

Wiederholungsklausur "C#" SS 2013

<i>Nachname, Vorname</i>	
<i>Abschluss (BA, MA, FKN etc.)</i>	
<i>Matrikelnummer, Semester</i>	
<i>Versuch (1/2/3)</i>	

Bitte füllen Sie zuerst den Kopf des Angabenblattes aus!

Die Klausur dauert 90 Minuten.

Es sind maximal 45 Punkte zu erreichen.

Es sind keine Hilfsmittel zugelassen.

Bitte beantworten Sie alle Fragen direkt auf das Angabenblatt.

Nutzen Sie ggf. die Rückseite und kennzeichnen Sie dies entsprechend.

Eigene Schmierblätter sind nicht erlaubt.

Bei mehreren oder mehrdeutigen Lösungen wird die schlechtere Lösung gewertet. Streichen Sie daher ungültige Lösungen eindeutig durch.

Verwenden Sie nur *C#* für Programmieraufgaben.

Viel Erfolg!

Aufgabe 1:

(3 Punkte)

Nennen Sie insgesamt drei formale und/oder inhaltliche Gemeinsamkeiten bzw. Unterschiede zwischen den Datentypen **short**, **int** und **long**.

Aufgabe 2:

(5 Punkte)

Finden Sie syntaktische und semantische Fehler in folgendem Programmfragment. Unterstreichen Sie die Fehler und begründen Sie, warum ein Fehler vorliegt. Ein Fehlertyp, der mehrfach auftritt, wird nur einmal gewertet.

```
// Test auf Reversität zweier Strings:  
// Zwei Strings sind revers, wenn S die Umkehrung von T ist  
// z. B. sind S = "abc" und T = "cba" Reverse voneinander  
static int AreReverses(string S, string T)  
{  
    if (S.Length() == T.Length())  
    {  
        for (int I = 0, I < S.Length(), I++)  
        {  
            if (S[I] != T[T.Length()-I]) return false;  
        }  
    }  
    else return false;  
}
```

Aufgabe 3:**(6 Punkte)**

Ermitteln Sie, welche Werte die folgende Funktion für die Parameter $N = 0, N = 1, \dots, N = 5$ berechnet und zurückgibt. Tragen Sie das Ergebnis in die darunter stehende Tabelle ein.

```
// Funktion F, die für ihren Parameter N bestimmte Werte berechnet
static int F(int N)
{
    int X = 0;           // Funktionsergebnis, das unten gesetzt wird
    while (N > 0)       // kreise in Schleife, solange N > 0
    {
        if (N % 3 == 0) // wenn N restlos durch 3 teilbar ist
            N = N / 3 - 1; // teile N durch 3 und subtrahiere 1
        else           // sonst
            N = N + 1; // erhöhe N um 1
        X++;          // zähle mit in X, wie oft Schleife durchlaufen wird
    }
    return X;        // gib Anzahl Schleifendurchläufe zurück
}
```

N	0	1	2	3	4	5
F(N) (bzw. X)						

Aufgabe 4:

(8 Punkte)

Schreiben Sie eine Funktion `char[,] Reverse(char[,] Matrix)`, die den Inhalt der *quadratischen* Zeichenmatrix `Matrix` (zweidimensionales Array aus `char`) umkehrt, d. h. das linke obere Zeichen wird mit dem rechten unteren vertauscht, das zweite obere mit dem zweitletzten unteren usw. Die Ausdehnung der Matrix in der 1. Dimension kann über `Matrix.GetLength(0)` ermittelt werden (da die Matrix quadratisch ist, ist die vertikale und horizontale Größe gleich). Als Ergebnis soll eine neue Matrix angelegt und zurückgegeben werden, die die Reversion von `Matrix` enthält.

Beispiel für 3×3-Matrix:

'1'	'2'	'3'
'4'	'5'	'6'
'7'	'8'	'9'

 →

'9'	'8'	'7'
'6'	'5'	'4'
'3'	'2'	'1'

Beispiel für 4×4-Matrix:

'A'	'B'	'C'	'D'
'E'	'F'	'G'	'H'
'I'	'J'	'K'	'L'
'M'	'N'	'O'	'P'

 →

'P'	'O'	'N'	'M'
'L'	'K'	'J'	'I'
'H'	'G'	'F'	'E'
'D'	'C'	'B'	'A'

Aufgabe 5:

(8 Punkte)

Schreiben Sie eine Funktion `bool IsReduplicated(string S)`, die den übergebenen String `S` daraufhin testet, ob zweimal direkt hintereinander die selbe Zeichenfolge erscheint. Beispiele für reduplizierte Strings: "blabla", "jaja", "11" usw. Überlegen Sie, welche Bedingungen `S` erfüllen muss, damit es überhaupt einen reduplizierten String enthalten kann. Die Groß- und Kleinschreibung braucht nicht beachtet zu werden.

Aufgabe 6:

(15 Punkte)

Definieren Sie eine Klasse `Number` mit folgenden Attributen und Methoden (inkl. Instanzkonstruktor):

- privates ganzzahliges *Attribut* `MyNumb`: enthält einen Wert, der von unten stehenden Methoden benutzt wird.
- öffentlicher *Instanzkonstruktor* `Number(int N)`: `MyNumb` soll bei der Instanziierung (d. h. bei der potenziellen Erzeugung einer Instanz der Klasse `Number` per Deklaration und Initialisierung durch den `new`-Operator) den Wert von `N` erhalten.
- öffentliche boolesche *Methode* `IsZero()`: Die Methode soll `true` zurückgeben, wenn `MyNumb` den Wert 0 aufweist.
- öffentliche boolesche *Methode* `IsNegative()`: Die Methode soll `true` zurückgeben, wenn `MyNumb` einen Wert kleiner als 0 aufweist.
- öffentliche boolesche *Methode* `IsEven()`: Die Methode soll `true` zurückgeben, wenn `MyNumb` eine gerade Zahl ist.

Schreiben Sie den vollständigen Klassenrahmen hin und implementieren Sie das einzige Attribut sowie den Konstruktor und die drei Methoden. Eine konkrete Instanz muss nicht erzeugt werden.