



## **Cahier de texte d'une matière**

MATIÈRE : SLAM4 - Réalisation et maintenance de composants logiciels

Nom du Professeur : Philippe.Pilla

Annee scolaire : 2015-2016

<b>PÉRIODE</b>	<b>COURS OU TRAVAUX PRATIQUES</b>
<b>01/09/2015</b> <b>8h-12h</b>	Présentation de l'année. Petit exercice de programmation en mode console. Traitement de chaîne de caractères.
<b>01/09/2015</b> <b>8h-12h</b>	Présentation de l'année. Petit exercice de programmation en mode console. Traitement de chaîne de caractères.
<b>07/09/2015</b> <b>13h-17h</b>	Nous commençons aujourd'hui notre suite de travaux pratiques - cours sur le thème gestion d'un hotel. Cette suite de TP commence par la programmation d'une application locale écrite en langage C# dans l'environnement Visual Studio 2013 Ultimate. L'ensemble des classes de cette application seront réutilisées dans une bibliothèque de classes qui servira dans la suite du cours. Cette bibliothèque de classes sera utilisée dans un premier projet WEB développé en ASP.NET/C# puis dans un autres projet WEB ASP.NET mais de type MVC (Modèle-Vue-Contrôleur). Cette réutilisation du code, outre d'être un des principes fondamentaux de la programmation Objet permet de répondre aux exigences de la matière SLAM4 qui veut voir la programmation de composants réutilisables.  L'application locale que nous commençons aujourd'hui va se faire en plusieurs étapes : Un cahier des charges est donné à l'étudiant. ainsi que le script SQL de la base de données. Nous élaborons ensemble le diagramme de classe au tableau de manière à familiariser l'étudiant avec la notation UML (Unified Modeling Language) d'une part et de fournir une architecture applicative solide. Nous commençons par développer aujourd'hui la partie permettant d'obtenir la disponibilité des Chambres de l'hôtel. (Menu - classe de contrôle singleton - classes métier )
<b>08/09/2015</b> <b>8h-12h</b>	Suite du lundi 7 septembre 2015
<b>14/09/2015</b> <b>13h-17h</b>	Étape 2 de l'application locale : Gestion des dates d'occupation des chambres d'hôtel dans une classe d'association (nouvelle notion). Ces objets d'association permettra de joindre par la suite les objets de type Cchambre et de type Cclient afin de gérer les disponibilités et les réservations en faisant appel le moins souvent possible au disque du serveur.
<b>15/09/2015</b> <b>8h-12h</b>	Suite du 14 septembre 2015
<b>21/09/2015</b> <b>13h-17h</b>	Implémentation de la classe Cclient et gestion des réservations des clients. La classe d'affichage affiche dorénavant les trois état LIBRE-OCCUPEE-RESERVEE ainsi que le nom du client qui occupe ou qui réserve la chambre.
<b>22/09/2015</b> <b>8h-12h</b>	Les classes Cclient (métier) et Cclients (contrôle) sont implémentées. Nous modifions la classe d'association pour rajouter la référence vers un objet de type Cclient. Nous enrichissons la classe métier Cclient et la classe d'association CInfoOccupation d'une méthode enregistrer() qui permettra d'assurer la persistance des données client et de réservation/occupation des chambres vers la base 'GESTION HOTEL' présente sur le SGBD SQL SERVER . Nous modifions notre diagramme de classes en conséquence.
<b>28/09/2015</b> <b>13h-17h</b>	Terminé travail du 22/09. Suite du travail sur TP Gestion Hotel : Implémenter la gestion des réservations des chambres - modification du menu de la fenêtre principale - création de deux interfaces graphiques supplémentaires une pour la réservation , une pour la recherche des anciens clients.

