|  |
| --- |
| **BTS SERVICES INFORMATIQUES AUX ORGANISATIONS** |

**E5 : PRODUCTION ET FOURNITURE DE SERVICES INFORMATIQUES**

**Durée : 4h**

|  |  |
| --- | --- |
|  |  |

Cas Music

Ce sujet comporte 17 pages dont 11 pages de documentation.

La candidate ou le candidat est invité-e à vérifier que le sujet qui lui a été remis est complet.

*Aucun matériel ni document autorisé*

**Ressources documentaires :**

1. Entretien avec Mme Lebec, gestionnaire de l’école de musique page 7
2. Extrait de l'état des lieux des logiciels utilisés dans les écoles de musique page 8
3. Extrait du script *donneesBase.sql* (base de données du logiciel Musicol) page 8
4. Extrait du script *structureBase.sql* (base de données du logiciel Musicol) page 9
5. Tarifs des tranches de quotient familial page 9
6. Schéma relationnel initial de la base de données Musicol page 10
7. Écran de présentation de la grille tarifaire (produit par *grilleView.php*) page 11
8. Éléments de documentation PHP page 11
9. Présentation et code de la page *grilleView.php* page 12
10. Diagramme de classes pour la planification des cours page 13
11. Code PHP des classes métier pour la planification des cours (extrait) page 13
12. Fiche de renseignement d'un instrument page 15
13. Fiche de prêts d'un instrument page 16
14. Liste des instruments enseignés et leur classe page 16
15. Modélisation conceptuelle de la base de données modifiée Musicol page 17

## Barème

|  |  |  |
| --- | --- | --- |
| **Mission 1** | Recherche de solutions pour la gestion de l’école de musique | 15 points |
| **Mission 2** | Évaluation du logiciel Musicol | 25 points |
| **Mission 3** | Adaptation du logiciel Musicol | 35 points |
| **Mission 4** | Évolution du logiciel Musicol | 25 points |
|  | **Total** | **100 points** |

Présentation du contexte

#### L'organisation cliente : l'école de musique municipale de la ville de D.

Inscrite dans le paysage culturel de la ville de D. (15 000 habitants) depuis une quarantaine d'années, l'école de musique municipale compte aujourd'hui quelques 300 usagers.

Ouverte à tous, enfants, adolescents, adultes, et couvrant le territoire de la communauté de communes, elle a pour vocation l'apprentissage de la musique.

Son projet d’établissement est axé vers les grandes missions suivantes :

* formation des musiciens (instrumentale, musicale, orchestrale, ateliers, etc.) ;
* diffusion, création (concerts, auditions) ;
* sensibilisation d’un large public et développement de partenariats locaux (éducation nationale, scène nationale voisine, maison des jeunes et de la culture, centres d’animation, pratiques amateurs, *master class*, conférences).

Dotée de locaux de qualité, l’école de musique dispose d’une équipe pédagogique de seize enseignants et propose au public différentes prestations d'enseignement musical.

L’école dispense des cours d’instruments individuels ou collectifs pour enfants, adolescents et adultes, en fonction des places disponibles. L’apprentissage est accessible à partir de 7 ans, sauf précision particulière pour certains instruments.

#### L'entreprise prestataire de services

La DSI - direction des systèmes d’information de la mairie de D. - regroupe deux personnes à temps plein afin de répondre aux besoins des différentes entités de la mairie réparties sur une dizaine de sites.

L’école de musique municipale est placée sous la tutelle de la mairie et profite à ce titre de services informatiques fournis par la DSI : bureautique, messagerie, accès au réseau et à internet, maintenance préventive des postes de travail et assistance aux utilisateurs.

La DSI assure également la fourniture de services spécifiques à l’école de musique : gestion des inscriptions, facturation et suivi des règlements.

#### Le projet

Avec les outils informatiques actuellement en place, la gestion de l’école de musique est jugée peu efficace par les personnels de l’école. La DSI de la mairie mène une étude en vue de moderniser ces outils.

Intégré-e en tant que salarié-e à la DSI, vous assistez M. Lejoyau, responsable de la DSI, et participez à cette étude et aux différentes missions qui la composent.

Un dossier documentaire vous est remis pour compléter votre information.

Mission 1 : Recherche de solutions pour la gestion de l’école de musique

Documents à utiliser : 1, 2

Actuellement la gestion de l’école de musique est assurée par Mme Lebec, secrétaire administrative, au moyen d’une application nommée GEM (gestion de l’école de musique), développée sous tableur et prenant en charge les inscriptions des usagers, la facturation et le suivi des règlements. GEM est constituée de différents modèles de tableaux que Mme Lebec ouvre, complète et enregistre, selon le besoin, dans un dossier adéquat.

