

Name, Vorname	Klasse	Punkte	Note

Zeit 90 Minuten

Totale Punktzahl 90

Hilfsmittel Beliebige persönliche **Unterlagen, Bücher** und einfacher **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; "weiss nicht" führt dagegen zu keinem Abzug (zu wissen, dass man es nicht weiss, ist auch schon was). Aussagen liegen zum Teil in verneinter Formulierung vor)

ja nein weiss
 nicht

a)

			In ASP.NET kann die Bibliothek der Steuerelemente für die Web Programmierung erweitert werden.
			ASP.NET kann auch mit VB.NET verwendet werden.
			Entwicklung mit Microsoft MVC setzt keine Kenntnisse von JavaScript voraus.
			In ASP.NET lassen sich keine SOAP WebServices implementieren.
			Die Zugriffsberechtigung für einzelne Web Seiten wird in ASP.NET über die Registry gesteuert.

b)

			WinRT unterstützt COM nicht.
			Partielle Klassen sind dasselbe wie abstrakte Klassen.
			Die Implementation der Generics in C# unterscheidet sich nicht zu derjenigen in Java.
			In einer Variablen vom Typ dynamic kann ein beliebiger Wert/Objekt gespeichert werden.
			Bei der Funktionalen Programmierung steht das Lambda für eine nicht weiter benannte Funktion

c)

			VB 6.0 wurde ebenfalls in IL Code übersetzt.
			In VB.NET lassen sich beliebige C# Methoden aufrufen.
			Adressarithmetik ist in C# nicht möglich.
			Werteüberläufe von Integer-Arithmetik führen in C# zwingend zu einem Laufzeitfehler.
			DataSets wenden das pessimistische Lockingverfahren an.

Aufgabe 2: Vererbung(10 Punkte)

Schreiben Sie auf, was das untenstehende VB Programm ausgibt.

```
Interface Car
    Sub WhatAreYou()
End Interface

Interface SportsCar
End Interface

Class GermanCar Implements Car
    Public Overridable Sub WhatAreYou()
        Console.WriteLine("I am a GermanCar")
    End Sub
End Class

Class Golf Inherits GermanCar
    Public Shadows Sub WhatAreYou()
        Console.WriteLine("I am a Golf")
    End Sub
End Class

Class GolfGTI Inherits Golf Implements SportsCar
    Public Shadows Sub WhatAreYou()
        Console.WriteLine("I am a Golf GTI")
    End Sub
End Class

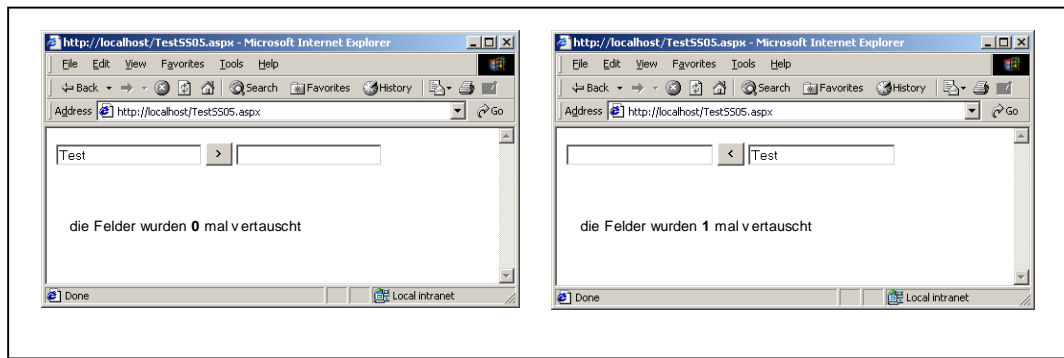
Class Opel Inherits GermanCar
    Public Overridable Shadows Sub WhatAreYou()
        Console.WriteLine("I am an Opel")
    End Sub
End Class

Class Manta Inherits Opel Inherits Golf Implements SportsCar
    Public Overrides Sub WhatAreYou()
        Console.WriteLine("echt cool, du")
    End Sub
End Class

Dim car1 As Car = New GolfGTI
Dim car2 As Car = New Manta
```

