



Linnéuniversitetet

Kalmar Vaxjö

Laborationsanvisning

Gissa det hemliga talet

Steg 1, laborationsuppgift 4



Författare: Mats Looch

Kurs: ASP.NET Web Forms

Kurskod: 1DV406

Upphovsrätt för detta verk

Detta verk är framtaget i anslutning till kursen ASP.NET Web Forms (1DV406) vid Linnéuniversitetet.

Du får använda detta verk så här:

Allt innehåll i detta verk av Mats Looock, förutom Linnéuniversitetets logotyp, symbol och kopparstick, är licensierad under:



Creative Commons Erkännande 4.0 Internationell licens.

<http://creativecommons.org/licenses/by/4.0>

Det betyder att du i icke-kommersiella syften får:

- kopiera hela eller delar av innehållet
- sprida hela eller delar av innehållet
- visa hela eller delar av innehållet offentligt och digitalt
- konvertera innehållet till annat format
- du får även göra om innehållet

Om du förändrar innehållet så ta inte med Linnéuniversitetets logotyp, symbol och/eller kopparstick i din nya version!

Innehåll

Uppgift	5
Inledning	Fel! Bokmärket är inte definierat.
Presentationslogiklagret	Fel! Bokmärket är inte definierat.
Textfälten och validering	Fel! Bokmärket är inte definierat.
Alternativknapparna	Fel! Bokmärket är inte definierat.
Kommandoknappen	Fel! Bokmärket är inte definierat.
Temperaturtabellen	Fel! Bokmärket är inte definierat.
Stilmallar	Fel! Bokmärket är inte definierat.
Affärslogiklagret	Fel! Bokmärket är inte definierat.
Hantering av fel på servern	Fel! Bokmärket är inte definierat.
Mål	Fel! Bokmärket är inte definierat.
Tips	Fel! Bokmärket är inte definierat.

Uppgift

Inledning

Skriv en webbapplikation med hjälp av ASP.NET Web Forms och C# där användaren ska ha sju försök på sig att gissa ett hemligt tal i det slutna intervallet mellan 1 och 100.



Figur 1. Användaren klarade av att gissa det hemliga talet på sjunde försöket.

Användaren ska kunna skriva in ett heltal i ett textfält och posta formulärdata genom att klicka på en kommandoknapp. En historik över gjorda gissningar ska visas. Endast resultatet för den senaste gissningen behöver presenteras. Inga HTML-element ska renderas till klienten i onödan.

Användargränssnittlagret

Viss del av funktionaliteten måste implementeras med JavaScript. Du väljer själv om du vill använda dig av ett JavaScript-bibliotek, som till exempel jQuery, eller inte.

Då användaren kan posta en gissning ska textfältet ha fokus, och innehållet ska vara markerat. Kommandoknappen för att skicka en gissning ska vara tillgänglig.

Kan en gissning inte göras ska textfältet och kommandoknappen för att posta en gissning inte vara tillgängliga. Kommandoknappen för att skapa ett nytt hemligt tal ska däremot vara tillgänglig och ha fokus.

Presentationslogiklagret

Du får fritt utforma formuläret där inmatningen av en gissning görs. Presentationslogiklagret ska i möjligaste mån endast innehålla kod som har med hanteringen av kontroller att göra. Kod som har med hantering, förutom validering, av en gissning får under inga omständigheter placeras i presentationslogiklagret (detta gäller så väl aspx- som "code-behind"-fil) utan ska placeras i en affärslogiklagerklass.



Figur 2. Användaren står i begrepp att skicka en första gissning.

Då det sker en GET av sidan ska åtminstone ett textfält och kommandoknapp renderas på klienten genom att använda serverkontrollerna TextBox och Button.

Validering

Innehållet i textfältet ska valideras så att innehållet garanterat kan tolkas som en relevant gissning, d.v.s. ett heltal i det slutna intervallet mellan 1 och 100. Valideringen ska utföras i första hand på klienten. Men det är oerhört viktigt att all data valideras på servern oavsett om det validerats på klienten eller inte.

Textfältet måste vara ifyllt på ett korrekt sätt.

- Det får inte vara tomt.
- Texten måste kunna tolkas som ett heltal i det slutna intervallet mellan 1 och 100.

Med hjälp av kontrollen `RequiredFieldValidator` kan du säkerställa att ett textfält inte är tomt.

Kontrollen `RangeValidator`¹ använder du till att undersöka om ett textfälts innehåll kan tolkas som ett heltal mellan två värden. Egenskaperna `Type`, `MaximumValue`, och `MinimumValue` måste du sätta till lämpliga värden.



