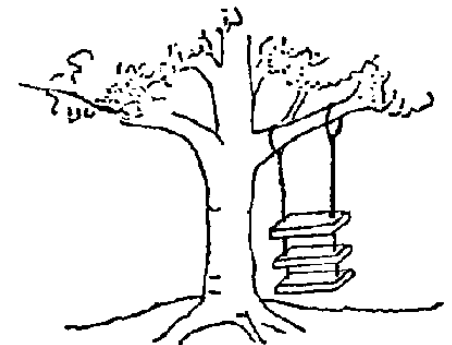


Funktionen Beispiele

Software Entwicklung



Überblick

- Taschenrechner 2 Zahlen
- Celsius - Fahrenheit
- Taschenrechner mit beliebig vielen Zahlen
- Wochentage

- Tic Tac Toe
- Schach - Chessboard



Taschenrechner 2 Zahlen

Aufgabe Taschenrechner:

- Erstelle ein Menü für einen Taschenrechner in der Main, Tätige die Ausgabe es Ergebnisses in der Main
- Auswahl für
 - Addieren, Subtrahieren, Dividieren oder Multiplizieren
- Der Benutzer soll 2 Zahle eingeben können. Das Eingeben der Zahlen soll ein eine Funktion ausgelagert sein.
- Die Berechnung mit den beiden Zahlen soll jeweils in einer Funktion stattfinden.

Taschenrechner mit 2 Zahlen

```
static int GetNumber() {  
    Console.WriteLine("Bitte geben Sie eine Zahl ein");  
    int number = Int32.Parse(Console.ReadLine());  
    return number;  
}  
  
static int AddNumbers(int x, int y) {  
    return x + y;  
}  
static int SubNumbers(int x, int y) {  
    return x - y;  
}  
static int MulNumbers(int a, int b) {  
    return a * b;  
}  
static double DivNumbers(int num1, int num2) {  
    return num1 / num2;  
}
```

```
static void Main(string[] args) {  
    Console.WriteLine("Bitte wählen Sie zwischen Add/Mul/Div/Sub");  
  
    string read = Console.ReadLine();  
    int number1 = GetNumber();  
    int number2 = GetNumber();  
    double result;  
    switch (read) {  
        case "Add":  
            result = AddNumbers(number1, number2); break;  
        case "Sub":  
            result = SubNumbers(number1, number2); break;  
        case "Mul":  
            result = MulNumbers(number1, number2); break;  
        case "Div": {  
            result = DivNumbers(number1, number2); break;  
        }  
        default:  
            Console.WriteLine("Bitte wählen Sie Add/Sub/Mul oder Div");  
            result = 0.0;  
            break;  
    }  
  
    Console.WriteLine("Das Ergebnis lautet {0}", result);  
}
```



Celsius - Fahrenheit

Gliedere die Funktionen noch besser als in der Musterlösung...

Beispiel: Celsius - Fahrenheit

- Erstelle ein Programm, das im Menü abfragt ob von Celsius zu Fahrenheit oder von Fahrenheit zu Celsius gerechnet werden soll.
- Anschließend wird in einem Switch-Case
 - die Temperatur eingelesen
 - die Berechnung zum Umrechnen aufgerufen
 - die Ausgabe getätigt.
- zum Berechnen gibt es zwei statische Funktionen:
 - `int CelsiusToFahrenheit(int temp);`
 - $\text{fahrenheit} = (\text{celsius} * 9 / 5) + 32;$
 - `int FahrenheitToCelsius(int temp);`
 - $\text{celsius} = (\text{fahrenheit} - 32) * 5 / 9;$