	I am
car2.WhatAreYou	
car1.WhatAreYou	
CType(car2, Opel).WhatAreYou	
CType(car1, Golf).WhatAreYou	
CType(car2, Manta).WhatAreYou	
CType(car2, GermanCar).WhatAreYou	

Aufgabe 3: ASP.NET (10 Punkte)



Programmieren Sie den C# CodeBehind mit folgender Funktionalität; um die ASPX Seite müssen Sie sich nicht kümmern; gehen Sie statt dessen von folgenden Annahmen aus: Die beiden Textboxen sollen *Ping* und *Pong*, der Knopf einfach *Button1* und das Label *Label1* heissen.

- Beim ersten Drücken des Knopfes soll der Text in der linken Textbox in die rechte kopiert werden.
- Zusätzlich soll die Anzahl der Vertauschungen (während der Benutzer Session!) im Label wie oben angezeigt werden (inklusive dem Text!)
- Das linke Textfeld soll gelöscht werden.
- Gleichzeitig soll die Beschriftung im Knopf von ">" zu "<" geändert werden.
- Durch nochmaliges Drücken des Knopfes soll der Text vom rechten Textfeld in das linke kopiert werden.
- Das rechte Textfeld soll anschliessend gelöscht werden.
- Gleichzeitig soll die Beschriftung im Knopf von "<" zu ">" geändert werden.
- Obiger Ablauf soll maximal 3 mal während der Session wiederholt werden können.

```
public partial class _TextPingPong : Page {

    protected void Button1_Click(object sender, EventArgs e) {
```

$$\left. \begin{array}{l} \{ \\ \} \end{array} \right\}$$

Aufgabe 4: Parallele Programmierung (10 Punkte)

Damit das Hauptprogramm durch eine Output Operationen nicht verlangsamt wird, soll die (File-) Schreiboperation asynchron jede Sekunde (*im Hintergrund*) erfolgen. Der Aufruf werde über statische Methoden implementiert. Es kann/soll die Queue-Datenstruktur mit den Methoden *Enqueue* und *Dequeue* und dem Property *Count* verwendet werden. Die AsyncWriter-Klasse soll konventionelle mittels Threads implementiert und Thread-Safe sein.

```
public class AsyncWriter {

    Queue<string> outQ;
    String FileName = @"c:\temp\out.txt";

    private static void FlushFile() {

        }

    public static void WriteAsync(String s) {

        }

    static AsyncWriter() {

        }

}
```

Aufgabe 5: Fehlersuche (10 Punkte)

Gegeben seien folgende C# Interface und Klasse

```
1      public interface BasicInterface {
2          int k;
3          public BasicInterface(int k) {k = k;}
4          public void foo (int d){
5              k = (int)d * 4.2;
6              int j = 1.3;
7              return 3 % j * k;
8          }
9      }
10     public class DerivedClass implements BasicInterface {
11         public DerivedClass () {}
12         public int bar(double k) {
13             k = super.foo(k);
14             return (double)k*2;
15         }
16     }
```

Korrigieren Sie 8 Fehler die vom Compiler gemeldet werden ohne den *vermeintlichen* Zweck des Programms zu ändern (mehrere Fehler auf einer Zeile sollen einzeln korrigiert werden). *Vorsicht: falsch angezeigte Fehler führen zu Abzug*

Zeile	Vorgeschlagene Korrektur

Aufgabe 6: Operator Overloading (10 Punkte)