<b>29/09/2015</b> <b>8h-12h</b>	Suite travail du 28/09. Introduction à la notion de délégué en C# - Réalisation d'un petit programme permettant de mettre en œuvre cette notion - Les délégués (classe delegate) seront réutilisés ultérieurement dans la gestion des événements en langage C# (classe Event).
<b>05/10/2015</b> <b>13h-17h</b>	Suite travail du 29. Notion d'héritage, de polymorphisme d'héritage et redéfinition de méthode : L'application 'Gestion Hotel' doit distinguer les clients et les clients Entreprise : Nous modifions les classes métier afin de mettre en place un héritage entre la classe Cclient et CclientEise. La collection des clients est maintenant polymorphe puisqu'elle abrite désormais des clients et des clients entreprise. La méthode enregistrer de la classe Cclient est redéfinie dans CclientEise (Virtual/override).
<b>06/10/2015</b> <b>8h-12h</b>	Modification de l'application Gestion Hotel Locale en implémentant un héritage entre client et ClientEise (Client Entreprise)
<b>12/10/2015</b> <b>13h-17h</b>	Fini travail du 6/10 + Nous commençons l'étude du langage ASP.NET/C# : Je propose à l'étudiant de réaliser une suite d'exercices simples permettant de l'initier à la programmation WEB en environnement ASP.NET sous Visual Studio 2013 et de comprendre les principaux mécanismes de ce type de programmation. Etude de l'architecture d'une page ASP.NET, Les événements d'une page ASP.NET (Load et Init), Mécanisme du ViewState.
<b>13/10/2015</b> <b>8h-12h</b>	Suite de l'initiation ASP.NET : variable de session , variable d'application - Utilisation d'une page maître - Passage de paramètres en GET et en POST d'une page ASP à une autre. Suite à cette initiation l'étudiant sera remis dans le contexte de départ (gestion d'un hôtel) pour réaliser les TP .
<b>02/11/2015</b> <b>13h-17h</b>	Réalisation du TP n°1 en asp.net/C# dans le contexte GESTION_HOTEL: Affichage de la disponibilité des chambres dans un dataGridView. Classes utilisées DataTable - DataGridView - Composant DLL contenant les classes métier développées dans le cadre de l'application locale.
<b>03/11/2015</b> <b>8h-12h</b>	Réalisation TP n°2 asp.net/C# : Afin de palier aux limites de la classe DataGridView le développeur décide de remplacer le DataGridView par un tableau dynamique HTML/C# construit dans la page ASP.NET. En cliquant sur une cellule du tableau les données relatives au numéro de chambre à la date et à l'état de la chambre sont passées à une page pour affichage à l'aide d'un javascript qui appelle une URL et transmet les données par la méthode GET
<b>09/11/2015</b> <b>13h-17h</b>	Suite du TP n°2 + le développeur remplace le calendrier ASP.NET par un calendrier réalisé à l'aide du framework javascript JQUERY. L'étudiant choisit le sujet de son premier PROJET pour l'épreuve E4 de l'examen dans le contexte fourni par le ministère : Entreprise GSB du secteur industrie pharmaceutique.
<b>10/11/2015</b> <b>8h-12h</b>	Nous commençons la mise en place de la plateforme d'examen dans notre ferme de serveurs :Installation d'un routeur sous DEBIAN 6 + 2 serveurs : 1 serveur de base de données MYSQL sous DEBIAN 6 et un serveur WEB sous WINDOWS SERVER 2008. L'architecture réseau est fournie dans un PDF avec 2 fichiers d'aide à la configuration des serveurs.
<b>16/11/2015</b> <b>13h-17h</b>	Recherche de sujet pour projet1 épreuve E4 examen - Interrogation (DS) - Correction DS au tableau + codage en C# de certaines réponses aux questions - Héritage - polymorphisme - constructeur - Classe abstraite.