Funktionen zum Berechnen

```
public static double CelsiusToFahrenheit(string temperatureCelsius)
{
    // Convert argument to double for calculations.
    double celsius = Double.Parse(temperatureCelsius);
    // Convert Celsius to Fahrenheit.
    double fahrenheit = (celsius * 9 / 5) + 32;
    return fahrenheit;
}
```

```
public static double FahrenheitToCelsius(string temperatureFahrenheit)
{
    // Convert argument to double for calculations.
    double fahrenheit = Double.Parse(temperatureFahrenheit);
    // Convert Fahrenheit to Celsius.
    double celsius = (fahrenheit - 32) * 5 / 9;
    return celsius;
}
```



```
static void Main(string[] args)
```

```
{
```

```
    Console.WriteLine("Please select the convertor direction");  
    Console.WriteLine("1. From Celsius to Fahrenheit.");  
    Console.WriteLine("2. From Fahrenheit to Celsius.");  
    Console.Write(":");
```

```
    string selection = Console.ReadLine();  
    double F, C = 0;
```

```
    switch (selection)  
    {
```

```
        case "1":
```

```
            Console.Write("Please enter the Celsius temperature: ");  
            F = CelsiusToFahrenheit(Console.ReadLine());  
            Console.WriteLine("Temperature in Fahrenheit: {0:F2}", F);  
            break;
```

```
        case "2":
```

```
            Console.Write("Please enter the Fahrenheit temperature: ");  
            C = FahrenheitToCelsius(Console.ReadLine());  
            Console.WriteLine("Temperature in Celsius: {0:F2}", C);  
            break;
```

```
        default:
```

```
            Console.WriteLine("Please select a convertor.");  
            break;
```

```
}
```

Main mit Aufruf der Funktionen



Taschenrechner - Array

Taschenrechner Add / Mul mit beliebig vielen Zahlen

- Verändern Sie den Taschenrechner so, dass der Benutzer die Anzahl der Werte, die er summieren, subtrahieren, multiplizieren oder dividieren möchte angibt. Füllen Sie ein Array mit den eingelesenen Werten. Führen Sie die Berechnung laut Wunsch des Benutzers durch: Addieren oder Multiplizieren.

```
Bitte wählen Sie zwischen Add/Mul
Add
Wieviele Werte möchten Sie eingeben?
Bitte geben Sie eine Zahl ein: 4

Bitte geben Sie eine Zahl ein: 2

Bitte geben Sie eine Zahl ein: 3

Bitte geben Sie eine Zahl ein: 6

Bitte geben Sie eine Zahl ein: 2

Ergebnis lautet: 13
```

Optional kann die Funktionalität um Dividieren und Subtrahieren ergänzt werden.

Array als Parameter

- Zahlen als Parameter
- Berechnung Add/Mul durchführen
- Ergebnis retour geben

```
static int AddNumbers(int[] arr) {  
    int sum = 0;  
    for (int i = 0; i < arr.Length; i++) {  
        sum += arr[i];  
    }  
    return sum;  
}  
  
static int MulNumbers(int[] arr) {  
    int sum = 0;  
    for (int i = 0; i < arr.Length; i++) {  
        sum *= arr[i];  
    }  
    return sum;  
}
```

Array als Rückgabewert

- Erstelle ein Array beliebiger Größe, befülle das Array mit Werten des Benutzers

```
static int[] GetArray(int amount) {  
    int[] arr = new int[amount];  
    for (int i = 0; i < arr.Length; i++) {  
        arr[i] = GetNumber();  
    }  
    return arr;  
}
```

```
static int GetNumber() {  
    Console.Write("Bitte geben Sie eine Zahl ein: ");  
    int number = Int32.Parse(Console.ReadLine());  
    Console.WriteLine();  
    return number;  
}
```

Ausgabe einfach

```
static void PrintArray(int[] arr) {  
    for (int i = 0; i < arr.Length; i++) {  
        Console.Write($"{arr[i]}, ");  
    }  
}
```

5, 6, 6, 3,

Ausgabe besonders

3 + 4 + 5 + 6 = Ergebnis lautet: 18

- Ausgabe des Arrays mit:
num1+num2+num3 = result
 - Angepasst für Add/Mul
 - Letzter Wert wird mit = ausgegeben

```
static void PrintArray(int[] arr, char c) {  
    for (int i = 0; i < arr.Length-1; i++) {  
        Console.Write($"{arr[i]} {c} ");  
    }  
    Console.Write($"{arr[arr.Length-1]} = ");  
}
```

