Universität Regensburg

PD Dr. J. Reischer 13.02.2013

Klausur "C#" WS 2012/2013

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

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 `if` und `switch`.

Aufgabe 2: (6 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 Palindrom: Ein Palindrom ist eine Zeichenkette, die vorwärts
// wie rückwärts gelesen identisch ist, d. h. symmetrisch bezüglich der
// Mitte ist (z. B. "", "X", "121", "ABBA", "RELIEFPFEILER")
static boole IsPalindrom(string S)
{
  if (S.Length() = 0)
    return true;
  else
    if (S.Length() = 1)
      return true;
    else
    {
      N = S.Length();
      for (int I = 0; I <= N; I++)
        if (S[I] != S[N-I]) return false;
      return true;
    };
};
```

Aufgabe 3: (6 Punkte)

Ermitteln Sie, welche Werte die folgende Funktion für die Parameter N = 0, N = 1, …, 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 C = 0;            // Funktionsergebnis, das unten gesetzt wird
  while (N > 0)         // kreise in Schleife, solange N > 0
  {
    if (N % 2 == 0)     // wenn N gerade ('%' = modulo = Divisionsrest)
      N = N / 2 - 1;    // teile N durch 2 und subtrahiere 1
    else                // sonst
      N = N + 1;        // erhöhe N um 1
    C++;                // zähle mit in C, wie oft Schleife durchlaufen wird
  }
  return C;             // gib Anzahl Schleifendurchläufe zurück
}
```

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

Aufgabe 4: (8 Punkte)

Schreiben Sie eine Funktion `string Convert(char[][] Matrix)`, die den Inhalt der *quadratischen* Zeichenmatrix `Matrix` (zweidimensionales Array aus `char`) rückwärts, d. h. von unten nach oben und rechts nach links entgegen der üblichen Leserichtung in einen String wandelt und als Funktionsergebnis zurückgibt (s. Beispiele unten). Die Matrix kann beliebig groß sein außer 0×0. Die Ausdehnung der Matrix in einer Dimension kann mit `Matrix.Length` ermittelt werden (da die Matrix quadratisch ist, ist die vertikale und horizontale Größe gleich).

Beispiel für 3×3-Matrix:

'R'	'E'	'B'
'M'	'E'	'T'
'P'	'E'	'S'

Ergebnisstring:
"SEPTEMBER"

Beispiel für 4×4-Matrix:

'N'	'I'	'L'	'O'
'R'	'A'	'C'	'T'
'B'	'E'	'I'	'L'
'S'	'N'	'A'	'H'

Ergebnisstring:
"HANSLIEBTCAROLIN"

Aufgabe 5: (8 Punkte)

Schreiben Sie eine Funktion `string ReplaceChars(string S)`, die im übergebenen String S alle großen und kleinen Umlaute sowie das scharfe 'ß' durch entsprechende Umschreibungen ersetzt: 'ä' → "ae", 'ö' → "oe", 'ü' → "ue", 'Ä' → "Ae", 'Ö' → "Oe", 'Ü' → "ue", 'ß' → "sz". Verändern Sie hierzu nicht S selbst, sondern bauen Sie sukzessive einen neuen String (z. B. R) auf, der 'normale' Zeichen aus S übernimmt und bei Umlauten/'ß' die jeweilige Ersatzdarstellung anfügt.

Beispiel: "bärbeißig" ⇒ "baerbeiszig", "Übergröße" ⇒ "Uebergroesze" usw.

Aufgabe 6: (14 Punkte)

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

- *Attribut* `private string MyWord`: enthält eine Zeichenkette, die ein Wort darstellt, das dann durch die unten stehenden Methoden verarbeitet werden soll.
- *Methode* `public bool IsCapital()`: Die Methode (Funktion) soll `true` zurückgeben, wenn alle Zeichen in `MyWord` nur aus Großbuchstaben bestehen.
- *Methode* `public bool HasVowel()`: Die Methode (Funktion) soll `true` zurückgeben, wenn in `MyWord` mindestens ein Vokal (auch Umlaut) enthalten ist.
- *Instanzkonstruktor* `public Word(string S)`: `MyWord` soll bei der Instanzierung (d. h. bei der potenziellen Erzeugung einer Instanz der Klasse `Word` per Deklaration und Initialisierung durch `new`-Operator) den Wert von `S` erhalten.

Schreiben Sie zunächst nur den vollständigen Klassenrahmen `class Word {…}` hin, d. h. das Attribut sowie die beiden Methoden- und den Konstruktor-Rahmen (also jeweils nur Kopf und Rumpf der Methode bzw. des Konstruktors ohne eigentlichen Algorithmus). Sofern Sie noch Zeit haben, können Sie auch noch die entsprechende Funktionalität (Algorithmus) ausprogrammieren. Sie brauchen keine konkrete Instanz zu erzeugen.