Figur 3. Felmeddelande då användaren inte matat in något i textfältet.



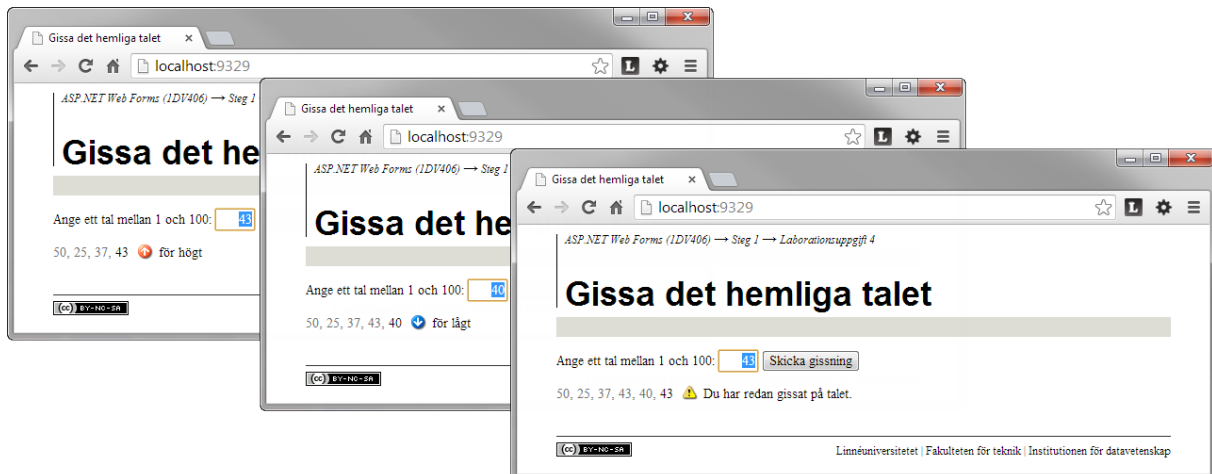
Figur 4. Felmeddelande då ett för stort tal har matats in.

Presentation av resultat av gissning

När en användare gjort en gissning ska resultatet av gissningen, samt tidigare gjorda gissningar, presenteras. Här kan det vara lämpligt att använda kontroller av typen `Label` och i "code-behind"-filen tilldela dem den text som ska visas.

Om användaren har gissat för högt, för lågt eller på ett tidigare gissat tal ska textfältet ha focus och dess innehåll ska vara markerat samt kommandoknappen för att posta en gissning ska vara tillgänglig. Kan inte användaren göra fler gissningar ska textfältet och kommandoknappen inte vara tillgängliga och istället ska kommandoknappen användaren klickar på för att skapa ett nytt hemligt tal vara tillgänglig och ha fokus. Den kontroll som ska ha fokus bestämmer du på lämpligt sätt.

¹ En `RangeValidator` kontrollerar även typen varför det inte finns behov av en separat valideringskontroll som kontrollerar att ett heltal matats in.



Figur 5. Resultaten av gissning på ett för högt, lågt respektive tidigare gissat värde.

Har användaren gissat rätt, eller förbrukat samtliga gissningar, ska det inte gå att göra fler gissningar. Istället ska en ny kommandoknapp visas som användaren kan klicka på för att slumpa ett nytt hemligt tal.



Figur 6. Användaren erbjuds möjligheten att slumpa ett nytt tal efter att ha gissat rätt.

Misslyckas användaren att gissa rätt hemligt tal efter sju försök ska det hemliga talet presenteras.



Figur 7. Resultat då användaren misslyckats att gissa det hemliga talet.

Tänk på att placera kontroller som inte ska visas vid GET av sidan i en `PlaceHolder` så att du enkelt kan bestämma med egenskapen `Visible` om de ska renderas till klienten eller inte.

För att hantera det hemliga talet behöver presentationslogiklagret en referens till affärslogikobjektet. Denna referens sparas lämpligen i en sessionsvariabel som kapslas in i en privat egenskap i "code-behind"-filen.

Observera att endast ett affärslogikobjekt får instansieras under en session. Då användaren för första gången gör en GET av sidan ska ett nytt affärslogikobjekt skapas och initieras med ett nytt slumpstal. Finns det redan ett affärslogikobjekt vid GET, eller POST, av sidan ska det användas och vid behov initieras med ett nytt slumpstal.