<b>17/11/2015</b> <b>8h-12h</b>	<p>Les étudiants rentre dans la phase "mode projet" , pour la réalisation de leur projet N°1 concernant l'épreuve E4 jusqu'à la semaine du 18 Décembre. Cette période sera entrecoupée de session de cours/TP.</p> <p>Pour le projet le contexte est celui fourni par le ministère. Il s'agit de l'entreprise "GSB" du domaine de l'industrie pharmaceutique dont le contexte informatique économique et social est connu de l'étudiant. Pour cette épreuve les projets doivent être présentés dans un environnement informatique proche de celui qui leur est décrit dans le contexte fourni. Cet environnement est recréé à l'aide de machines virtuelles abritées dans notre hyperviseur "Vsphere". Au minimum chaque étudiant aura un routeur sous Linux Debian 7, et deux serveurs capables d'accueillir les serveurs web Apache 2 et IIS 7.5 ainsi que les SGBD MYSQL et SQL SERVER. En ce qui concerne le développement proprement dit, les apprentis doivent obligatoirement présenter une modélisation objets (Diagramme de classe et diagramme de séquence des parties du code les plus complexes et une modélisation des données (MCD/MLD). Pour le premier projet, la programmation se fera en ASP.NET/C# avec une base de données hébergée dans le SGBD MYSQL. L'architecture logicielle devra être rigoureusement celle étudiées en cours (classe d'affichage - classe de contrôle - classe métier). L'étudiant devra présenter outre la restitution des connaissances acquises lors des cours et travaux pratiques, une (des) particularité(s) innovante(s) qu'il choisira et qui sera(ont) un facteur déterminant au niveau de sa notation en fonction de leur difficulté et de la qualité de leur implémentation.</p>
<b>23/11/2015</b> <b>13h-17h</b>	Suite guidage projet 1 épreuve E4.
<b>24/11/2015</b> <b>8h-12h</b>	Suite guidage projet 1 épreuve E4.
<b>30/11/2015</b> <b>13h-17h</b>	Suite guidage projet 1 épreuve E4.
<b>01/12/2015</b> <b>8h-12h</b>	<p>Suite guidage Projet 1 épreuve E4.</p> <p>Nous avons fait un point sur la technologie LINQ dans C#. Linq toObject et Linq toXml qui pourront être utile dans la mise en œuvre des projets des étudiants. Les étudiants avaient un article à lire sur le sujet. Ces technologies permettent de traiter les fichiers XML et objets un peu de la même qu'une table dans une base de données relationnelle.</p> <p>Par ailleurs comme ces technologies sont basées sur un mode d'interrogation proche du SQL nous avons fait aussi quelques exercices de rappel en SQL (Opérations de regroupement, clause Exists, jointure where/inner Join ...)</p>
<b>07/12/2015</b> <b>13h-17h</b>	Suite guidage projet 1 épreuve E4.
<b>08/12/2015</b> <b>8h-12h</b>	Suite guidage projet 1 épreuve E4.
<b>14/12/2015</b> <b>13h-17h</b>	Suite guidage projet 1 épreuve E4.
<b>04/01/2016</b> <b>13h-17h</b>	<p>Cours sur les services Web SOAP. Un polycopié détaillant la mise en œuvre d'un Service Web SOAP a été fourni aux étudiants. Le sujet propose la mise en place d'un service WEB dans un premier temps auto-hébergé dans une application que je propose de développer. Sont détaillés les notions de proxy côté client, d'implémentation d'interface représentant le contrat de service exposé par le Service Web. Une capture de trames présente l'architecture de l'enveloppe SOAP-XML dans laquelle est encapsulée la réponse fournie au client. Afin de pratiquer les étudiants ont pour travail d'implémenter un web Service de connexion pour leur projet n°1 permettant l'autorisation d'accès à leur application web ASP.NET.</p>

<b>05/01/2016</b> <b>8h-12h</b>	Suite cours Web Service.
<b>11/01/2016</b> <b>13h-17h</b>	Suite guidage projet 1 épreuve E4.
<b>12/01/2016</b> <b>8h-12h</b>	Suite guidage projet 1 épreuve E4.
<b>18/01/2016</b> <b>13h-17h</b>	Suite guidage projet 1 épreuve E4.
<b>19/01/2016</b> <b>8h-12h</b>	Suite guidage projet n°1 épreuve E4.
<b>25/01/2016</b> <b>13h-17h</b>	Cours sur la sérialisation d'objet. Principe - les différents types de sérialisation binaire, XML, en mémoire - Les étudiants doivent réaliser un TP dans lequel ils doivent sérialiser une classe de contrôle qui contient une collection d'objets métier qui seront déclarés aussi "serializable" et qui seront sérialisés en cascade à partir de la sérialisation de la classe de contrôle. La sérialisation se fera dans un fichier dans un premier temps en binaire puis en XML. Nous profitons de ce cours pour faire un parallèle avec les web services SOAP dans lesquelles les réponses du web service sont sérialisés en mémoire au format SOAP-XML pour l'envoi au client.
<b>26/01/2016</b> <b>8h-12h</b>	Suite TP sérialisation d'objet . Nous reviendrons plus tard dans l'année sur la sérialisation au moment de l'étude des web service RESTFUL.
<b>01/02/2016</b> <b>13h-17h</b>	Cours sur les collection de type dictionnaire : intérêt par rapport aux listes - Exercice : utilisation des principales méthodes (ContainsKey et TryGetValue). Nous avons aussi fait un point sur les structures en les opposant aux classes (Exécution dans la pile plutôt que dans le tas - stockage par valeur contre stockage par référence) car une collection de type dictionnaire correspond à un ensemble de structure ayant deux attributs correspondant à la valeur stockées et sa clef.
<b>02/02/2016</b> <b>8h-12h</b>	Finir l'exercice sur la mise en place d'une collection de type dictionnaire. Nous avons repris une ancienne application dont la gestion des objets au niveau de la classe de contrôle se faisait avec des collections de type LIST que nous avons remplacées par des type DICTIONNARY.
<b>09/02/2016</b> <b>8h-12h</b>	Suite MVC - l'exercice est à terminer pour le prochain cours (après vacances février)
<b>29/02/2016</b> <b>13h-17h</b>	correction exercice MVC. Étude de la technologie AJAX (Asynchronous JavaScript and XML) dans le cadre d'un projet ASP.NET. L'étude d'AJAX se fait ici "à l'ancienne" c'est à dire en manipulant dans le code javascript la classe XmlHttpRequest pour une bonne compréhension du mécanisme. Il existe maintenant des framework par exemple jquery qui permettent une utilisation transparente de AJAX. - A partir d'un schéma compréhension des différents modules intervenant dans la technologie AJAX. - Mise en œuvre de AJAX dans un exercice : il s'agit ici de cliquer sur une zone d'un tableau HTML affichant les disponibilités des chambres et de mettre une chambre dans l'état "réservée" alors qu'elle est affichée "Libre" - Je profite de l'étude de AJAX pour montrer que dans un projet asp.net non MVC l'utilisation de AJAX n'est pas possible du fait de l'existence du champ caché "ViewState" qui enregistre l'empreinte de la page aspx qui interdit toute modification de la page côté client par AJAX.
<b>01/03/2016</b> <b>8h-12h</b>	Suite AJAX - MVC.

