**Übungsaufgabe Customer Management Interface (Abschlussübung OOP)**

Es ist eine Anwendung zu erstellen, über die Kundenkontakte angelegt und gepflegt werden können.

**Ein Kundendatensatz besteht aus folgenden Bestandteilen:**

* ID
* Vorname (Pflichtangabe)
* Nachname (Pflichtangabe)
* Email-Adresse (Pflichtangabe)
* Geburtsdatum (optional)
* Wohnort (optional)

**Die Funktionalität soll umfassen:**

1. Anzeigen aller in der DB vorhandenen Kunden in Tabellenform
   1. Vor- und Nachname sollen in einer Tabellenspalte zusammengefasst werden
   2. Die Email-Adresse soll als ‚mailto‘-Link dargestellt werden, der eine Betreffzeile übergeben bekommt (‚Nachricht von meineseite.de‘) sowie ein bcc an die Email-Adresse ‚customerservice@meineseite.de‘
   3. Das Geburtsdatum soll im EU-Format dargestellt werden
   4. Die letzte Spalte der Tabelle soll hinter jedem Kunden zwei Links beinhalten: Löschen und Bearbeiten des Kunden (URL-Parametersteuerung)
2. Formular zum Anlegen eines neuen Kunden
   1. Die Pflichtfelder müssen geprüft und dementsprechend auch alle Formularfelder nach dem Absenden vorbelegt sein, falls es einen Fehler gab
   2. Das Geburtsdatum kann wahlwiese als Triple-Select oder als Textfeld realisiert werden
   3. Vor dem Speichern in die DB muss geprüft werden, ob die Email-Adresse des Kunden bereits in der DB vorhanden ist
   4. Das Formular muss Fehlermeldungen ausgeben können
   5. Nach Absenden des Formulars muss der Sachbearbeiter entweder eine Erfolgsmeldung oder eine Fehlermeldung erhalten
3. Formular zum Editieren eines bestehenden Kunden
   1. Das Formular soll bei Klick auf den Link ‚bearbeiten‘ aufgerufen werden und das bestehende Formular zum Neuanlegen eines Kunden ersetzen
   2. Aufgrund der DEBUG-Ausgaben soll nach dem Klick auf den ‚bearbeiten‘-Link direkt zum Formular gesprungen werden (Stichwort: HTML-Anker)
   3. Bei Aufruf des Bearbeiten-Formulars muss dieses mit den entsprechenden Kundendaten vorausgefüllt sein
   4. Das Formular muss die gleichen Feld- und DB-Prüfungen vornehmen, wie das zum Anlegen eines neuen Kunden
   5. Das Formular muss Fehlermeldungen ausgeben können
   6. Nach Absenden des Formulars muss der Sachbearbeiter entweder eine Erfolgsmeldung oder eine Fehlermeldung erhalten
   7. Über dem Formular soll sich ein Link befinden, über den der Bearbeitungsmodus verlassen und zum Formular zum Neuanlegen eines Kunden zurückgekehrt werden kann
4. Löschen eines Kunden
   1. Vor dem Löschen eines Kundendatensatzes soll eine Sicherheitsabfrage erscheinen, die entweder bestätigt oder abgebrochen werden kann.
   2. Nach Abschluss des Löschvorgangs soll eine Rückmeldung an den Sachbearbeiter erfolgen

**Hinweise**

* Vorgegeben ist ein Interface, aus dem sowohl das Klassen- als auch das DB-Modell (DB-Name: ‚customer\_management‘) abzuleiten ist. Der Tabellenname lautet wie in der Klasse ‚customer‘ und das Spaltenpräfix soll ‚cus\_‘ lauten.
* Die Setter-Methoden sollten zumindest eine Datentyp-Prüfung sowie einen cleanString()-Aufruf beinhalten.
* Das Geburtsdatum muss validiert werden (keine ungültigen Datumsangaben wie 31.02. erlaubt) und soll in der Tabellenansicht im EU-Format ausgegeben werden.
* Der Setter für das Geburtsdatum sollte das Datum in das ISO-Format umwandeln (eine entsprechende Funktion hierfür findet sich in der Include-Datei ‚dateTime.inc.php‘), da später in der DB nur ISO-Datumsformate gespeichert werden dürfen.
* Die benötigten Klassen sollen automatisch geladen werden
* Der Constructor soll entweder automatisch funktionieren (ebenso das nachträgliche Ändern von Objekt-Attributswerten), oder manuell, wobei auch hier sowohl leere als auch gefüllte Objekte erstellt werden können müssen.
* (Am besten zum Schluss): Umbauen des Projekts gem. MVC-Designpattern
* Alle Dateien sowie die Klassen und die in ihr befindlichen Methoden (auch der Konstruktor) sollen mittels Annotations dokumentiert werden. Bei Verwendung des autoConstructors mit Trait ist in der Klasse ein entsprechender Hinweis mittels Annotation anzubringen.

