Protokoll 03 Kontrollstrukturen u. statische Methoden

Aufgabe:

-Erstellen einer Anwendung welche in der Konsole verschiedenste Formen zeichnen kann.

-Mit besonderem Fokus auf der Objektorientierung und wieder Verwendbarkeit von Funktionen.

Lösungsansatz:

-im main nur ein Funktionsaufruf der „Auswahl“ Funktion

-welche durch einen Switch die verschiedenen  
 Hauptfunktionen aufruft und bei einer 5   
 aussteigt

-und danach mit einem „thread.sleep“ die Form anzeigt um   
 danach die Console zu clearen

-jede Form in 2 Funktionen unterteilt (input, Draw)

-die Input Funktionen kümmert sich um das UI mit   
 cw und cr

-die Draw zeichnet dann nach den Eingabe des Benutzers  
 die jeweilige Form

Source Code:

Main + Abfrage (UI)

static void Main(string[] args)

{

Console.BackgroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.Black;

Console.Clear();

Abfrage();

Console.WriteLine("end");

}

//user input for Menu

static void Abfrage()

{

Console.WriteLine("Auswahl");

Console.WriteLine("[1=vertical, 2= horizontal, 3= Rechteck, 4 = Grid, 5

beenden]");

int ausw = Convert.ToInt32(Console.ReadLine());

if (ausw != 5)

{

switch (ausw)

{

case 1:

//Vertical

ImputVLine();

System.Threading.Thread.Sleep(2000);

Console.Clear();

break;

case 2:

//Horizontal

ImputHLine();

System.Threading.Thread.Sleep(2000);

Console.Clear();

break;

case 3:

//Rect

ImputRect();

System.Threading.Thread.Sleep(2000);

Console.Clear();

break;

case 4:

//grid

ImputGrid();

System.Threading.Thread.Sleep(2000);

Console.Clear();

break;

}

// spielt den selben song noch mal! alles klar den selben song und

los

Abfrage();

}

else

{

//close application if input was 5

return;

}

}

Vertical:

static void ImputVLine()

{

//input

Console.WriteLine("top :");

int top = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("left :");

int left = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("height :");

int height = Convert.ToInt32(Console.ReadLine());

Console.Clear();

//Draw

DrawVLine(top, left, height);

}

static void DrawVLine(int top, int left, int height)

{

//todo: buffer size fail if input grater than ~100

Console.SetCursorPosition(left, top);

for (int i = 0; i < height+1; i++)

{

Console.Write("\*\n");

Console.SetCursorPosition(left, top + i);

}

}

Horizontal:

static void ImputHLine()

{

//input

Console.WriteLine("top :");

int top = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("left :");

int left = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("width :");

int width = Convert.ToInt32(Console.ReadLine());

Console.Clear();

DrawHLine(top, left, width);

}

static void DrawHLine(int top, int left, int width)

{

Console.SetCursorPosition(left, top);

for (int i = 0; i < width; i++)

{

Console.Write("\*");

}

}

Rechteck:

static void ImputRect()

{

//input

Console.WriteLine("top :");

int top = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("left :");

int left = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("height :");

int height = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("width :");

int width = Convert.ToInt32(Console.ReadLine());

Console.Clear();

DrawRect(top, left, height, width);

}

static void DrawRect(int top, int left, int height, int width)

{

//todo: buffer size fail if input grater than ~100

Console.SetCursorPosition(top, left);

DrawHLine(top, left, width);

//hights -2 cus 2 height from vertical

DrawVLine(top+1, left, height-2);

DrawVLine(top+1, left+width-1, height-2);

DrawHLine(top+height-1, left, width);

}

Tabelle:

static void ImputGrid()

{

//input

Console.WriteLine("top :");

int top = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("left :");

int left = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("cellsize :");

int cellsize = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("nOfCols :");

int nOfCols = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("nOfRows :");

int nOfRows = Convert.ToInt32(Console.ReadLine());

Console.Clear();

DrawGrid(top, left, cellsize, nOfCols, nOfRows);

}

static void DrawGrid (int top, int left, int cellsize, int nOfCols, int

nOfRows)

{

for (int i = 0; i < nOfRows; i++)

{

for (int j = 0; j < nOfCols; j++)

{

DrawRect(top + (i \* cellsize) - (i\*1), left + (j \*

cellsize) -

(j\*1), cellsize, cellsize);

}

}

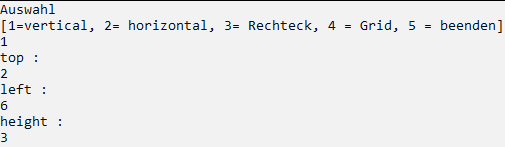
}

Edge Cases

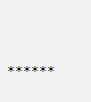
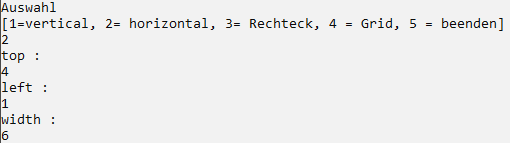
-es bestehen einige Exceptions bei den eingaben welche weder geprüft noch richtig verarbeitet werden (zu hoher imput bei den top/left, falsche typen, …)

Tests:

Vertikal:



Horizontal:



Rechteck:

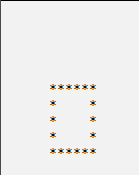
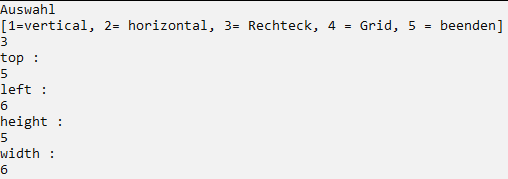


Tabelle:

