**Dokumentation**

Arbeitsauftrag 01

**Aufgabe 1:** Erstellen einer Konsolenanwendung

Rechtsklick auf die Solution → Add → New Project → Console app

**Aufgabe 2:** Konsolenanwendung  
Das folgende Programm akzeptiert 3 Konsolenargumente.

1. Zeilenanzahl
2. Spaltenanzahl
3. Füllungs-Character
   1. Im Falle eines Wortes wird hier der erste Buchstabe genommen.

using System;

namespace DrawCharacter

{

class Program

{

static void Main(string[] args)

{

if (args.Length < 3)

{

ThrowErrorAndExit("Zu wenig Konsolenargumente!");

}

// Input-Parsing

int columns = 0;

int rows = 0;

char cOutput = '\0';

if (!Int32.TryParse(args[1], out columns) ||

!Int32.TryParse(args[0], out rows) ||

args[2].Length != 1)

{

ThrowErrorAndExit("Ungültige Argumentinhalte.");

}

cOutput = args[2][0];

int newWidth = columns \* 2;

int newHeight = rows \* 2;

if (newWidth > Console.LargestWindowWidth ||

newHeight > Console.LargestWindowWidth)

{

ThrowErrorAndExit("Fenstergröße zu groß!");

}

Console.SetWindowSize(newWidth, newHeight);

Console.SetBufferSize(newWidth, newHeight);

// Call SeetWindowSize 2 times to remove Scroll bars

Console.SetWindowSize(newWidth, newHeight);

Console.BackgroundColor = ConsoleColor.Gray;

Console.ForegroundColor = ConsoleColor.Blue;

Console.Clear();

int offsetX = columns / 2;

int offsetY = rows / 2;

for (int y = 0; y < rows; y++)

{

Console.SetCursorPosition(offsetX, offsetY + y);

for (int x = 0; x < columns; x++)

{

Console.Write(cOutput);

}

}

Console.ReadKey(true);

}

/// <summary>

/// Zeigt eine Fehlermeldung an, wartet auf User-Input und beendet danach das Programm.

/// </summary>

/// <param name="message">Errormeldung</param>

private static void ThrowErrorAndExit(string message)

{

Console.WriteLine("Error: " + message);

Console.ReadKey(true);

Environment.Exit(1);

}

}

}

int offsetX = columns / 2;

int offsetY = rows / 2;

for (int y = 0; y < rows; y++)

{

Console.SetCursorPosition(offsetX, offsetY + y);

for (int x = 0; x < columns; x++)

{

Console.Write(cOutput);

}

}

Console.ReadKey(true);

}

/// <summary>

/// Zeigt eine Fehlermeldung an, wartet auf User-Input und beendet danach das Programm.

/// </summary>

/// <param name="message">Errormeldung</param>

private static void ThrowErrorAndExit(string message)

{

Console.WriteLine("Error: " + message);

Console.ReadKey(true);

Environment.Exit(1);

}

}

}

**Aufgabe 2:** Variablenkontrolle