Zur besseren Orientierung einige Screenshots, die die Arbeitsweise der Kundenverwaltung demonstrieren:

1. Standardansicht: Anzeige der vorhandenen Kunden in einer Tabelle, innerhalb der Tabelle Links für die Email sowie für das Löschen und das Bearbeiten von Datensätzen.

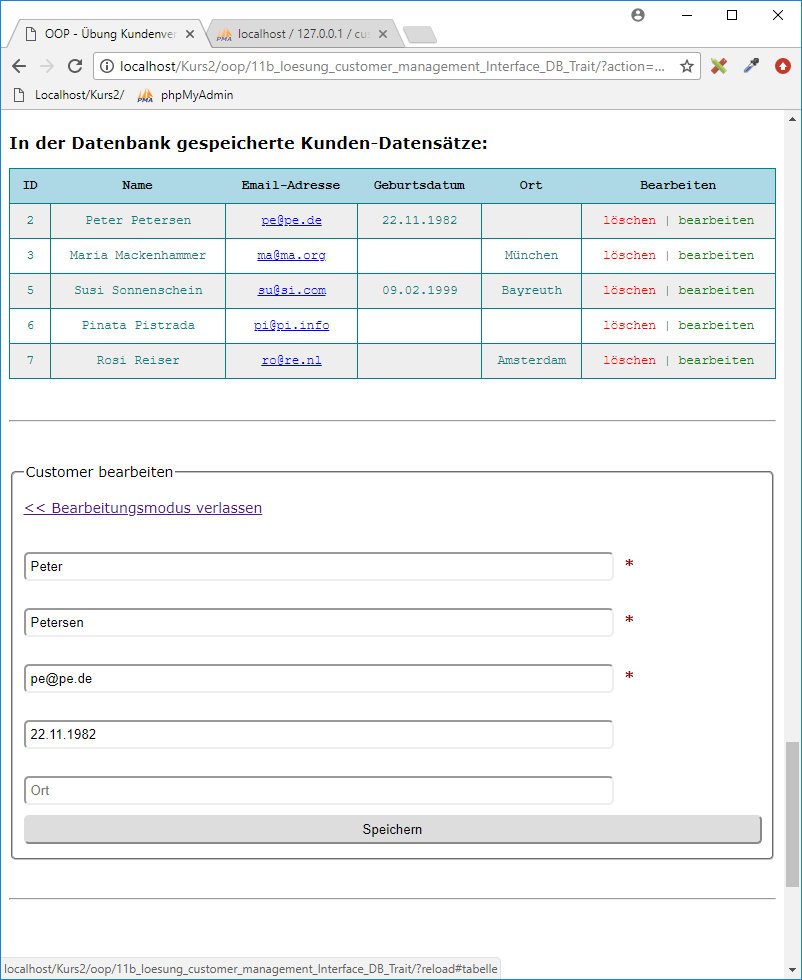
Darunter das Formular zum Anlegen eines neuen Kunden. Die Formularverarbeitung ruft zum Schluss die Customer-Methode ‚saveToDb()‘ auf.



1. Nachdem auf den Link ‚bearbeiten‘ geklickt wurde, soll das Formular gegen eines zum Ändern eines bestehenden Kundendatensatzes ausgetauscht werden.

Dieses Formular ist mit den Kundendaten aus der DB bereits vorbelegt. Hierzu wird in der URL-Parameterverarbeitung für ‚bearbeiten‘ die Customer-Methode ‚fetchFromDb()‘ aufgerufen werden, die die jeweiligen Kundendaten aus der DB ausließt, mit denen das Formular vorbelegt wird. Ein Klick auf ‚Bearbeitungsmodus verlassen‘ ruft die Standardansicht auf und springt via HTML-Anker zurück zum Tabellenanfang.

Die Formularverarbeitung ruft zum Schluss die Customer-Methode ‚updateToDb()‘ auf.



1. Nachdem auf den Link ‚löschen‘ geklickt wurde (URL-Parameterverarbeitung), soll eine Sicherheitsabfrage erscheinen (Design ist egal).

Die URL-Parameterverarbeitung für das Löschen des Datensatzes ruft im Fall der Bestätigung am Ende die Customer-Methode ‚deleteFromDb()‘ auf.