En vue de la mise en place d’une nouvelle solution applicative, et afin de préparer la réflexion, M. Lejoyau doit établir un bilan du mode de gestion actuel de l’école de musique et rechercher une solution répondant à ses besoins.

Il a, pour cela, pris contact avec la fédération des écoles de musique associatives et a obtenu qu’elle mène une enquête auprès de ses adhérents afin de connaître les solutions applicatives que chacune utilise dans sa gestion propre.

Afin de préparer le choix d’une ou deux solutions applicatives à évaluer, M. Lejoyau vous missionne pour diagnostiquer l'application existante et analyser cet état des lieux.

Votre mission consiste à comparer les solutions applicatives utilisées par les autres écoles de musique.

### Votre mission

1. **Critiquer la sécurité du poste de travail de Mme Lebec.**
2. Identifier, à partir de l'enquête de la fédération sur les logiciels utilisés dans les écoles de musique, les 3 solutions d'architectures applicatives et techniques et donner pour chacune d'elles ses intérêts et ses limites.
3. Proposer deux critères supplémentaires qui pourraient être retenus afin de comparer les différentes solutions, en précisant à chaque fois l'utilité du critère.

Mission 2 : Évaluation du logiciel Musicol

Documents à utiliser : 1, 3, 4, 6

Sur la forge logicielle de l’Adullact (association des développeurs et des utilisateurs de logiciels libres pour l'administration et les collectivités territoriales), M. Lejoyau a pu observer l’existence du logiciel nommé *Musicol*. Il s'agit d'une application *web* développée en PHP autour du cadriciel (*framework*) libre nommé *CakePHP* basé sur une architecture MVC (modèle vue contrôleur).

Compte-tenu des réflexions déjà engagées, le responsable de la DSI pense qu’une évaluation de ce produit permettra de vérifier à quel degré il peut convenir au besoin. Vous commencez par télécharger la documentation du logiciel *Musicol*. Dans le manuel du développeur figure le schéma relationnel de la base de données utilisée par l’application ainsi qu’une copie des scripts de création de la base de données.

Mme Lebec, quant à elle, a besoin pour la gestion de l'école de connaître les éléments suivants :

* l'intitulé des instruments qui ne font l'objet d'aucun cours ;
* pour chaque instrument, le nombre d'élèves inscrits à des cours (par exemple : Batterie 7, Piano 19 …) ;
* le libellé des cours contenant moins de 6 inscrits.

Votre mission consiste à évaluer si *Musicol* couvre les besoins fonctionnels de l'école de Musique.

### Votre mission

1. **Indiquer s’il est possible d’enregistrer dans la base de données un paiement en 4 fois comme le pratique l'école. Si non, proposer les modifications à effectuer pour le permettre.**
2. **Indiquer si un élève peut s’inscrire à plusieurs cours. Si non, proposer les modifications à effectuer pour le permettre.**
3. **Écrire les requêtes permettant d'obtenir les éléments dont Mme Lebec a besoin.**

Mission 3 : Adaptation du logiciel Musicol

Documents à utiliser : 5, 6, 7, 8, 9, 10, 11

*IMPORTANT : la candidate ou le candidat peut choisir de présenter les éléments de code à l’aide du langage de programmation de son choix ou de pseudo-code algorithmique.*

L’évaluation de *Musicol* est globalement satisfaisante. Cependant, certaines règles de gestion sont absentes de *Musicol*. C’est le cas de la gestion des tranches de quotient familial qui ne prend pas en compte le cas particulier des résidents "hors communauté de communes". Ceux-ci ne sont pas soumis à la présentation de leur quotient familial et sont donc associés à une tranche spécifique qui les rend redevables du tarif le plus élevé.

M. Lejoyau avait adapté l’application *Musicol* en créant une tranche d'identifiant 0 caractérisée par un quotient minimal négatif, représentant la tranche des résidents "hors communauté de communes" et en renseignant les tarifs propres à cette tranche selon les différents types de cours. Par conséquent, l’écran de présentation de la grille tarifaire de *Musicol,* produit au moyen de la page *grilleView.php,* expose cette tranche comme une plage de valeurs négatives, là où l’indication EXT serait plus claire.

Vous avez proposé une autre solution qui a été acceptée et développée en partie : le tarif des résidents "hors communauté de communes" a été directement ajouté dans la table *type\_cours*, la tranche d'identifiant 0 a été supprimée de la table *tranches* et la méthode *GetPrixExterieur*() a été implémentée dans la classe *TypeCours*. Il ne reste plus qu'à modifier le code de la page *grilleView.php* de manière à prendre en compte les modifications liées à ce tarif.