Main

```
public static void TestArray() {  
    Console.WriteLine("Bitte wählen Sie zwischen Add/Mul");  
    string read = Console.ReadLine();  
    Console.WriteLine("Wieviele Werte möchten Sie eingeben?");  
    int amount = GetNumber();  
    Console.WriteLine();  
    int[] arr = GetArray(amount);  
    double result;  
    switch (read) {  
        case "Add":  
            result = AddNumbers(arr);  
            PrintArray(arr, '+'); break;  
        case "Mul":  
            result = MulNumbers(arr);  
            PrintArray(arr, '*'); break;  
        default:  
            Console.WriteLine("Falsche Eingabe");  
            result = 0; break;  
    }  
    Console.WriteLine($"Ergebnis lautet: {result}");  
    PrintArray(arr);  
}
```

```
Bitte wählen Sie zwischen Add/Mul  
Add  
Wieviele Werte möchten Sie eingeben?  
Bitte geben Sie eine Zahl ein: 4  
  
Bitte geben Sie eine Zahl ein: 3  
Bitte geben Sie eine Zahl ein: 4  
Bitte geben Sie eine Zahl ein: 5  
Bitte geben Sie eine Zahl ein: 6  
  
3 + 4 + 5 + 6 = Ergebnis lautet: 18
```




Wochentag

Eingabe im Programm: (4 Zahlenwerte)

Wochentag: 1

Tag: 3

Monat: 11

Jahr 2014

Ausgabe in der Konsole:

Montag der 3. November 2014

Wochentag lang

- Erstelle ein Programm, das 4 Werte einliest: Wochentag Montag = 1, Sonntag = 7, Tag im Monat mit Werten zwischen 1 und 31, Monat zwischen 1 und 12 für Jänner bis Dezember und die Jahreszahl.
- Löse die Wochentage mit if/else und die Monate mit switch case. Setze alles in eine Schleife, dass der Benutzer beliebig lange (bis zur Abbruchbedingung) das Programm ausführen kann.
- Diese Werte sollen nun als gesamtes langes Datumsformat in der Console ausgegeben werden.

Wochentag lang

- **Eingabe im Programm: (4 Zahlenwerte)**
 - Wochentag: 1
 - Tag: 3
 - Monat: 11
 - Jahr 2014
-
- **Ausgabe in der Konsole:**
 - Montag der 3. November 2014

Gliedere in Funktionen

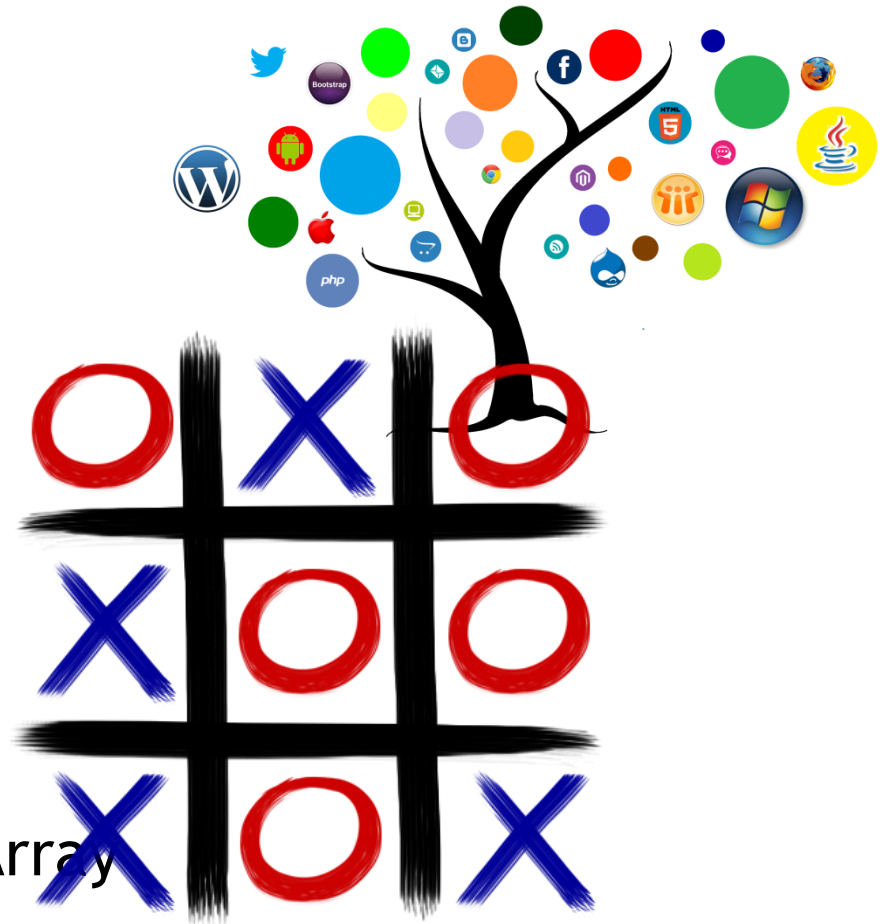
- Erstelle eine private Funktion für das Ermitteln des Wochentages
- Erstelle eine private Funktion für das Ermitteln des Monats
- Erstelle eine Funktion, die 4 Werte als Parameter erhält und eine Zeichenkette mit dem Datum Langformat retour bekommt.

