C# SFML – Klassen & Objekte

C# ist eine Objekt-Orientierte Programmiersprache. Deswegen ist der Code eines C# Programms immer in Klassen angeordnet. Bisher haben wir nur in der MainScreen Klasse gearbeitet. Um aber kompliziertere und größere Programme zu schreiben, sollten wir anfangen, Objekt-Orientiert zu arbeiten.

Im Projektmappen-Explorer kann man mit

Rechtsklick auf einen Ordner -> Hinzufügen -> Neues Element -> Klasse

eine neue C# Klassendatei in diesem Ordner erzeugen.

Jede neue Klasse sollte folgenden Standartaufbau haben:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace NameSpaceName

{

class KlassenName

{

}

}

Hinter "class" steht der Klassenname.

Nach "namespace" kommt immer der NameSpace, dieser ist quasi der Unterordner in der sich unsere Datei befindet. Wenn eine Datei in andere Ordner verschoben, sollte man den NameSpace anpassen.

Die "using" Anweisungen geben an, auf welche NameSpaces die Klasse zugreifen kann. Will man auf Klassen zugreifen, die nicht den selben NameSpace haben, muss man sie hier angeben.(Blöderweise ist die Anweisung "using System.Linq;" standartmäßig drin, erzeugt aber einen Error, weswegen man sie entfernen muss.)

Aus **Klassen** werden **Objekte** erzeugt, mit denen man arbeiten kann. Wir kennen schon verschiedene Klassen.

RectangleShape ist zum Beispiel eine Klasse.

Immer wenn wir "new RectangleShape()" benutzen, erzeugen wir ein **Objekt** aus dieser **Klasse.**

Jedes Objekt besteht hauptsächlich aus 2 Dingen. Seinen **Feldern** und **Methoden**.

**Felder** kann man auch als "Klassenvariablen" sehen. Es sind Variablen die in dem gesamten Objekt benutzt werden können, anders als die normalen Variablen in Methoden. **Felder** erkennt man daran, dass die **außerhalb von Methoden in ihrer Klasse deklariert werden.** Hier ein kleines Beispiel für Felder.

Klassendeklaration, ab hier beginnt die Klasse!

class Beispiel

{

Felder/Klassenvariablen

public int nummer = 3;

private string name;

double andereNummer;

public void tuMalWas()

Methode/Funktion

{

int i;

i = 5;

string s = "wort";

}

Gewöhnliche Variable in einer Methode

}

Klassen/Objekte können außerdem eine besondere Methode enthalten. Den **Konstruktor.** Den Konstruktor erkennt man daran, dass er **genauso heißt wie die Klasse**. Außerdem hat er **keinen Rückgabetypen**(kein "void")

class Beispiel

Konstruktor

{

public Beispiel()

{

}

public void IrgendeineMethode()

{

Normale Methode

}

}

Der Konstruktor ist besonders, weil er nicht wie eine normale Methode ausgeführt werden kann, sondern immer nach Erstellung des Objekts ausgeführt wird. Er ist also die erste Methode die das Objekt ausführt.

Man benutzt den Konstruktor üblicherweise um alle Startwerte zu setzen. Damit ist er Vergleichbar zur "Setup()" Methode die wir schon aus der MainScreen Klasse kennen.