En outre, comme décrit dans la plaquette de présentation de l’école, l’école de musique propose des cours individuels sur un instrument en particulier et des cours collectifs. Afin de différencier ces deux types de cours, M. Lejoyau a adapté la couche métier associée et produit le diagramme de classes métiers et le code PHP de ces classes. De même un planning des différents cours dispensés chacun des jours de la semaine étant établi chaque année, M. Lejoyau a ajouté une classe *Planning*.

Votre mission consiste à adapter la solution à ces modifications.

### Votre mission

1. Justifier le choix de stocker le tarif des résidents hors communauté de communes dans la table *type\_cours*.
2. Écrire les modifications demandées dans la page *grilleView.php* en ne reproduisant que la partie du code qui est modifiée, le cas échéant en situant les numéros de lignes.
3. **Écrire le code des méthodes *GetLibelle* des classes *CoursCollectif* et *CoursIndividuel* permettant de retourner le libellé du cours.**
4. **Écrire la méthode *ALieuLeJour* de la classe *Cours*.**
5. **Écrire la méthode *CoursJour* de la classe *Planning.***
6. **Indiquer de manière détaillée ce que retourne la méthode *CoursSemaine* de la classe *Planning*.**

Mission 4 : Évolution du logiciel Musicol

Documents à utiliser : 12, 13, 14, 15

*IMPORTANT : la candidate ou le candidat peut choisir de présenter les évolutions de la structure de la base de données sous la forme de son choix (schéma entité-association, diagramme de classes ou encore schéma relationnel).*

La communauté de communes alloue depuis de nombreuses années un budget annuel pour la maintenance et le renouvellement d'un parc d'instruments de musique. Ce parc est destiné à deux utilisations distinctes :

* des instruments qui restent à l'école de musique, par exemple des instruments ne pouvant facilement être déplacés (piano, harpe…) ou alors des instruments mis à disposition des professeurs pour leurs cours et ateliers ;
* des instruments qui sont prêtés gracieusement aux élèves débutants, ce qui évite aux familles d'investir, dès le début de l'apprentissage, dans un instrument souvent coûteux.

Pour le moment, la gestion du parc est manuelle et s'effectue de la façon suivante :

* Chaque instrument possède une étiquette discrètement collée sur laquelle est inscrit un code (par exemple le code **"*bas-2*"** signifie qu'il s'agit de la 2ème basse électrique du parc) et une fiche sur laquelle figure la description détaillée, une photo ainsi que les interventions de réglage et de réparation.
* Pour les instruments de prêt, une deuxième fiche récapitule les prêts aux élèves.

Dans le cadre de la maintenance du parc, l'école de musique travaille avec plusieurs professionnels partenaires (accordeurs de piano, luthiers, facteurs de divers instruments, …) à qui l'on confie les instruments pour des réglages ou des réparations. Au retour de chaque intervention, la fiche de l'instrument concerné est alors complétée.

En début d'année, les familles qui désirent emprunter un instrument font une demande motivée (non stockée dans la base de données). Une commission se réunit pour attribuer les instruments, la priorité étant donnée aux débutants. Le jour de la remise de l'instrument, l'école de musique établit un contrat avec la famille qui contient les éléments suivants :

* la date de signature et la date de fin de contrat généralement située vers le 7 septembre de l'année suivante ;
* un descriptif de l'état de l'instrument qui fait suite à une vérification minutieuse effectuée en présence de la famille dans lequel chaque détail est noté (par exemple un petit enfoncement dû à un choc,…) ;
* le numéro de la police d'assurance de la famille qui couvre les risques liés à l’usage de l'instrument (casse, perte ou vol).

Il est à noter qu'un instrument de prêt est fourni avec des accessoires (housse, câble jack, archet, amplificateur…) indispensables pour sa pratique et sa protection. Lors de la restitution de l'instrument, un an plus tard, le contrat est complété par l'état global. Si l'instrument doit subir un réglage ou une réparation non due à une usure normale, l'école de musique est en droit de facturer tout ou partie du prix de l'intervention, dans ce cas on notera la quotité demandée à la famille (100% pour tout le montant, 50% pour la moitié du montant,…).

### Votre mission

1. **Fournir un schéma de données présentant les évolutions de la structure de la base**

**de données relatives à la gestion du parc d'instruments : acquisitions, prêts,**

**interventions de réglage et de réparation.**

**Ressources documentaires**

# Entretien avec Mme Lebec, gestionnaire de l’école de musique

*M. Lejoyau* : **En quoi consiste votre activité au sein de l’école de musique ?**

*Mme Lebec* : J’assure la gestion administrative de l’école de musique pour ce qui est des inscriptions des usagers, de la facturation qui en découle et du suivi des règlements. Je réalise aussi la gestion des instruments de musique dont le nombre et la valeur ne sont pas à négliger. Certains sont à demeure et doivent être entretenus à la demande des professeurs (piano, orgue, etc.). D’autres sont ponctuellement prêtés aux élèves qui en font la demande.

