.NET

12. 6. 2008

Modulprüfung

Name, Vorname	Klasse	Punkte	Note

Zeit	80 Minuten
Totale Punktzahl	70
Hilfsmittel	Beliebige persönliche **Unterlagen**, **Bücher** und **Taschenrechner**, aber weder ein in C# programmierbarer Rechner noch Natel noch sonstige Kommunikationsmittel.
Abgabe	Füllen Sie das erste Aufgabenblatt aus und schreiben Sie **alle Lösungsblätter mit Ihrem Namen, Vornamen und Ihrer Klasse** sowie der entsprechenden **Aufgabennummer** an. **Geben Sie alle Aufgaben- und Lösungsblätter ab.**
Hinweis	Lesen Sie alle Aufgabenstellungen sorgfältig durch, bevor Sie mit der Bearbeitung der ersten Aufgabe beginnen. Die Aufgaben können in beliebiger Reihenfolge gelöst werden. Halten Sie sich nicht zulange an einer Aufgabe auf.

Aufgabe 1: Allgemeine Fragen zu C# (10 Punkte)

welche der folgenden Aussagen sind wahr *(Vorsicht: falsche Antworten führen zu Abzug)*

a)

ja	nein	weiss nicht	
			Die Fälle der Switch Anweisung müssen durch ganzzahlige, numerische Werte gekennzeichnet sein.
			Partial Classes verwendet man in .NET u.a. um den generierten Code vom selbstgeschriebenen zu separieren.
			Eine Methode, die zur Behandlung von Events verwendet wird, sollte keine Exceptions werfen.
			Die Fehleranfälligkeit des Fallthrough Mechanismus der Switch-Anweisung wurde in C# im Vergleich zu Java entschärft.
			double ist in.NET der Fliesskomma-Datetyp mit der grössten Genauigkeit.

b)

			Eine Assembly, in der eine native C-Funktion aufgerufen wird, muss als *unsafe* deklariert sein.
			Die .NET Code Access Security ersetzt die Sicherheitsmechanismen des darunterliegenden Windows Betriebssystems.
			COM ist eine der möglichen Technologien, um aus dem Managed Code heraus Unmanaged Code zuzugreifen.
			Mit der IL, d.h. dem Assembler von .NET, lassen sich die Mechanismen der CAS umgehen.
			Sämtliche Assemblies des *Global Assembly Caches* werden beim Starten von .NET in den Hauptspeicher geladen.

c)

			Schlägt die Validierung bei einer ASP.NET Seite innerhalb des Browsers fehl, dann wird ein Roundtrip ausgelöst.
			.NET stellt sicher, dass die Equals-Methode und der Gleichheitsoperator immer dasselbe Resultat liefern.
			Der Zuweisungsoperator kann mit `operator =` überschrieben werden.
			Falls nicht Code-Behind verwendet wird, können die Quellen der ASPX Seiten mittels dem Browsers angezeigt werden.
			Die Anwendung von optimistischem Locking setzt in .NET die Verwendung von DataSets voraus.

Aufgabe 2: Vererbung(10 Punkte)

Schreiben Sie auf, was das untenstehende Programm ausgibt.

```
interface SportsCar {
   void WhatAreYou();
}

interface GermanCar {
   void WhatAreYou();
}

abstract class Car  {
   public virtual void WhatAreYou(){Console.WriteLine("I am a Car"); }
}

abstract class Golf : GermanCar{
   public virtual void WhatAreYou() {Console.WriteLine("I am a Golf"); }
}

class GolfGTI : Golf, SportsCar {
   public override void WhatAreYou() {Console.WriteLine("I am a Golf GTI"); }
}

abstract class Opel : Car, GermanCar {
   public override void WhatAreYou(){Console.WriteLine("I am an Opel"); }
}

class Manta : Opel, SportsCar {
   public new void WhatAreYou() {Console.WriteLine("eeh du, echt cool"); }
}
```

```
Car car1 = new Manta();
Golf car2 = new GolfGTI();
```

((Opel)car1).WhatAreYou();	
((GolfGTI)car2).WhatAreYou();	
((Manta)car1).WhatAreYou();	
((SportsCar)car1).WhatAreYou();	
car2.WhatAreYou();	
car1.WhatAreYou();	