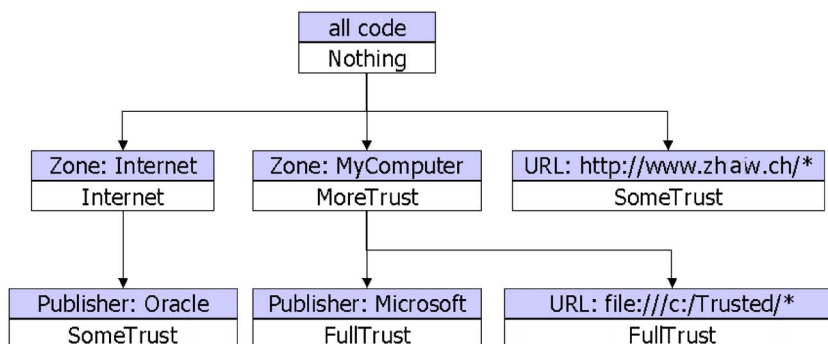
Entwickeln Sie einen Datentyp `Complex` (als Wertetyp), der Ihnen das Rechnen mit komplexen Zahlen ermöglicht. Real- und Imaginärteile sollen vom Typ `double` sein. Es sollen folgende Operationen unterstützt werden:

- a) Bei der (impliziten) Zuweisung eines `double` Wertes soll eine neue komplexe Zahl erzeugt werden.
- b) Bei der (expliziten) Zuweisung eines komplexen Wertes zu einer `double` Variablen soll der Betrag d.h. die Länge in der Gaußsschen Zahlenebene bestimmt werden.
- c) Die Addition soll unterstützt werden.
- d) Es sollen zwei komplexe Zahlen auf Gleichheit überprüft werden können (beide Teile müssen gleich sein).

Aufgabe 7: Security (10 Punkte)

Gegeben sei folgende Sicherheitspolicy mit den dazugehörigen Permission Sets.

- C: und D: seien lokale Laufwerke und N: M: und Z seien Netzwerklaufwerke.
- MyApp.Exe wird direkt von der ZHAW Homepage geladen
 - Die Aufrufkette `MyApp.Main` → `FileUtil.Process` → `FileBase.Process`.
- ForeignApp.Exe wird direkt von `http://www.natasha.ru` geladen
 - Die Aufrufkette `ForeignApp.Main` → `FileUtil.Process` → `FileBase.Process`.
- FileUtil.dll befinde sich im Verzeichnis `c:\Windows` und stamme von Microsoft (ist signiert)
- FileBase.dll befinde sich im Verzeichnis `c:\Trusted`



Permission Sets	<div>Internet FilePermission: Read: c:\sandbox</div> <div>SomeTrust FilePermission: Write: c:\sandbox , c:\temp Read: local</div> <div>MoreTrust FilePermission: Write: local, Read: local</div> <div>FullTrust FilePermission: Write: all, Read: all</div>
-----------------	---

Hinweis: Zone MyComputer -> Lokale Laufwerke Internet -> beliebige URL

von welchen Verzeichnissen und Laufwerken können

a) Über die Klasse MyApp gelesen werden?

b) Über die Klasse MyApp geschrieben werden?

c) Über die Klasse ForeignApp gelesen werden?

d) Über die Klasse ForeignApp geschrieben werden?

Aufgabe 8 Delegates, Lambda Ausdrücke (10 Punkte)

Gegeben sei:

```
delegate double MConverter(double x);

IList<double> convert(IList<double> km, MConverter f) {
    IList<double> res = new List<double> ();
    foreach (double d in data) {
        res.add(f(d));
    }
    return res;
}

IList<double> km = new List<double> {6,10,15,30,50,80,100,110,120};
```

Schreiben Sie ein C# Code Fragment, so das durch Aufruf obiger `convert` Methode eine Liste von Geschwindigkeiten (Kilometer/Stunde) Werten in eine neue Liste von Meilen pro Stunde Werten umgewandelt wird (1 Meile \approx 1.6 km)

a) unter Verwendung einer Methode der eigenen Klasse.

b) unter Verwendung einer anonymen Methode.

c) unter Verwendung eines Lambda Ausdrucks.

d) Gegeben seien folgende Benzinverbräuche in Liter pro 100 km (Prius, Golf, Hummer; MRAP)

```
IList<double> fuelConsumption = new List<double> {4, 5, 26, 44};
```

Schreiben Sie den Lambda Ausdruck, der unter Anwendung obiger `convert` Methode eine Liste von Miles/Gallone Werten berechnet (1 Gallone \approx 3.8l).