*M. Lejoyau* : **Quels outils utilisez-vous pour réaliser vos activités ?**

*Mme Lebec* : Pour les inscriptions, la facturation et le suivi des règlements, j’utilise GEM (gestion d’école de musique). C’est un ensemble de tableaux *Excel* fournis sous forme de modèles dédiés à ces usages. Pour ce qui est de la gestion des instruments, je n’ai pas d’outils spécifiques. Je m’organise de manière personnelle avec une prise de notes sur cahier et des documents bureautiques (tableur et traitement de texte), selon les situations.

*M. Lejoyau* : **Y a-t-il d’autres utilisateurs de GEM au sein de l’école de musique ou de la mairie ?**

*Mme Lebec* : Non, je suis la seule utilisatrice. L’ensemble des documents GEM se trouvent sur mon poste de travail, dans un sous-dossier nommé GEM du dossier " Mes Documents ".

*M. Lejoyau* : **Votre poste de travail est-il utilisé par d’autres personnes ?**

*Mme Lebec* : Oui, parfois des professeurs utilisent le poste lorsque je m’absente pour déjeuner ou lors de mes congés. D’ailleurs, je ne suis pas très contente car j’ai plusieurs fois remarqué qu’après utilisation, mon bureau sous Windows avait changé et que de nouveaux éléments apparaissaient sur mon navigateur internet. J'ai même une fois perdu un mois de travail suite à une erreur de manipulation d'un professeur.

*M. Lejoyau* : **Comment fonctionne la facturation ?**

*Mme Lebec* : La facturation consiste à demander aux familles le paiement des échéances dues, puis à les enregistrer et les transmettre au trésor public pour encaissement. Selon les souhaits exprimés à l’inscription, le service offert peut être de payer en une à quatre fois. S’il y a plusieurs échéances demandées (le plus courant), les règlements sont enregistrés et seront transmis au trésor public pour encaissement à la date prévue. Les actes de facturation sont enregistrés ou initiés à partir du document modèle *facturation.xlt*. J’en utilise un par année. Chaque fichier de facturation est configuré de sorte à utiliser le fichier des inscriptions de la même année. Grâce à quoi, il calcule le montant global dû par famille puis le restant dû en fonction des différents règlements déjà transmis au trésor public.

*M. Lejoyau* : **Quelles difficultés rencontrez-vous avec GEM ?**

*Mme Lebec* : Pour ce qui est des inscriptions, je ressaisis les informations concernant le responsable autant de fois qu’il y a de membres préinscrits dans la même famille. C’est assez pénible et fastidieux. Et cela peut engendrer des situations surprenantes, si, par exemple, je fais une erreur dans une ressaisie. On pourra alors trouver une famille non repérée et une facturation non valide. GEM ne gère qu’une partie des besoins. Une fois les inscriptions réalisées, il faut encore gérer les aspects pédagogiques (de la responsabilité des enseignants) par exemple l’organisation des séances. De plus, en vue d'assurer une meilleure rentabilité, l'école souhaite axer sa communication sur les cours à faible effectif et pour cela, GEM ne permet pas de connaître de manière automatique le nombre d'inscrits par cours.

L’objectif final est de pouvoir fournir un emploi du temps individualisé à chaque élève.

# Extrait de l'état des lieux des logiciels utilisés dans les écoles de musique

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **École** | **CRITÈRES** | | | | **REMARQUES** |
| **Fonctionnalités** | | | **Coût** |
| *Édition documents comptables* | *Gestion des paiements* | *Gestion des adhérents* |
| **Cognac** | Musicol | Musicol | Musicol | Logiciel libre et gratuit  Installation et maintenance assurées par un membre de l'association. | Application web nécessite un serveur pour l'installation. Possibilité de récupérer les cotisations pour les intégrer dans la comptabilité. |
| **Morlaix** | Maestro | Maestro | Maestro | Logiciel commercial coûte 2.800€ | Installation locale.  Difficultés à basculer automatiquement les cotisations sur un logiciel de comptabilité, sinon il est très complet. |
| **Toulouse** | Excel | Excel | Application Access | Licence Office Pro 2013 499 € | Application locale développée par un stagiaire, permet le suivi des élèves mais la partie facturation nécessite des modifications. |
| **Langeais** | Ciel Association Évolution | Ciel Association Évolution | