

Übung 1

Arbeitsaufwand: etwa 2 Stunden

Teil 1 – Mit der Schildkröte einen Kreis zeichnen

Ziel der Übung war es, mithilfe der „Turtle“ und Pseudoquelltext einen Kreis zu zeichnen.

Der Turtle wird mit angegeben, wie viele Segmente k , welchen Radius r und welchen Mittelpunkt x, y dieser Kreis aufweist.

Lösungsidee:

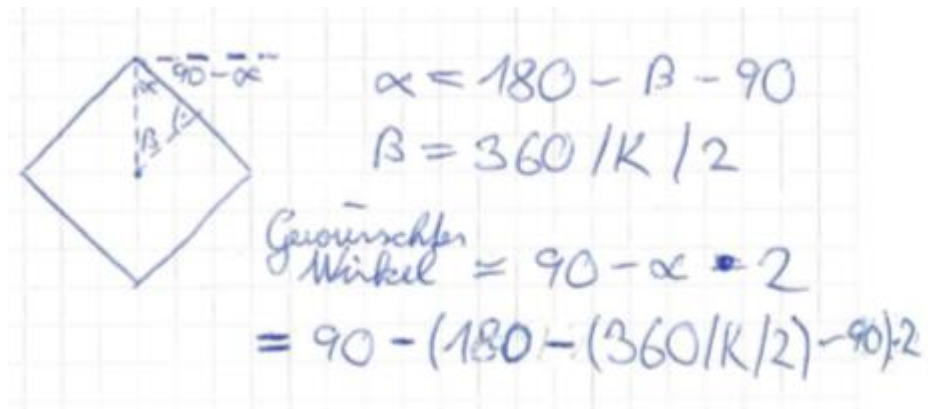
Wichtig zur Lösung dieses Problems ist die Länge der Segmente und der zu drehende Winkel. Diese Länge kann mithilfe der Trigonometrie im rechtwinkligen Dreieck gelöst werden.

Der Kreis besteht nämlich aus k gleichschenkligen Dreiecken, welche wiederum in zwei rechtwinklige Dreiecke aufgeteilt werden können.

Somit errechnet sich die Länge „ l “ wie folgt:

$$l = \sin((360^\circ / k) / 2) * r * 2$$

Der Winkel kann wie folgt berechnet werden:



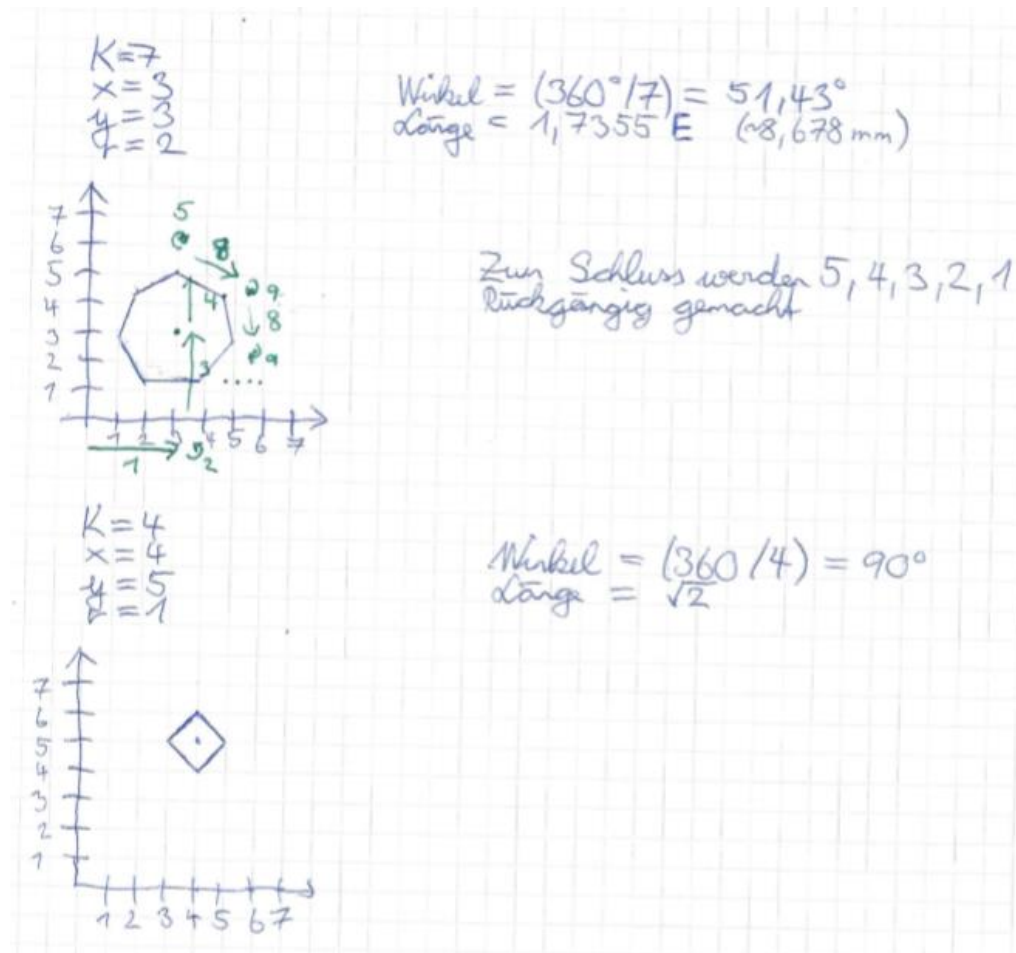
$$\text{Drehwinkel} = 90^\circ - (180^\circ - (360^\circ / k / 2) - 90^\circ) * 2 = (360^\circ / k)$$

