

PD Dr. J. Reischer

4.2.2015

## Klausur "C#" WS 2014/2015

<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 30 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!**

## Teil I: Konzeptionelles

### Aufgabe 1:

(4 Punkte)

Kreuzen Sie die korrekten Aussagen über C# an. Falsch angekreuzte Antworten geben Punktabzug innerhalb dieser Aufgabe. Achtung: Es gibt nicht unbedingt genau 1 Punkt pro korrekt gelöster Aussage!

- Der `int`-Typ kann keine negativen Zahlen darstellen.
- `int` ist der einzige Typ, der bei `for`-Schleifen verwendet werden kann.
- `short`-Werte passen alle in den `int`-Bereich.
- Der `char`-Typ ist mit `int` kompatibel.
- In `for`-Schleifen kann man nur aufwärts zählen.
- Eine `int`-Variable kann immer nur um 1 erhöht oder erniedrigt werden.
- Zwei `int`-Werte können dividiert werden.
- `int` ist der größte Ganzzahltyp.

## Teil II: Funktionsanalyse

### Aufgabe 2:

(6 Punkte)

Analysieren Sie unten stehende Funktion F dahingehend, welche Parameterwerte X welche Ergebniswerte F(X) erzeugen. Tragen Sie die Funktionsergebnisse in unten stehende Tabelle ein (zwei Versuche, sonst Tabelle selbst erweitern; streichen Sie Fehlversuche deutlich durch).

```
// Funktion F mit Ganzzahl-Parameter X
int F(int X)
{
    int Y = 0;
    while (X > Y)
    {
        Y++;
        X--;
    }
    return X;
}
```

Lösungstabelle:

X	0	1	2	3	4	5
F(X)						
F(X)						

### Teil III: Entwicklung

#### Aufgabe 3:

(4 Punkte)

Schreiben Sie eine Funktion `string MultiChar(char C, int N)`, die ein Zeichen C N-Mal hintereinander anfügt und als String zurückgibt.

Beispiel:

C = 'A', N = 2: Ergebnis = "AA"

C = 'x', N = 5: Ergebnis = "xxxxx"

C = '?', N = -1: Ergebnis = ""

**Aufgabe 4:**

(5 Punkte)

Schreiben Sie eine Funktion `bool Ascending(int[] Numbers)`, die prüft, ob alle Zahlen im Array `Numbers` in aufsteigender Reihenfolge sortiert sind. Die Funktion soll `true` zurückliefern, wenn alle Elemente sortiert sind, sonst `false`. Die Elemente sind aufsteigend sortiert, wenn das rechte Element nicht kleiner ist als das links davon.

*Beispiele:*

sortiert: {1, 2, 4, 8, 16, 32, 64}

{1, 1, 2, 2, 4, 4, 8}

unsortiert: {1, 2, 4, 16, 8, 32, 64}

**Aufgabe 5:**

(5 Punkte)

Schreiben Sie eine Funktion `string ExtractLetters(string S)`, die aus einem String `S` alle englischen Groß-/Kleinbuchstaben von 'A' bis 'Z' (ohne Umlaute) in der Original-Reihenfolge extrahiert und daraus einen neuen String generiert, der als Ergebnis zurückgegeben wird. Achten Sie darauf, dass es einfacher ist, einen neuen String aufzubauen, als das bestehende `S` zu verändern.

*Beispiele:*

`S = "A1b2C3d4E#"`: Ergebnis `"AbCdE"`

`S = "12 3_4#"`: Ergebnis `""`

**Aufgabe 6:**

(6 Punkte)

Schreiben Sie eine Funktion `int[,] CreateArray(int N)`, die ein quadratisches 2D-`int`-Array der Größe  $N \times N$  erzeugt und aufsteigend zeilenweise die Zahlen von 0 bis  $N^2 - 1$  einträgt. Achten Sie auf die Grenzfälle für  $N$ .

Beispiel:

$N = 2$ :

0	1
2	3

$N = 3$ :

0	1	2
3	4	5
6	7	8

$N = 4$ :

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15