Tic Tac Toe

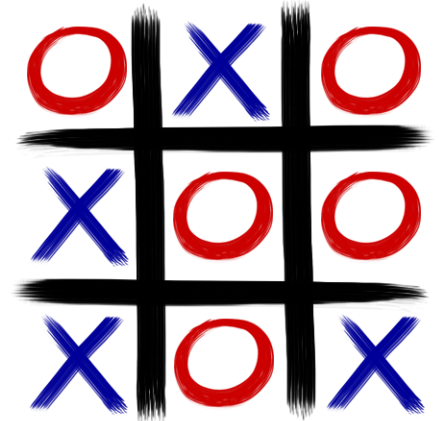
Implementiere Tic Tac Toe

Nutze ein 2 Dimensionales Array

Nutze Methoden



Tic Tac Toe



- Traditionell spielt der erste Spieler mit „X“ und der zweite Spieler mit „O“.
- Die Spieler tätigen abwechselnd ihre Spielzüge.
- Wenn ein Feld bereits mit einem „X“ oder „O“ besetzt ist, kann dieses nicht mehr überschrieben oder gelöscht werden.
- Es gibt nur zwei Spielzustände - Spielzug setzen oder Spiel gewonnen/fertig
- Der Spieler der als erster horizontal, vertikal oder diagonal seine Elemente gesetzt hat gewinnt.

Tic Tac Toe

- Überlege Funktionen die ausgegliedert werden könnten:
- zeichnet ein Spielfeld
 - `static void Board();`
- überprüft ob gewonnen wurde
 - `static int CheckWin()`

Programmcode: Variablen

```
class Program
{
    //making array and
    //by default I am providing 0-9 where no use of zero

    static char[] arr = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };
    static int player = 1; //By default player 1 is set
    static int choice; //This holds the choice at which position user want to mark

    // The flag variable checks who has won if it's value is 1 then some one has won
    // the match if -1 then Match has Draw if 0 then match is still running
    static int flag = 0;

    static void Main(string[] args)
    {
```


Ausgabe des Spielfelds:

```
// Board method which creates board
private static void Board()
{
    Console.WriteLine("    |    |    ");
    Console.WriteLine(" {0} | {1} | {2}", arr[1], arr[2], arr[3]);
    Console.WriteLine("_____|_____|____");
    Console.WriteLine("    |    |    ");
    Console.WriteLine(" {0} | {1} | {2}", arr[4], arr[5], arr[6]);
    Console.WriteLine("_____|_____|____");
    Console.WriteLine("    |    |    ");
    Console.WriteLine(" {0} | {1} | {2}", arr[7], arr[8], arr[9]);
    Console.WriteLine("    |    |    ");
}
```

Teilbereiche von: checkWin()

```
#region Horizontal Winning Condition
//Winning Condition For First Row
if (arr[1] == arr[2] && arr[2] == arr[3])
{
    return 1;
}
//Winning Condition For Second Row
else if (arr[4] == arr[5] && arr[5] == arr[6])
{
    return 1;
}
//Winning Condition For Third Row
else if (arr[6] == arr[7] && arr[7] == arr[8])
{
    return 1;
}
#endregion
```

Teilbereiche von: checkWin()

```
#region vertical Winning Condition

//Winning Condition For First Column
else if (arr[1] == arr[4] && arr[4] == arr[7])
{
    return 1;
}

//Winning Condition For Second Column
else if (arr[2] == arr[5] && arr[5] == arr[8])
{
    return 1;
}

//Winning Condition For Third Column
else if (arr[3] == arr[6] && arr[6] == arr[9])
{
    return 1;
}
#endregion
```