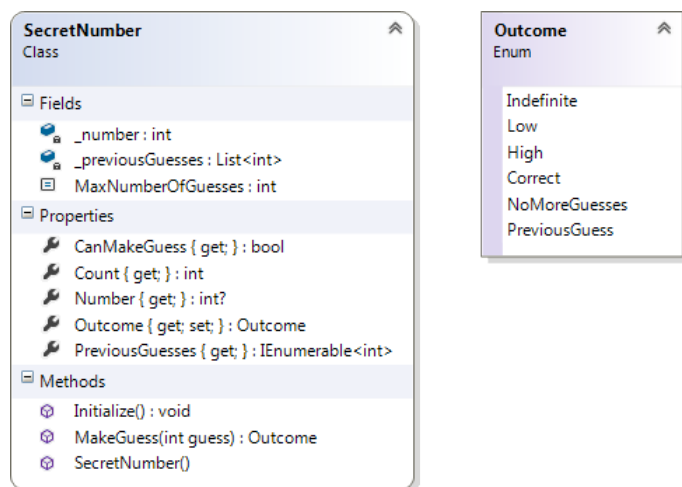
Affärslogiklagret

I figur 8 nedan ser du ett klassdiagram med förslag på hur du kan utforma klassen som ska hålla ordning på det hemliga talet. Du behöver inte utforma din klass på samma sätt utan kan skapa en egen om du så vill.

Genom att omsorgsfullt utforma klassen som ska hålla ordning på det hemliga talet kommer det att bli enklare att för dig att skriva koden som tillhör användargränssnitt- och presentationslogiklagren.

Tänk på att klassen inte ska innehålla några beroenden och exempelvis inte innehålla någon kod som har med hantering av textsträngar (det är en uppgift för presentationslogiklagret) eller ASP.NET att göra.

Exempel på olämplig implementation av klassen är att den beror av användargränssnitt- och presentationslogiklagret beträffande validering av en gissning. Klassen kan inte lita på att detta är implementerat på ett korrekt sätt varför egen validering måste göras. Att en gissning är giltig måste undersökas på lämpligt sätt. Är en gissning inte ett tal i det slutna intervallet mellan 1 och 100 kastas till exempel lämpligen ett undantag av typen `ArgumentOutOfRangeException`. Om användargränssnitt- och presentationslogiklagret råkat utformas så användaren kan göra fler gissningar än sju väljer du själv hur du vill hantera felet (kasta ett undantag eller returnera ett lämpligt värde).



Figur 8. Förslag på medlemmar i klassen² och listvärden i den uppräkningsbara typen. Det framgår inte att set-metoden för egenskapen `Outcome` är privat.

Konstant

`MaxNumberOfGuesses` ger antalet försök en användare har på sig att gissa det hemliga talet.

Fält

`_number` innehåller det hemliga talet.

`_previousGuesses` lagrar samtliga gissningar gjorda sedan aktuellt hemligt tal slumpades fram.

Egenskaper

`CanMakeGuess` ger ett värde som indikerar om en gissning kan göras eller inte.

`Count` ger antalet gjorda gissningar sedan det hemliga talet slumpades fram.

² Var är fälten? Från och med C# 3 behöver du inte skapa fält som kopplas till egenskaper. Genom att använda autoimplementerade egenskaper behöver du inte fält utan det sköter C#-kompilatorn om automatiskt åt dig.

Number ger eller sätter det hemliga talet. **Observera typen! Den är ”nullable”**. Så länge som en gissning kan göras ger egenskapen värdet null. Kan inte en gissning göras ger egenskapen det hemliga talets värde. `get`-metoden är publik.

Outcome ger resultatet av den senast utförda gissningen.

PreviousGuesses ger en referens till en samling innehållande gjorda gissningar. **Observera att den underliggande typen är `ReadOnlyCollection<int>`**. För att undvika en ”privacy leak” av det privata fältet `_previousGuesses` ger egenskapen en ”read-only”-referens till `List`-objektet vilket omöjliggör manipulering av gjorda gissningar utanför affärslogikklassen.

Konstruktor

Klassens enda konstruktor ser till att då ett objekt instansieras av klassen har ett giltigt slumpstal, samt skapar en instans av `List`-objektet med plats för sju element som ska innehålla gjorda gissningar sedan det hemliga talet slumpats fram.

Metoder

`Initialize` initierar klassens medlemmar. `_number` tilldelas ett slumpstal i det slutna intervallet mellan 1 och 100. Eventuella element i `_previousGuesses` tas bort genom anrop av metoden `Clear`. Egenskapen `Outcome` tilldelas värdet `Indefinite`.

`MakeGuess` returnerar ett värde av typen `Outcome` som indikerar om det gissade talet är rätt, för lågt, för högt, en tidigare gjord gissning eller om användaren förbrukat alla gissningar. Är inte gissningen i det slutna intervallet mellan 1 och 100 kastas undantaget `ArgumentOutOfRangeException`.

