C# und MySQL /MariaDB

Als erstes benötigt man die MySql.Data.dll, die man als Verweis hinzufügen muss.
Dazu öffnen wir die Packet-Manager-Konsole durch den Tastenkürzel **ALT + T + N + O** und geben dort ein:
[crayon-677fa36d12736923374908/]
und bestätigen das Konsolenfenster mit Enter.

2. Namespace hinzufügen:
[crayon-677fa36d12741562087874/]
3. StringBuilder erzeugen. Alternativ geht natürlich ein gewöhnlicher String:
[crayon-677fa36d12743858124955/]
4. Connection String erzeugen. Port kann evtl. ein anderer sein:
[crayon-677fa36d12745247722619/]

> *In der Regel umschließt man Datenbankverbindungen in einem try/catch. Dies wird hier von meiner Seite aber aus Gründen der besseren Übersicht nicht getan. Solltet Ihr das so verwenden wollen, empfiehlt es sich definitiv einen try/catch Block um rum zu verwenden*

Methode um ein SQL Befehl auszuführen
[crayon-677fa36d12747594067796/]
Beispiel:
[crayon-677fa36d12748916027501/]
Methode um einen Befehl aus der Datenbank zu erhalten
[crayon-677fa36d1274a161370706/]

Beispiel:
[crayon-677fa36d1274b239377007/]

Tastenkürzel

Strg + K, Strg + C = Zeile auskommentieren

Strg + K, Strg + U = auskommentierte Zeile wieder aktivieren

Strg + K, Strg + D = Zeilen wieder richtig ausrichten

Strg + M + L = kompletten Codeabschnitt reduzieren

Strg + Umschalt. + L = Zeile löschen

Strg + X = Zeile Ausschneiden

Revidiert am 2015-05-17 (Billige Tastenkürzel raus und wirklich interessante rein)

Strg + Alt + B = Breakpointfenster mit allen verfügbaren Breakpoints anzeigen

---

Polymorphie

Unter Polymorphie versteht man, wenn eine Klasse von der anderen erbt.

Dabei nimmt eine Klasse die Rolle der Elternklasse und die anderen, die einer Kindklasse an.

Bedingung:

1. Eigenschaft/Methode muss gleich heißen
2. Die Kindklasse erbt die Elternklasse (LKW : PKW)
3. die Methode der Eltern muss virual sein
4. die Methode des Kindes muss override sein (Das heißt die Elternmethode wird überschrieben/ ergänzt)

das base.[Methodenname] implementiert die Methode aus der Elternklasse in die Kindklasse
[crayon-677fa36d12eed817403110/]
erstellen wir nun ein Objekt von PKW und lassen die Methode aufrufen, bekommen wir als Ausgabe
[crayon-677fa36d12ef3934445017/]
tun wir dasselbe mit der LKW Klasse, bekommen wir als Ausgabe:
[crayon-677fa36d12ef4454033749/]

sealed – Klasse versiegeln

Generell kann jede Klasse von einer anderen Erben. Möchte man aber vermeiden, dass von einer Klasse geerbt werden soll, nutzt man den Ausdruck sealed. Dies ist Sinnvoll, wenn man weiß, dass man in der Vererbung in der letzten Instanz angekommen ist.
[crayon-677fa36d12ef6920653442/]

abstract – Abstrakte Klassen

Abstrakte Klassen sind Klassen, die reine vererbbare Klassen sind. Das bedeutet man kann aus der Klasse kein Objekt mehr erzeugen.
[crayon-677fa36d12ef8673111330/]

Eigene Mini Programmiersprache schaffen mithilfe von Regulären Ausdrücken

Filtert nur die Ausdrücke aus, die mit[beginnen und mit] enden.

Match liefert nur den erstgefundenen Wert,

MatchCollection dagegen alle gefundene.
[crayon-677fa36d13116847310013/]
Nun könnte man über ein switch/case eine Abfrage erstellen.

Z.B. case FELD1:

tue dies oder jenes

Quellen:

http://www.regexr.com/ – Online Editor

http://www.mycsharp.de/wbb2/thread.php?threadid=41009 – Tutorial

http://www.dotnetperls.com/regex-match – Tutorial

Binding Elementbinding

Möchte man eine Elementeigenschaft an die andere binden, muss man in die Quelleigenschaft gehen und ein {Binding … } einsetzen.

Zuerst wird das Objekt, dann dessen Eigenschaft abgefragt.

ElementName (welches Element?)

Path(welche Eigenschaft?)

Bei Path könnte auch etwas anderes stehen. SelectedIndex z.B.

oder Items[0] für das erste Item in der ListBox
[crayon-677fa36d132c8665031175/]
Das Visual Studio bietet mit dem Intellisense eine große Hilfe bei der Selektion.