<b>08/03/2016</b> <b>13h-17h</b>	<p>Etude la mise en place d'une application web de type MVC (Modèle-Vue-Contrôleur)</p> <p>L'étude se fait à partir d'un document original fourni aux étudiants basé sur un exemple toujours sur le thème de la gestion d'un Hôtel développé en MVC ASP.NET - C# :</p> <ul style="list-style-type: none"> <li>- Compréhension globale du MVC</li> <li>- Compréhension du MVC dans une infrastructure Microsoft ASP.NET</li> <li>- Mise en place d'un projet MVC ASP.NET dans Visual Studio 2013</li> <li>- Étude de l'architecture du projet</li> <li>- Mise en place de la table de routage dans le fichier GLOBAL.ASAX et étude de la logique d'une route</li> <li>- Mise en place en place d'un contrôleur et d'une vue simples</li> <li>- Réutilisation du code (classes de contrôle et classes métier) de l'application GESTION HOTEL non MVC dans une bibliothèque de classes (DLL) afin de s'en servir comme modèle (M de MVC)</li> <li>- Distinction modèle de domaine / modèle de vue</li> <li>- mettre en place une page maître dans asp.net MVC</li> <li>- utilisation des HTML HELPERS afin de simplifier l'écriture du code HTML</li> </ul> <p>Ce travail sera compléter par une mise en oeuvre dans un petit exercice</p>
<b>15/03/2016</b> <b>8h-12h</b>	<p>Suite guidage projet 1 épreuve E4.</p>
<b>21/03/2016</b> <b>13h-17h</b>	<p>Suite guidage projet 1 épreuve E4.</p>
<b>22/03/2016</b> <b>8h-12h</b>	<p>Détermination projet n°2 : Comme pour le projet numéro 1 l'étudiant devra utiliser les technologies étudiées en cours mais pas forcément sur le même plateforme et pas forcément avec le même langage (par exemple MVC avec symphonie et AJAX avec jquery), la base de données est forcément MYSQL. L'étudiant devra fournir obligatoirement un travail sur une technologie non étudiée en cours ce que j'appelle l'innovation. L'étudiant doit fournir obligatoirement le diagrammes de classes complet de l'application avec une granularité moyenne ainsi que le diagramme de séquence des parties complexes de l'application. Un MCD ou un MLD est également demandé en ce qui concerne la couche de persistance. L'étudiant rédigera une petite note expliquant succinctement les fonctionnalités de son projet.</p> <p>Le descriptif du projet est déposé sur le réseau dans un fichier Excel qui détaille les choix de l'étudiant concernant le projet. Ce descriptif est validé par le professeur.</p>
<b>28/03/2016</b> <b>13h-17h</b>	<p>Les étudiants sont soit :</p> <ul style="list-style-type: none"> <li>- sur leur projet n°1 et le finissent</li> <li>- commencent leur projet n°2</li> </ul>
<b>29/03/2016</b> <b>8h-12h</b>	<p>Interrogation - correction interrogation</p>
<b>04/04/2016</b> <b>13h-17h</b>	<p>BTS BLANC</p>
<b>05/04/2016</b> <b>8h-12h</b>	<p>BTS BLANC</p>