Teilbereiche von: checkWin()

```
#region Diagonal Winning Condition

else if (arr[1] == arr[5] && arr[5] == arr[9])
{
    return 1;
}
else if (arr[3] == arr[5] && arr[5] == arr[7])
{
    return 1;
}
#endregion
```

Teilbereiche von: checkWin()

```
#region Checking For Draw
// If all the cells or values filled
// with X or O then any player has won the match
else if (arr[1] != '1' && arr[2] != '2' && arr[3] != '3'
        && arr[4] != '4' && arr[5] != '5' && arr[6] != '6'
        && arr[7] != '7' && arr[8] != '8' && arr[9] != '9')
{
    return -1;
}
#endregion
```



Array-Beispiel Chessboard

Implementiere Schach

Nutze ein 2 Dimensionales Array

Nutze Methoden



Erstelle ein Schachspiel

- Teil 1:
 - Erstelle ein Array mit 8x8 und positioniere die Schachfiguren. Gib die Schachfiguren in der Konsole aus.
- Teil 2:
 - Erstelle eine Ausgabe mit Zeile 1-8 und Reihe A-H.
 - Lese vom Benutzer Eingaben für das Versetzen von Figuren ein
- Teil 3:
 - Verschiebe die Schachfiguren (ohne Fehlerüberprüfung auf gültige Spielzüge)
 - Baue eine mögliche Rückgängig-Funktionalität ein

Teil 1:

- Erstelle ein neues Projekt
- Lege eine Klasse Array_Chessboard an
- Erstelle eine Main mit:
 - `String[,] chessboard;`
- Befüllt das Array mit leeren Werten „-“
 - `FillChessBoard(out chessboard);`
- Initialisiert das Board mit Spielfiguren
 - `InitializeFigures(chessboard);`
- Gibt das Array in der Console aus
 - `PrintChessBoard(chessboard);`

Schachbrett mit leeren Werten füllen

```
public static void FillChessBoard(out String[,] chessboard)
{
    int size = 8;
    chessboard = new String[size, size];
    for (int row = 0; row < chessboard.GetLength(0); row++)
    {
        for (int col = 0; col < chessboard.GetLength(1); col++)
        {
            chessboard[row, col] = "-";
        }
    }
}
```

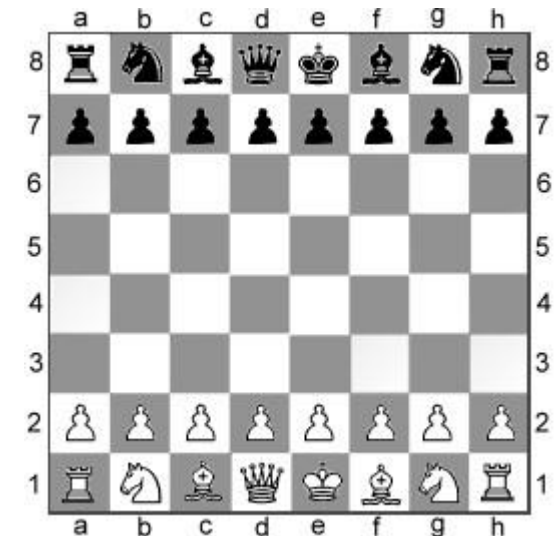
Schachbrett ausgeben mit Spaltenbeschriftung

```
public static void PrintChessBoard(String[,] chessboard, String[] alphabet)
{
    Console.Write("Schachbrett: \n\t");
    foreach (String col in alphabet)
        Console.Write("{0}\t", col);
    Console.WriteLine();

    for (int row = 0; row < chessboard.GetLength(0); row++)
    {
        Console.Write("{0}\t", row + 1);
        for (int col = 0; col < chessboard.GetLength(1); col++)
        {
            Console.Write("{0}\t", chessboard[row, col]);
        }
        Console.WriteLine();
    }
    Console.WriteLine("\n");
}
```