Somit muss die Turtle sich nur zum Mittelpunkt begeben, den Radius weitergehen, sich um $90 +$ halber Drehwinkel drehen und dann 4 mal das Zeichnen der Länge l und das Drehen um den Drehwinkel wiederholen. Zum Schluss begibt sich die Turtle wieder zurück zum Anfang.

Lösung

1. gehe x
2. drehe 90
3. gehe y
4. gehe r
5. drehe $-(90 + (360 / k) / 2)$
6. absenken
7. for i from 1 to k by 1 do
8. gehe $\sin((360 / k) / 2) * r * 2$
9. drehe $-(360/k)$
10. end for
11. anheben
12. drehe $(360/k)/2 - 90$
13. gehe r + y
14. drehe -90
15. gehe x
16. drehe 180

Testfälle



Teil 2 – Einen Algorithmus analysieren

Ziel dieser Übung war die Analyse des „Mengenüberdeckungsproblems“ und dem damit vorgegebenen „optimalen“ Algorithmus. Dessen Korrektheit bzw. Inkorrektheit soll bewiesen werden.

Lösungsidee:

„Was passiert, wenn sich viele Elemente der größten Menge mit denen der kleinen Menge überlappen, die kleinen Mengen sich aber gegenseitig NICHT überlappen?“

Lösung

Nehmen wir als Beispiel folgende Werte:

$$U = \{1, 2, 3, 4, 5, 6\}$$

$$S = \{S1, S2, S3, S4\}$$

$$S1 = \{1, 2, 3\}$$

$$S2 = \{1, 4\}$$

$$S3 = \{2, 5\}$$

$$S4 = \{3, 6\}$$

Würden wir dieses Beispiel mithilfe des vorgegebenen Algorithmus lösen so entstünde folgendes:

1. S1 auswählen
2. $S = \{S2, S3, S4\}$, $U = \{4, 5, 6\}$
3. S2 auswählen
4. $S = \{S3, S4\}$, $U = \{5, 6\}$
5. S3 auswählen
6. $S = \{S4\}$, $U = \{6\}$
7. S4 auswählen
8. $S = \{\}$, $U = \{\}$

In diesem Fall besteht die Lösungsmenge aus $\{S1, S2, S3, S4\}$. Somit wurden 4 Mengen S benötigt.

Betrachtet man jedoch die Werte, so kann leicht erkannt werden, dass die Menge U auch nur durch S2, S3 und S4 abgebildet werden, was diesen Algorithmus widerlegt:

1. S2 auswählen
2. $S = \{S1, S3, S4\}$, $U = \{2, 3, 5, 6\}$
3. S3 auswählen
4. $S = \{S1, S4\}$, $U = \{5, 6\}$
5. S4 auswählen
6. $S = \{S1\}$, $U = \{\}$

Teil 3 – Hello World

Ziel dieser Übung war es, mithilfe von C++ und der IDE Visualstudio in einem Konsolenfenster „Hello World!“ auszugeben.

