Örebro universitet
Institutionen för teknik
Thomas Padron-McCarthy (Thomas.Padron-McCarthy@tech.oru.se)

Tentamen i

PDA-applikationer med .NET

(distanskurs)

lördag 10 juni 2006 kl 8:00 - 13:00 i L001

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, gamla kursplanen:	Maximal poäng är 36. För betyget G krävs 18 poäng. För betyget VG krävs 27 poäng.
Poängkrav, nya kursplanen:	Maximal poäng är 26. För betyget G krävs 13 poäng. För betyget VG krävs 20 poäng.
Resultat och lösningar:	Meddelas på kursens hemsida senast måndag 19 juni 2006.
Visning och frågestund:	Måndag 19 juni 2006 kl 12:00-12:30 i mitt rum (T2220). Efter visningen kan tentorna 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!

Två olika kursplaner

Den här tentan gäller både för den gamla och den nya kursplanen:

- Om du läst kursen enligt **den gamla kursplanen** (höstterminen 2005), när det **inte** var några obligatoriska inlämningsuppgifter, ska du göra uppgift 1.
- Om du läst kursen enligt **den nya kursplanen** (vårterminen 2006), när det ingick obligatoriska inlämningsuppgifter, ska du **inte** göra uppgift 1.

Uppgift 1 (10 p)

Den här uppgiften ska man bara göra om man läst kursen enligt **den gamla kursplanen** (höstterminen 2005), när det **inte** ingick några obligatoriska inlämningsuppgifter i kursen.

Här är en Pocket PC-applikation skriven i C# för .NET Compact Framework. Den har en kryssruta och två knappar. Man kan kryssa i och ur rutan genom att klicka i rutan, men också genom att klicka på de två knapparna.

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 (5 p)

Den här uppgiften, och alla följande uppgifter, ska alla göra.

När man köper hamburgare kan man välja mellan olika tilltugg: pommes frites, potatismos och (för den nyttige) sallad. Det här PDA-programmet är tänkt att användas för att välja vad man vill ha:

(a) (2p)

Beskriv hur man gör för att få de tre radioknapparna att "sitta ihop", så att man bara kan markera en av dem i taget.

(b) (3p)

Knapparna **Nyttigt** och **Gott** används för att begränsa vilka val man kan göra. Om man klickar på **Nyttigt**, kan man sen bara välja potatismos eller sallad, men inte pomme. Om man klickar på **Gott** kan man sen bara välja pomme eller potatismos, men inte sallad.

Skriv callback-funktionerna för klick på knapparna. (Man kan lösa det här på flera olika sätt, men enklast är förmodligen att använda egenskapen **Enabled**.)

Uppgift 3 (5 p)

Miljön .NET från Microsoft finns som ".NET Framework", som är den vanliga .NET-miljön för skrivbords- och serverdatorer, och ".NET Compact Framework" för mobila enheter som PDA:er. Diskutera hur dessa två miljöer skiljer sig när det gäller:

a) Intermediate Language
b) klassbibliotek
c) skillnaden mellan "hanterad kod" (managed code) och "ohanterad kod"
d) utvecklingsverktyg
e) kontrollen **ComboBox**

Uppgift 4 (3 p)

a) Vad innebär händelsestyrd programmering ("event-driven programming")?

b) Vad är en händelse ("event")?

c) Vad är en händelsehanterare ("event handler")?

Uppgift 5 (3 p)

a) (2p)

Skräpsamling ("garbage collection") innebär att systemet automatiskt städar bort data som inte längre används. I sina klasser kan man skapa två metoder som har med skräpsamlingen att göra: **Finalize** och **Dispose**. Vad gör dessa metoder?

b) (1p)

Finalize är det interna .NET-namnet på metoden. I C#-koden som man skriver ska den heta något annat. Vad?

Uppgift 6 (3 p)

a) (1p)

Hur startar man en tråd i C#?

b) (2p)

Ibland vill man göra saker som tar lång tid, till exempel skapa komplicerade fönster eller koppla upp sig via Internet, som svar på att användaren klickar på en knapp. Det kan vara klokt att att inte gör allt det jobbet i hanterarmetoden, utan att man i stället startar en egen tråd för det. Varför?

Uppgift 7 (1 p)

Pocket PC-handdatorer visar normalt inte någon muspekare på skärmen. Varför inte?

Uppgift 8 (6 p)

Det är mycket vanligt att ett tillämpningsprogram, vare sig det är ett vanligt datorprogram eller ett PDA-program, hanterar en samling data som sen ska visas på skärmen så att användaren kan studera och ändra i dessa data. Komihåglistan från inlämningsuppgift 2 och 3 är ett exempel: listan finns i någon form inuti programmet, men visas också på skärmen på ett eller flera sätt.

Diskutera några sätt att lösa dels lagringen (var och hur sakerna lagras), dels kopplingen mellan det som lagras och det som visas. (Det är i första hand .NET Compact Framework vi är intresserade av här, men ta gärna upp en mer prinicipiell nivå också.)

Skiljer det mellan (1) en vanlig skrivbordsdator, (2) en serverdator, (3) en PDA med trådlöst nät, och (4) en PDA utan trådlöst nätverk?