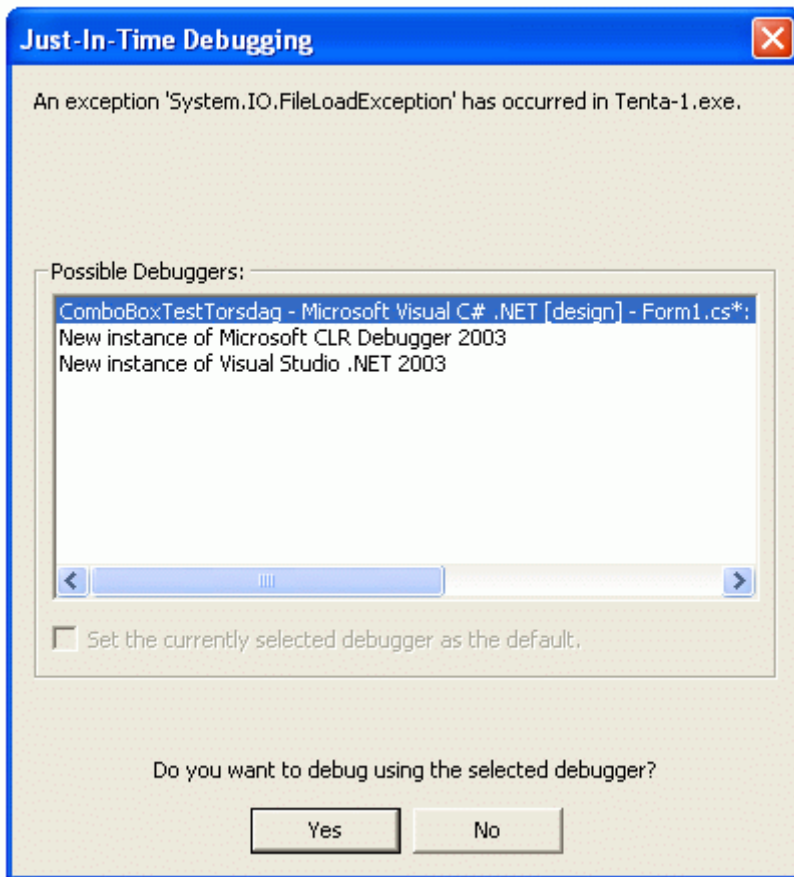
a)



b)



c)



Uppgift 3 (5 p)

Vid all programutveckling är det viktigt att provköra programmen för att försöka hitta fel. Det gäller förstås även när man utvecklar tillämpningar för .NET Compact Framework.

a) (3p)

När man utvecklar program för en vanlig skrivbordsdator kan man oftast provköra programmet direkt på den datorn man programmerar på. Med .NET Compact Framework går inte det, för programmet är ju avsett att köras på exempelvis en handdator. Om man har en handdator tillgänglig, kan man provköra på den, men annars brukar man använda en emulator.

Vilka nackdelar och problem finns det med att testköra sina program i en emulator?

b) (2p)

Boken rekommenderar att man testar sin applikation på samma sorts handdator som ens användare senare ska använda. Varför det? Om jag testar min skrivbordsapplikation på en Dell-PC så kan man ju köra den lika bra på en PC från Hewlett-Packard eller Compaq?

Uppgift 4 (3 p)

En viktig användning av hårddiskar i vanliga datorer är att lagra data så att de finns kvar när datorn stängs av. En Pocket PC har normalt inte någon hårddisk, utan har andra mekanismer för att åstadkomma att data finns kvar när datorn stängs av. Hur gör Pocket PC för att lagra data så de finns kvar när datorn stängs av?

Uppgift 5 (2 p)

Det står ganska mycket i boken om typsytetmet i .NET. Varför det? Det är ju C# vi programmerar i? Förklara!

Uppgift 6 (2 p)

När ett program ska lagra data med hjälp av ADO.NET kan det använda antingen en uppkopplad (connected) eller en icke uppkopplad (disconnected) lösning. Den uppkopplade lösningen är hela tiden uppkopplad mot en databashanterare, medan den icke uppkopplade inte är det, men vad finns det mer för skillnader?

Uppgift 7 (10 p)

Här är en enkel texteditor i form av en Pocket PC-applikation för .NET Compact Framework. Det finns en textruta (en kontroll av typen **TextBox**) för att redigera innehållet i en fil, en annan textruta (också en **TextBox**) för att skriva in filnamn, och tre knappar för att öppna en ny fil, för att spara den redigerade texten, och för att infoga innehållet från en fil:



Om det kastas ett undantag, till exempel för att man försöker skriva en fil i en filkatalog som inte finns, får man ett felmeddelande i form av en **MessageBox**. Den skriver bara ut typen av undantag som kastades, vilket kan hämtas ur egenskapen **Message** (en **string**) i klassen **Exception**. (Forts...)



Om man försöker öppna en fil som inte finns för läsning (antingen med **Öppna**-knappen eller med **Infoga**-knappen) får man i stället ett särskilt felmeddelande:



Redigeringsrutan finns i en variabel som heter **redigeringsrutan**, och filnamnsrutan i en variabel som heter **filnamnsrutan**. Knapparna finns i tre variabler som heter **öppnaknappen**, **sparaknappen** och **infogaknappen**.

a) (5p)

Skriv metoden **öppnaknappen_Click**, som är **Click**-händelsehanterare för **Öppna**-knappen.

b) (5p)

Skriv metoden **sparaknappen_Click**, som är **Click**-händelsehanterare för **Spara**-knappen.