Aufgabe 9: Vermischte Fragen (10 Punkte)

Es ist jeweils nur **eine Antwort** vollständig und richtig, d.h. kreuzen Sie die beste Antwort an. Falsche oder mehrfache Antworten führen zu Abzug. Wenn nicht anders angegeben, beziehen sich die Fragen auf C#.

1. Welche Methoden sind in VB.NET ohne Instanzierung aufrufbar?

- ☐ *Static* Methoden
- ☐ *Shared* Methoden
- ☐ *Internal* Methoden
- ☐ Overloaded Methoden

2. Wie gross ist der double Datentyp in .NET?

- ☐ 16 Bit
- ☐ 32 Bit
- ☐ 64 Bit
- ☐ Abhängig von der Rechnerplattform

3. Um in ASP.NET zu überprüfen, ob ein Feld eine gültige, nicht leere E-Mail Adresse enthält, muss man:

- ☐ einen RegularExpressionValidator einsetzen.
- ☐ einen RequiredFieldValidator einsetzen.
- ☐ einen CompareValidator einsetzen.
- ☐ einen RegularExpressionValidator und einen RequiredFieldValidator einsetzen.

4. Welches Aussage ist korrekt?

- ☐ Virtuelle Methoden können *statisch* deklariert sein
- ☐ Eine Klasse darf gleichzeitig als *sealed* und als *abstract* deklariert sein
- ☐ Virtuelle Methoden können *abstract* deklariert sein
- ☐ Keine der obigen Aussagen

5. Welcher Datentyp ist nicht Teil des CTS?

- ☐ decimal
- ☐ int
- ☐ ushort
- ☐ alle sind Teil des CTS

6. Delegates werden verwendet für:

- ☐ Schnittstellenbeschreibung und Ereignisbenachrichtigung
- ☐ Ereignisbenachrichtigung und Rückruf-Methoden
- ☐ Reflection-Methoden und Rückruf-Methoden
- ☐ Rückruf-Methoden

7. Wo werden Wertetypen gespeichert?

- ☐ In einer Global Assembly Cache
- ☐ Auf dem Heap
- ☐ Auf dem Stack
- ☐ In einer Queue

8 Was passiert, wenn der Wertebereich eines Fließkommazahlentyps überschritten wird

- ☐ Das Vorzeichen wechselt.
- ☐ Es wird eine Overflow Exception ausgelöst.
- ☐ Hängt von der Implementation der CLR ab.
- ☐ nichts von all dem

9. Um auf eine IP V4 Adresse zu prüfen, müssen Sie folgenden Ausdruck im RegularExpressionValidator verwenden:

- ☐ \d{3}.\d{3}.\d{3}.\d{3}
- ☐ \d{3}-\d{3}-\d{3}-\d{3}
- ☐ \d{3}\.\d{3}\.\d{3}
- ☐ \d{3}(\.\d{3}){3}

10. Um die Performance bei parallelem, nur lesendem Datenbankzugriff zu erhöhen, kann man:

- ☐ pessimistisches Locking verwenden
- ☐ den Isolation Level auf Read Uncommitted setzen
- ☐ den Isolation Level auf Serializable setzen
- ☐ einen anderen Datenbanktreiber verwenden

11. Was wird als *Unsafe Code* bezeichnet?

- ☐ Dasselbe wie Unmanaged Code.
- ☐ Code, der ein Sicherheitsproblem enthält.
- ☐ Code, der in einem Block der mit *Unsafe* markiert wurde steht.
- ☐ Der Internet Explorer
- ☐ Sämtlicher Code, der von Microsoft stammt.

12. Ein Microsoft Evangelist kann durchaus auch folgendes sein:

- ☐ Ein Sarkastiker
- ☐ Ein Zyniker
- ☐ Ein Agnostiker
- ☐ Ein Atheist
- ☐ Alles, wenn Microsoft nur genug dafür bezahlt