

Klausur "C#" SS 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. Java- und anderer Code wird als Fehler gewertet.

Viel Erfolg!

Teil I: Konzeptionelles

Aufgabe 1:

(3 Punkte)

Hinweis: Die Anzahl korrekter Antworten stimmt nicht notwendigerweise mit der Anzahl vergebener Punkte überein, d. h. eine Antwort kann auch mehr oder weniger als 1 Punkt ergeben. Falsche Antworten führen zu entsprechendem Punkteabzug; es können jedoch insgesamt nicht weniger als 0 Punkte für Aufgabe 1 erzielt werden. Fehlende Antworten werden weder positiv noch negativ gewertet. Setzen Sie ein deutliches Kreuz zur Kennzeichnung einer Antwort; zur Ungültigmachung einer Antwort füllen Sie das Quadrat komplett aus.

- Ein **bool**-Wert kann nicht in einen String verwandelt werden.
- Eine **while**-Schleife kann auch 0 Mal durchlaufen werden.
- Eine **foreach**-Schleife wird immer mind. 1 Mal durchlaufen.
- Ein **switch** kann auch mit einem **bool**-Wert erfolgen.
- Arrays müssen mind. 1 Wert enthalten bzw. Ausdehnung 1 besitzen.
- Arrays können nur aus Ganzzahltypen erzeugt werden.
- Objekttypen werden durch **class** definiert.
- Die Begriffe 'Instanz' und 'Klasse' bezeichnen dasselbe.

Teil II: Funktionsanalyse

Aufgabe 2:

(5 Punkte)

Analysieren Sie unten stehende Funktion F dahingehend, welche Parameterwerte N welche Ergebniswerte $F(N)$ 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 N
int F(int N)
{
    do
    {
        N = N / 2;
        N = N * 3 + 1;
    }
    while (N <= 10);
    return N;
}
```

Lösungstabelle:

N	2	4	6	8	10
F(N)					
F(N)					

Teil III: Entwicklung

Aufgabe 3:

(4 Punkte)

Schreiben Sie eine Funktion `int[] ToUnicode(char[] Z)`, die ein Char-Array `Z` erhält und dieses als Integer-Array der Unicode-Werte der einzelnen Zeichen zurückgibt.

Hinweis: `Z` kann beliebige Chars enthalten, nicht nur Buchstaben und Ziffern!

Beispiele:

`Z[] = {'A', 'B', 'C', 'D', 'E'}:` Ergebnis = {65, 66, 67, 68, 69}

`Z[] = {'0'}:` Ergebnis = {48}

`Z[] = {}:` Ergebnis = {}

Aufgabe 4a:

(5 Punkte)

Schreiben Sie eine Funktion `string Extract(string S, int N)`, die aus einem String jedes N-te Zeichen beginnend mit dem 0-ten extrahiert und zu einem neuen String zusammensetzt, der als Ergebnis zurückgegeben werden soll. Bei nicht sinnvollen Werten für N soll das Resultat ein Leerstring sein.

Beispiele:

S = "\$€%", N = 1: Ergebnis = "\$€%"
S = "abcdef", N = 2: Ergebnis = "ace"
S = "0123456789", N = 3: Ergebnis = "0369"

Aufgabe 4b:

(1 Punkte)

Beschreiben und begründen Sie kurz, welche Werte für N nicht sinnvoll sind und wie dies oben ausgeschlossen wurde.

Aufgabe 5:

(6 Punkte)

Schreiben Sie eine Funktion `char[,] Convert(char[][] Z)`, die ein ausgefranztes 2D-Array `Z` aus `L` Zeilen und `K` Spalten erhält (d. h. immer gleiche Anzahl Spalten pro Zeile, aber nicht unbedingt quadratisch) und in ein rechteckiges mit gleichem Array-Inhalt verwandelt, das dann zurückgegeben werden soll. `Z` enthält mind. 1 Zeile und 1 Spalte, d. h. `Z` hat nicht 0 Zeilen oder 0 Spalten.

Aufgabe 6:

(6 Punkte)

Definieren Sie eine Klasse `Pixel` mit folgenden Members:

- zwei öffentliche `int`-Variablen `X` und `Y`, die die Koordinate des Pixels darstellen;
- ein öffentlicher Konstruktor, der die beiden `int`-Variablen mit übergebenen Werten initialisiert;
- eine öffentliche Methode `Pixel Clone()`, die eine echte Kopie der jeweiligen `Pixel`-Instanz zurückgibt.