# Arbeitsblatt 5: MVC-IET-gibb kennenlernen

## Ausgangslage

Wir werden in drei Wochen mit dem Bilderdatenbank-Projekt beginnen. Diese Applikation wollen wir mit dem MVC-Muster erstellen. Dazu haben Sie drei Möglichkeiten:

1. Sie verwenden das MVC-IET-gibb, das Ihnen zur Verfügung gestellt wird. Es handelt sich dabei um eine prozedurale Eigenentwicklung.  
   **Für den Grossteil der Lernenden wird empfohlen, diese Vorgabe zu verwenden!**Denn das ganze Modul baut exemplarisch darauf auf, die Arbeitsblätter sind darauf ausgerichtet und Sie kriegen die beste Unterstützung durch die Lehrperson.
2. Sie verwenden das bbc-MVC, das Ihnen ebenfalls zur Verfügung gestellt wird. Dieses Framework ist objektorientiert und empfiehlt sich für Lernende, die entweder bereits damit gearbeitet haben oder die lieber nach dem objektorientierten Programmierparadigma entwickeln möchten. Auch hier kriegen Sie Unterstützung durch die Lehrperson, mit einer Einführung in das Framework und mit Workshops.
3. Sie erstellen eine eigene MVC-Struktur, wobei Sie auch eine **selber erstellte Struktur** von einer früheren Web-Applikation verwenden dürfen. Diese Variante empfiehlt sich, wenn Sie bereits eine Web-MVC-Struktur entwickelt haben oder über Erfahrung in der Web-Entwicklung verfügen. Es spielt dabei keine Rolle, ob sie den objektorientierten oder prozeduralen Ansatz verwenden.

Alle anderen Frameworks (wie Laravel, Symfony, CodeIgniter, AngularJS, usw.) dürfen im Projekt nicht verwendet werden. In diesem Arbeitsblatt wird das MVC-IET-gibb vorgestellt.

## Ziele

* Sie verstehen das Muster zur Strukturierung von Software *Model View Controller* *MVC*:  
  Model = Datenmodell, View = Präsentation, Controller = Programmsteuerung.
* Sie analysieren und dokumentieren die einzelnen Komponenten des MVC-IET-gibb Frameworks.
* Sie verstehen das Framework so weit, dass Funktionen zur Beispielapplikation *Kontakteverwaltung* hinzugefügt werden können. Zu einem späteren Zeitpunkt werden Sie eine Anwendung mit diesem Framework erstellen.

## Unterlagen / Quellen

* Dokument *AB151MVC-IET-gibb\_Basics.pdf* enthält eine Einführung in das Framework.
* <http://www.php.net/> ist die erste Quelle für Dokumentation und Hilfe zu PHP.
* Im Modulordner finden Sie unter *09\_Diverses* die gesamte PHP-Dokumentation von *php.net* als Hilfedatei.

## Vorgehen

* Falls Sie nicht bereits wissen, welches Framework Sie verwenden wollen, installieren Sie das MVC-IET-gibb und das bbc-MVC und schauen beide näher an.
* Entscheiden Sie sich, womit Sie Ihre Applikation entwickeln wollen.
* Falls Sie das MVC-IET-gibb wählen, fahren Sie mit diesem Arbeitsblatt weiter. Falls Sie sich für das bbc-MVC entscheiden, dann finden Sie Informationen dazu im Arbeitsblatt 5a. Falls Sie sich für eine Eigenentwicklung entscheiden, dann haben Sie drei Wochen Zeit, um sich die Struktur zu erarbeiten, bevor wir mit dem Projekt beginnen.
* Installieren Sie die Beispielapplikation *Kontakteverwaltung* gemäss Anleitung unten.
* Untersuchen Sie die Struktur der Anwendung und durchleuchten Sie die einzelnen Komponenten und Dateien.
* Beantworten Sie die Fragen und schreiben Sie die Antworten direkt in dieses Dokument.