Aufgabe 3: Synchronisation/Programmfragment (10 Punkte)

a) Gegeben sei eine Klasse Player, deren Run Methoden in eigenen Threads laufen. Vervollständigen Sie die Methode `FoulPlayerAndGetBall`, so dass die Ballübergabe "kontrolliert" (d.h. Thread safe) erfolgt.

```
Public class Player {

      public bool HasBall;
      public Ball Ball;

      public void FoulPlayerAndGetBall(Player fromPlayer) {
            this.HasBall = true;
            fromPlayer.HasBall = false;
            this.Ball = fromPlayer.Ball;
            fromPlayer.Ball = null;
      }

      public voin Run() {
          while(true) {
           Player p = Playground.Players[random() *22];

            _____________________________

            _____________________________

            if (p != this && p.HasBall ) FoulPlayerAndGetBall(p);

            _____________________________

            _____________________________

            Thread.Sleep(1000);
          }
      }


}
```

b) Schreibe Sie eine Methode die 22 Player Threads startet, und nach 90 Minuten alle Threads wieder stoppt. Bevor das Programm beendet werden kann, müssen aber alle Spieler auch wirklich terminiert sein. (Hinweis: join verwenden).

_____

_____

_____

_____

_____

_____

_____

Aufgabe 4: Reguläre Ausdrücke (10 Punkte)

Gegeben sei die WebSeite: http://www.tages-anzeiger.ch/dyn/news/spielplan/index.html die als wesentliche Information folgendes enthält

SPIELPLAN

Sa	07.06.	18.00	Schweiz – Tschechien	0 : 1	A
		20.45	Portugal – Türkei	2 : 0	A
So	08.06.	18.00	Österreich – Kroatien	0 : 1	B
		20.45	Deutschland – Polen	2 : 0	B
Mo	09.06.	18.00	Rumänien – Frankreich	0 : 0	C
		20.45	Niederlande – Italien	3 : 0	C

als HTML

…

```
<span class="cachtel">Schweiz–Tschechien</span></a>
<td width="80">0:1</td>
<span class="cachtel">Portugal–Türkei</span></a>
<td width="80">2:0</td>
```

…

a)Schreiben Sie einen regulären Ausdruck, mittels dem Sie <Land1>, <Land2> , <Tore1> , <Tore2> aus dem obigen HTML Dokument herauslesen können

b) Schreiben Sie eine Methode FetchPage, die die obige Seite liesst und als String retourniert.

```
String Url =  "http://www.tages-anzeiger.ch/dyn/news/spielplan/index.html"
String FetchPage() {
```

Aufgabe 5: Fehlersuche (10 Punkte)

Gegeben sei folgende VB.NET Klasse.

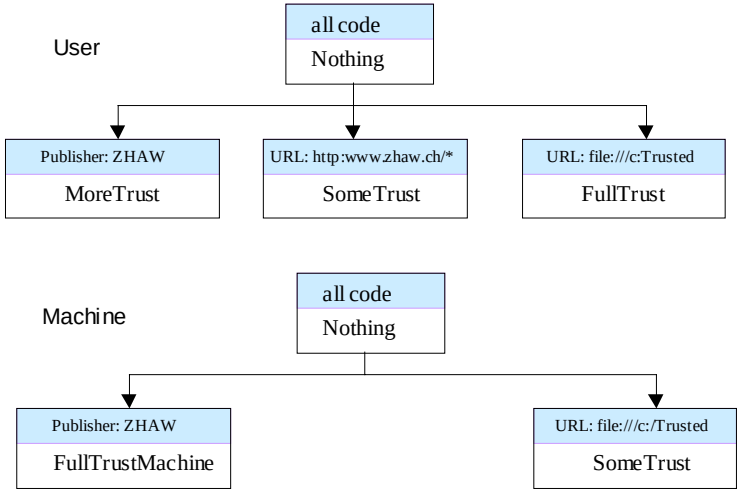
```
1 Public Class RadioClass
2  Static dim power as Boolean;
3
4  Public Sub Main(ByVal args() As String)
5    Dim tc = New RadioClass(true)
6    powerLevel = tc.Test(3)
7    Dim a(,) As String = {"R24","DRS1","DRS2",_
8                          "SWF3","B3","Wannsee"}
9    For i As Integer = LBound(a) To UBound(a)
10      For k As Integer = 0 To a.GetLength(0)
11              Console.WriteLine("Hello " + a(i,k))
12      Next
13   Next
14   Dim j As int
15    Dim d As Double
16    Console.WriteLine("Hello World!" & a[0,0])
17   End Sub

18  Shared Sub Test(x As Integer) As Double
19        x += 1
20        Test = x
21  End Function

22  Public Sub New()
23     power = 0
24  End Sub

25  Enum Radio : Start : Tune : End Enum

26  Public Sub New(ByRef powerLevel As Boolean)
27    MyBase.New()
28    This.power = powerLevel
29  End Sub
30 End Class
```

Korrigieren Sie es auf *möglichst einfache Art* **6 Fehler** ohne die Bedeutung des Programms zu verändern.