Initialisiere das Array mit Figuren

- K -> König
- D -> Dame
- B -> Bauer
- T -> Turm
- S -> Springer
- L -> Läufer



- Ausgabe des befüllten Schachbretts in der Console

```

1      A      B      C      D      E      F      G      H
2      P1_T    P1_S    P1_L    P1_D    P1_K    P1_L    P1_S    P1_T
3      P1_B    P1_B    P1_B    P1_B    P1_B    P1_B    P1_B    P1_B
4      -      -      -      -      -      -      -      -
5      -      -      -      -      -      -      -      -
6      -      -      -      -      -      -      -      -
7      P2_B    P2_B    P2_B    P2_B    P2_B    P2_B    P2_B    P2_B
8      P2_T    P2_S    P2_L    P2_D    P2_K    P2_L    P2_S    P2_T
Drücken Sie eine beliebige Taste

```

Zugriff auf die Elemente

- Lese vom Benutzer zwei Positionen ein: Zeile und Spalte ein für die Figur mit Start und Zielposition.
- Verschiebe auf das gewünschte Element.
- Erstelle eine Rückgängig Funktion.

8	a8	b8	c8	d8	e8	f8	g8	h8
7	a7	b7	c7	d7	e7	f7	g7	h7
6	a6	b6	c6	d6	e6	f6	g6	h6
5	a5	b5	c5	d5	e5	f5	g5	h5
4	a4	b4	c4	d4	e4	f4	g4	h4
3	a3	b3	c3	d3	e3	f3	g3	h3
2	a2	b2	c2	d2	e2	f2	g2	h2
1	a1	b1	c1	d1	e1	f1	g1	h1
	a	b	c	d	e	f	g	h

Verschiebe ein Element

```
public static String MoveFigure(String[,] chessboard,  
    int fromRow, int fromCol, int toRow, int toCol)  
{  
    String old = chessboard[toRow, toCol];  
    chessboard[toRow, toCol] = chessboard[fromRow, fromCol];  
    chessboard[fromRow, fromCol] = "-";  
    return old;  
}
```

```
public static void UndoMove(String old, String[,] chessboard,  
    int fromRow, int fromCol, int toRow, int toCol)  
{  
    chessboard[fromRow, fromCol] = chessboard[toRow, toCol];  
    chessboard[toRow, toCol] = old;  
}
```

Einlesen vom Benutzer

```
public static void GetUserInfos(out int fromRow, out int fromCol, out int toRow, out int toCol, String[] alphabet)
{
    String input;
```

```
    //Zeile und Spalte einlesen - Startposition
```

```
    Console.WriteLine("\nVerschiebe Spielfigur - Startposition:");
    Console.WriteLine("Bitte geben Sie die Zeile ein:");
    input = Console.ReadLine();
    fromRow = Convert.ToInt32(input) - 1; //Arrays zählen ab 0 -> Zeileneingabe -1
    Console.WriteLine("Bitte geben Sie die Spalte ein:");
    input = Console.ReadLine();
    input = input.ToUpper();
    fromCol = Array.IndexOf(alphabet, input);
    Console.WriteLine("Spalte {0} Interne Nummer {1}", input, fromCol);
    Console.WriteLine("Verschiebe Spielfigur - Startposition: {0}{1}.", input, fromRow+1);
```

```
    //Zeile und Spalte einlesen - Startposition
```

```
    Console.WriteLine("\nVerschiebe Spielfigur - Zielposition:");
    Console.WriteLine("Bitte geben Sie die Zeile ein:");
    input = Console.ReadLine();
    toRow = Convert.ToInt32(input) - 1; //Arrays zählen ab 0 -> Zeileneingabe -1
    Console.WriteLine("Bitte geben Sie die Spalte ein:");
    input = Console.ReadLine();
    input = input.ToUpper();
    toCol = Array.IndexOf(alphabet, input);
    Console.WriteLine("Spalte {0} Interne Nummer {1}", input, toCol);
    Console.WriteLine("Verschiebe Spielfigur - Startposition: {0}{1}.", input, toRow+1);
```

```
    //Fehlerüberprüfung:
```

```
    if (fromCol > 8 || fromRow > 8 || toRow > 8 || toCol > 8 || fromCol < 0 || fromRow < 0 || toRow < 0 || toCol < 0)
        ErrorMsg(out fromRow, out fromCol, out toRow, out toCol);
```

```
}
```

```
private static void ErrorMsg(out int fromRow, out int fromCol,
    out int toRow, out int toCol)
{
    Console.WriteLine("Falsche Eingabe!");
    fromRow = 0;
    fromCol = 0;
    toRow = 0;
    toCol = 0;
}
```