## Zeit

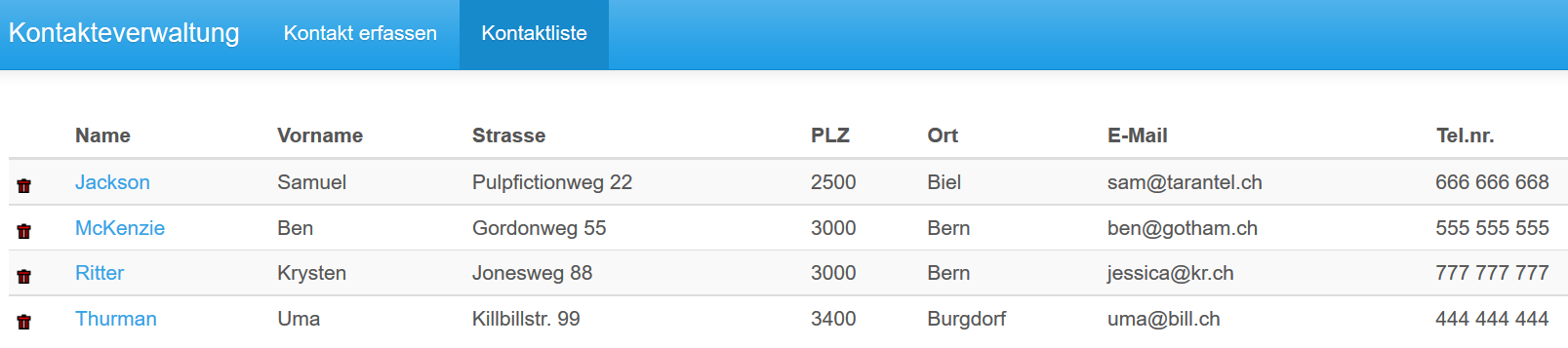
2-3 Lektionen

## Auftrag

#### Aufgabe 1

Installieren Sie die Beispielanwendung *Kontakteverwaltung* und probieren Sie diese aus:

1. Entpacken Sie das Archiv *kontakte.zip* im Wurzelverzeichnis oder einem Unterverzeichnis des Webservers (*htdocs*/), z.B. in den Ordner *m151* (was in der Folge angenommen wird).
2. Starten Sie phpMyAdmin <http://localhost/phpmyadmin/> .
3. Aus dem vorherigen Auftrag *AB151-03MariaDB-Crashkurs* sollte die Datenbank *kontakte* vorhanden sein, mit den beiden Tabellen *kontakte* und *orte* und einigen Datensätzen pro Tabelle.  
   Falls dies nicht der Fall ist, können Sie die DB mit dem Skript *kontakte.sql* erstellen. Diese Datei finden Sie im Verzeichnis *sql* in der Verzeichnisstruktur, die Sie unter Punkt 1 erstellt haben. Importieren Sie die Datei in *phpMyAdmin* oder führen Sie den gesamten Inhalt (via Copy&Paste) im SQL-Fenster von *phpMyAdmin* aus.
4. Starten Sie die Anwendung: <http://localhost/m151/kontakte/php/index.php> . Es werden alle erfassten Kontakte angezeigt:



1. Wählen Sie *Kontakte erfassen* über das Menü und erfassen Sie mehrere Kontakte.

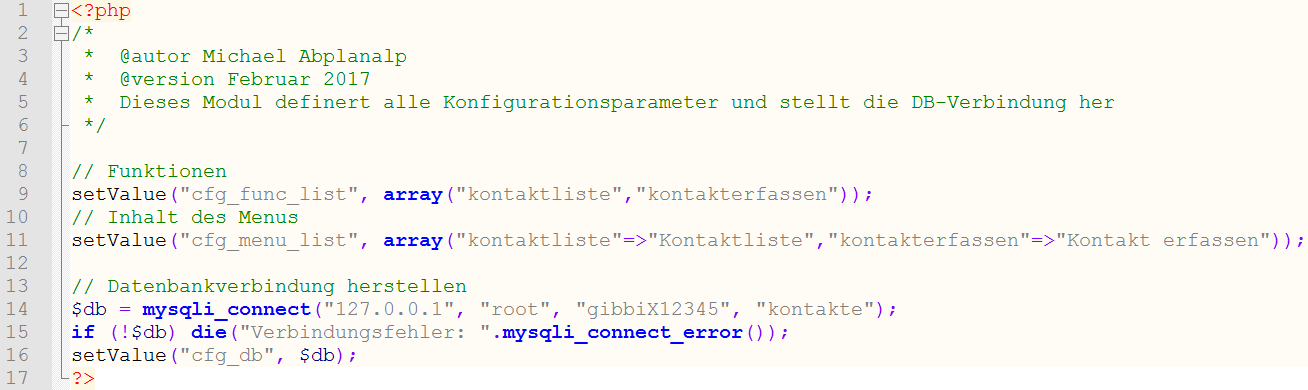


1. Unter dem Menüpunkt *Kontaktliste* sehen Sie alle Kontakte. Der Nachname ist ein Link, über den Sie die Kontakte bearbeiten könne. Mit dem Papierkorb-Icon löschen Sie die gewünschten Einträge (siehe Tabelle oben).

#### Aufgabe 2

Das Framework ist im Dokument *AB151MVC-IET-gibb\_Basics.pdf* beschrieben. Schauen Sie sich die Komponenten und Dateien genauer an und beantworten Sie die folgenden Fragen.

