

# Klausur WS 2008/2009

## Komponenten und Frameworks

**Füllen Sie bitte zunächst dieses Deckblatt vollständig aus und unterschreiben es.**

Name		Vorname	Matrikelnummer
Semester	Studiengang (I, BI, MI)		HÜ-Zulassung aus Semester
Ich bestätige die Richtigkeit der gemachten Angaben.		Datum	Unterschrift

---

Bearbeiten Sie die Aufgaben bitte auf den folgenden Blättern. Verwenden Sie bei Bedarf auch die Rückseiten. Jede Aufgabe sollte allerdings nur auf dem Blatt bearbeitet werden, auf dem sich die Aufgabenstellung befindet.

Als Hilfsmittel sind zugelassen:

- alle eigenen schriftlichen Unterlagen bzw. Ausdrücke
- sowie maximal zwei Bücher .

Bearbeitungsdauer: **90 Min.**

Viel Erfolg!

Aufgabe	Punkte	Erreicht
1	20	
2	20	
3	20	
4	20	
5	20	
Summe	100	

---

## Aufgabe 1 (Komponenten)

---

- (a) Geben Sie eine Definition des Begriffes *Software-Komponente*! (4)
- (b) Geben Sie eine Definition des Begriffes *Framework*! Was unterscheidet ein Framework von einer Klassenbibliothek? (3)
- (c) Worin unterscheiden sich die Typen **object** und **System.Object**? (3)
- (d) Wie wird in .NET die Plattform- und Sprachenunabhängigkeit erreicht? (4)
- (e) Was versteht man unter einer Assembly? (3)
- (f) Welche Informationen enthält das Manifest einer Assembly? (3)

---

## Aufgabe 2 (Delegates)

---

- (a) Definieren Sie einen Deleгатentyp `XYZHandler` so, dass eine Variable von diesem Typ z.B. auf die Methode

`int[] foo (string str, bool b)` verweisen kann.

(3)

- (b) Wie hängt ein mit dem Schlüsselwort **delegate** definierter Deleгатentyp `XYZHandler` mit der Klasse `System.Delegate` zusammen?

(3)

- (c) Worin unterscheidet sich ein Event-Typ `E` von einem Deleгатen-Typ `D`?

(4)

- (d) Gegeben seien die Deklarationen

```
class C {  
    delegate object D1(string);  
    delegate string D2(object);  
  
    public static void m1(string str){...};  
    public object m2(string str){...};  
    public static int[] m3(string str){...};  
    public static object m4(object o){...};  
    public static string m5(string str){...};  
}
```

Welche der folgenden Methoden `m1` bis `m4` sind mit `D1`, bzw. `D2` verträglich? Begründen Sie!

	Mit D1	Begründung	Mit D2	Begründung
m1				
m2				
m3				
m4				
m5				

(10)

---

## Aufgabe 3 (Remoting)

---

- (a) Nennen Sie die in .NET Remoting vorgesehen drei Arten der Aktivierung eines entfernt Objektes (remote object)! Geben Sie für jeden Art an, *wann*, *wie oft* und *von wem* das entfernte Objekt erzeugt wird! (6)
- (b) Wie lange lebt ein entferntes Objekt (remote object)? Unterscheiden Sie nach den Aktivierungsarten und erläutern Sie den Zusammenhang mit der (verteilten) Speicherbereinigung! (5)
- (c) Was ist eine serialisierbare Klasse? Welche Objekte müssen beim entfernten Methodenaufruf (z.B. in .NET Remoting) serialisierbar sein? (3)
- (d) Ein entfernter Methodenaufruf (remote method call) soll genau so wirken wie ein lokaler. Welche Aufgaben muss die Infrastruktur im Hintergrund erfüllen, damit dies möglich wird? Erläutern Sie anhand des Daten, die vom Client zum Server und zurück übertragen werden. (6)

---

## Aufgabe 4 (DB-Zugriff mit ADO.NET)

---

- (a) Welche beiden Zugriffsarten auf eine Datenbank werden von ADO.NET über die Schnittstelle `IDataReader` bzw. die Klasse `DataSet` angeboten? Erläutern Sie beide Zugriffsarten und geben Sie ihre Vor- und Nachteile an! (4)
- (b) Welche Kommandotypen kann ein Objekt der Klasse `SqlCommand cmd` besitzen? Welche Bedeutung hat dann jeweils die Zeichenkette in `cmd.CommandText`? (4)
- (c) Die Schnittstellen `IDataRecord` und `IDataReader` verwendet man, um auf Ergebnisse von `SELECT`-Anweisungen zugreifen zu können. Welches Entwurfsmuster steht dahinter und mit welchem Ziel wurde es hier eingesetzt? (4)
- (d) Wozu dient ein *connection string*? Welche Information enthält er im Allgemeinen? (4)
- (e) Die Schnittstelle `IDbConnection` erweitert die Schnittstelle `IDisposable`. Warum? Wozu ist die Schnittstelle `IDisposable` überhaupt nötig, es gibt doch eine automatische Speicherbereinigung? (4)

---

## Aufgabe 5 (Threads, Asynchrone Beauftragung)

---

(a) Eine Aufgabe (beschrieben z.B. durch eine Methode `Rfoo(P)`) soll *asynchron* bearbeitet werden: D. h. der Thread, der die Bearbeitung der Aufgabe anstößt, soll direkt weiter arbeiten können und nicht warten müssen, bis die Aufgabe erledigt ist. Erläutern Sie vier verschiedene Wege, auf denen das erreicht werden kann!

(8)

(b) Wozu dient die Klasse `BackgroundWorker`? Ihre Verwendung wird unter bestimmten Umständen in Windows Forms Anwendungen empfohlen. Welche sind das? Erläutern sie!

(4)

(c) Wozu dient die Methode `Control.BeginInvoke(..)` im Namensraum `System.Windows.Forms`? Erläutern Sie!

(5)

(d) Wie unterbricht man einen Thread? Erläutern Sie!

(3)