Hantering av fel på servern

Skulle ett oväntat fel av något slag inträffa ska användaren slippa se en ”gul-ful” sida med ett automatgenererat felmeddelande. Istället ska du visa en sida med ett användarvänligare(?) meddelande. Figur 9 visar ett exempel på en sådan sida.



Figur 9. Anpassad sida som visas vid fel istället för en ”gul-ful” sida.

Krav

Sammanställning av krav som måste uppfyllas:

1. Validering måste ske så att det endast går att posta en gissning som är ett heltal i det slutna intervallet mellan 1 och 100. Validering ska ske på klienten och servern.
2. Användaren har sju (7) försök på sig att gissa rätt tal. Det ska inte gå att göra fler gissningar.
3. Endast ett affärslogikobjekt får skapas per session, vilket innebär att ett och samma objekt ska återanvändas under en och samma session.
4. Alla hantering av det hemliga talet placerar du i en klass som ensam får utgöra affärslogiklagret. Klassen ansvarar för att på lämpligt sätt meddela resultat om användaren gissat rätt tal, gissat för lågt, gissat för högt, redan gissat på talet och om användaren förbrukat alla gissningar.

5. Affärslogikobjektet ska, förutom det hemliga talet, även lagra de gissningar som gjorts efter att det hemliga talet slumpats fram.
6. En gissning görs genom att en metod i klassen anropas. Metoden ska returnera lämpligt värde (enligt punkt 4 ovan) beroende på resultatet av gissningen. Det är inte tillåtet att från affärslogikklassen returnera strängar av något slag. Det rekommenderas starkt att använda en uppräkningsbar typ (enum) som returvärde.
7. Klienten ska inte på något sätt, så länge antalet gissningar inte är förbrukade, kunna komma åt det hemliga talet, som slumpas fram.
8. Antalet gissningar användaren har på sig ska lagras i form av en namngiven konstant i en affärslogikklass.
9. Ingen kontroll får använda sig av "view state". Attributet ViewStateMode till Disabled, i Page-direktivet.
10. Fler element än nödvändigt ska inte renderas till klienten.
11. Underlätta för användaren genom att ge lämpligt element fokus då sidan visas.
12. Vid oförutsedda fel ska en egen felsida visas.

Mål

Du ska:

- Ha förståelse för att en sessionsvariabel är en variabel som är knuten till en session och aldrig lämnar servern.
- Förstå vikten av att validera data innan det behandlas vidare, både på klienten och på servern samt vara väl förtrogen med hur du validerar innehållet i ett textfält med hjälp av valideringskontrollerna som ASP.NET erbjuder.
- Ha fått insikt om att data som skickas till affärslogiklagret måste valideras av affärslogiklagret då valideringen kan vara felaktigt implementerad i användargränssnitt- och presentationslogiklager.
- Förstå att då användaren klickar på en knapp sker en "postback" och en händelse skapas som kan tas om hand i "code-behind"-filen.
- Kunna ta hand om händelsen Click som inträffar då användaren klickar på en knapp.
- Kunna använda Label-kontrollen för att placera ut text som bestäms i "code-behind"-filen.
- Förstå hur en Placeholder kan användas för att bestämma om en grupp kontroller ska renderas till klienten eller inte.

Tips

- Valideringskontroller
 - "Types of Validation for ASP.NET Server Controls",
[http://msdn.microsoft.com/en-us/library/bwd43d0x\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/bwd43d0x(v=vs.100).aspx)
- "Session"
 - <http://msdn.microsoft.com/en-us/library/ms178581.aspx>
 - Tänk på att en sessionsvariabel kan "kapslas in" av en egenskap (protected?). I många fall underlättar detta i stor utsträckning hanteringen av en sessionsvariabel.

- "View State"
 - Pro ASP.NET 4.5 in C#, Fifth Edition, 879-888.
 - *Managing View State in ASP.NET 4 Using the New ViewStateMode Property*,
<http://www.4guysfromrolla.com/articles/071410-1.aspx>.
- ReadOnlyCollection<T>
 - <http://msdn.microsoft.com/en-us/library/ms132474.aspx>
- List<T>.AsReadOnly
 - [http://msdn.microsoft.com/en-us/library/e78dcd75\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/e78dcd75(v=vs.110).aspx)
- "Nullable Types"
 - <http://msdn.microsoft.com/en-us/library/1t3y8s4s.aspx>
- enum
 - [http://msdn.microsoft.com/en-us/library/sbbt4032\(v=vs.120\).aspx](http://msdn.microsoft.com/en-us/library/sbbt4032(v=vs.120).aspx)