Lösungsideen:

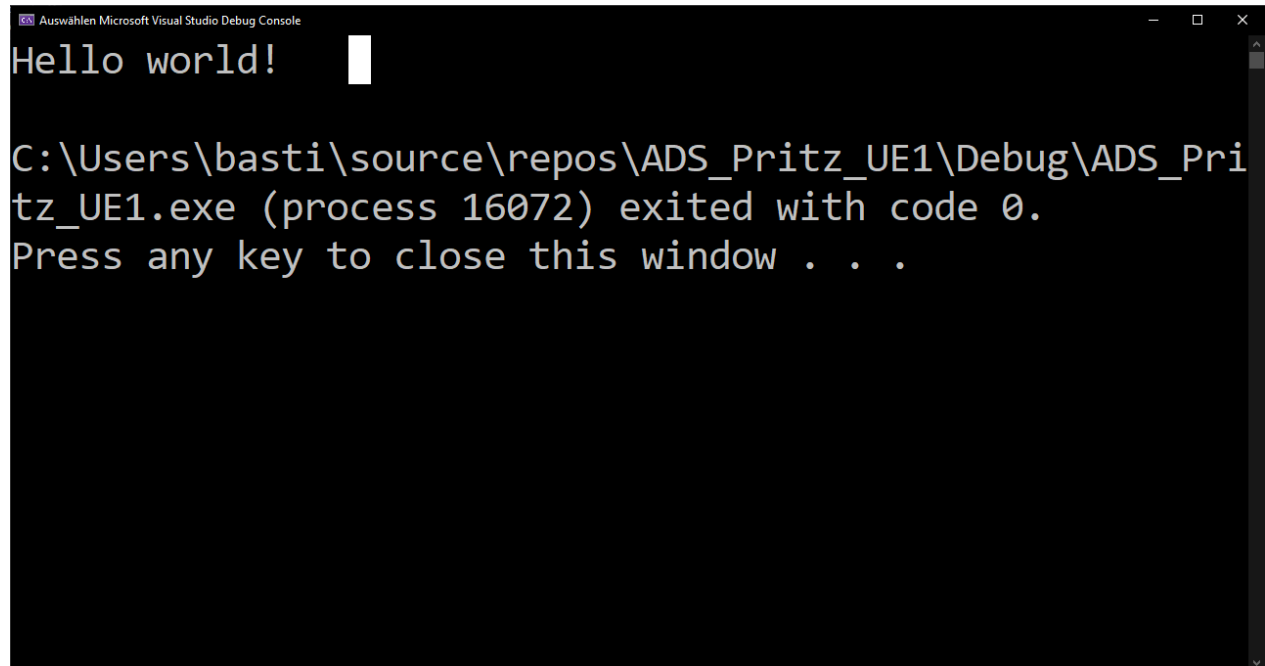
Um etwas auf die Konsole zu schreiben, wird in C++ der Befehl „std::cout“ verwendet.

Der leichteste Weg wäre also die ganze Zeichenkette „Hello world“ mit dem Befehl „std::cout“ auszugeben.

Alternativ könnte der String aber auch in seine Einzelteile „chars“ (String = Char Array) aufgeteilt werden und jedes Zeichen einzeln über eine Schleife ausgegeben mit „std::cout“ werden.

Aus Gründen der Simplizität habe ich mich für die Ausgabe des ganzen Strings entschieden.

Lösung:

A screenshot of a Visual Studio Debug Console window. The title bar at the top reads "Auswählen Microsoft Visual Studio Debug Console". The console output shows "Hello world!" followed by a cursor. Below that, it says "C:\Users\basti\source\repos\ADS_Pritz_UE1\Debug\ADS_Pritz_UE1.exe (process 16072) exited with code 0." and "Press any key to close this window . . .".

```
Auswählen Microsoft Visual Studio Debug Console
Hello world!
C:\Users\basti\source\repos\ADS_Pritz_UE1\Debug\ADS_Pritz_UE1.exe (process 16072) exited with code 0.
Press any key to close this window . . .
```