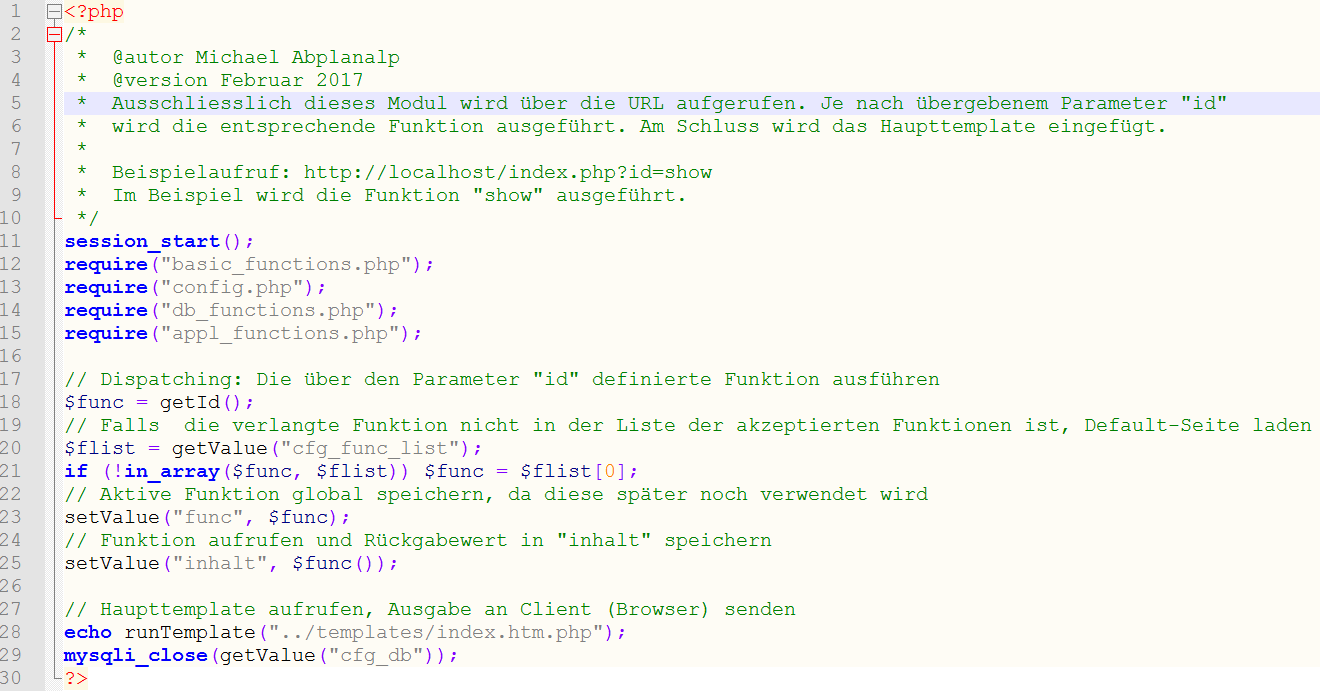
**config.php**



1. Wie und auf welcher Zeile wird die Datenbankverbindung hergestellt? Notieren Sie die Bedeutung der einzelnen Parameter!

1. Was passiert in Zeile 16 mit *$db*? Warum wird das gemacht?

**index.php**



1. Welche Dateien werden eingebunden (Zeilen 12-15)? Was ist die Aufgabe dieser Dateien?

1. Was macht die Funktion *getId()* und was ist der Rückgabewert?

1. Was genau passiert in Zeile 25? Offensichtlich wird ein Wert im globalen Array unter dem assoziativen Bezeichner "inhalt" gespeichert. Aber was genau bewirkt *$func()* (schauen Sie dazu das Modul *appl\_functions.php* genauer an!)?

1. Bisher gab es verschiedene Statements und es wurden Funktionen ausgeführt. Jedoch erfolgte noch keine HTML-Ausgabe an den Client. Wie erfolgt diese Ausgabe? (Siehe auch nächste Frage.)

**index.htm.php**

Dies ist das Haupttemplate, das das grundsätzliche Layout festlegt und in verschiedene Bereiche gliedert (Menü-Nvigation, Content-Bereich und Fusszeile).



1. Was wird in den Zeilen 5-8 gemacht und warum?

1. Was macht die Funktion *getMenu()* in Zeile 26? Den Code finden Sie im Modul  
   *basic\_functions.php*.

1. Was genau passiert in Zeile 31?

#### Aufgabe 3

Das Hauptziel von Modul 151 ist es, mit dem MVC-IET-gibb eine eigene Anwendung zu erstellen. In einem ersten Schritt analysieren wir die Beispielapplikation. In einem zweiten Schritt erweitern wird die Kontakteverwaltung mit zusätzlichen Funktionen.

Was braucht es, um eine neue Funktion inkl. Menüeintrag zu erstellen? Beschreiben Sie die Schritte so gut als möglich!

**Hinweis** In mind. drei Modulen müssen Anpassungen vorgenommen werden: *config.php* (2 Ergänzungen), *appl\_functions.php* (1 neue Funktion) und *db\_functions.php* (1 oder mehrere SQL-Statements). Zudem muss ein neues Template *neuefunktion.htm.php* erstellt werden. Evtl. ist es sinvoll, zusätzliche (Unter)funktionen in *appl\_functions.php* und/oder *basic\_functions.php* erstellen.