Zeile	vorgeschlagene Korrektur

Aufgabe 6: Security (10 Punkte)

Gegegen sei folgende Security Policy auf Stufe Benutzer und Maschine:



Permission Set SomeTrust erlaubt das Lesen von Dateien.

Permission Set MoreTrust, FullTrust und FullTrustMachine erlauben das Lesen und Schreiben von Dateien.

Sie rufen eine Methode in einer EXE Datei direkt von www.zhaw.ch/~rege auf, die wiederum eine Methode einer Assembly A, die sich im Verzeichnis c:/Trusted befindet aufruft. Diese wiederum ruft eine Methode einer von der ZHAW signierten Assembly B auf.

Welche Permission Sets kommen für die EXE-Datei zur Anwendung?

Welche Persmission Sets kommen für die Assembly A zur Anwendung?

Welche Persmission Sets kommen für die Assembly B zur Anwendung?

Werden Sie vom Verzeichnis c:\Trusted lesen können?

Werden Sie ins Verzeichnis c:\Trusted schreiben können?

Aufgabe 7: Vermischte Fragen (10 Punkte)

Kreuzen Sie die beste Antwort an (möglichst vollständig und richtig).

1. Wann wird ein Roundtrip in ASP.NET ausgelöst ?
☐ Jedesmal wenn ein TextFeld, das mit AutoPostBack = true markiert ist, verändert wird.
☐ Jedesmal wenn der "Reload" Knopf im Browser gedrückt wird
☐ Jedesmal wenn ein Knopf im Formular gedrückt wird.
☐ In allen den obigen Fällen.

2. Was bedeutet es, wenn in C# virtual vor eine Property Deklaration steht
☐ geht nicht, ist ein Fehler
☐ das Property kann überschrieben werden
☐ das Property kann überladen werden
☐ das Property kann zwar gelesen aber nicht gesetzt werden

3. Um bei ASP.NET die Korrektheit der Validation zu überprüfen muss man auf der Serverseite:

☐ nichts tun, wird automatisch auf dem Klienten überprüft.
☐ das Property IsValid überprüfen.
☐ die Methode Validate() aufrufen.
☐ die Felder einzeln überprüfen.

4. In welchen der folgenden String Definitionen muss der '\' nicht speziell behandelt werden?
☐ string s = "'n Test string";
☐ string s = #"n Test string";
☐ string s = @"n Test string";
☐ string s = "n Test string";

5. Das Type-Object eines Objekts bekommt man mittels:
☐ dem typeof()-Operator
☐ der GetType()-Methode
☐ dem Type()-Operator
☐ dem is-Operator

6. Wo ist die Versionsinformation einer Assembly gespeichert?
☐ Wird vom GAC verwaltet
☐ Als Wert des const Strings AssemblyVersion
☐ Im <appname>.exe.config File
☐ Im Manifest

7. Wie findet man heraus, dass es sich bei einem EXE um eine .NET Assembly handelt?
☐ Man kann es mit den Standardwerkzeugen nicht herausfinden.
☐ Man überprüft den Hauptspeicherbedarf nach Ausführung; grösser 100 MBytes → .NET
☐ Man wendet ILDASM auf die Assembly an, wenn kein Fehler auftritt → .NET Assembly
☐ Man öffnet die Assembly mit REGEDIT → erster Eintrag →"HKCR = .NET"

8. Was ist der unterschied zwischem ref und out Parameter?
☐ kein Unterschied
☐ ref Parameter werden als Referenz, out als Wert übergeben
☐ ref Parameter müssen beim Aufruf schon initialisiert sein.
☐ out Parameter dürfen beim Aufruf nicht initialisiert sein.

9. Wie definiert man ein nur-lese Property in C#?
☐ mittels vorangestelltem "readonly"
☐ mittels vorangestelltem "const"
☐ geht nicht
☐ der set-Teil wird weggelassen

10.Welcher Typ ist nicht CLS konform?
☐ int
☐ ushort
☐ decimal
☐ eunm

11.Wie bestimmt man die Länge eines Arrays in C#?
☐ Mit dem Count Property
☐ Mit dem Length Property
☐ Mit der Length() Methode
☐ Mit der UBound Methode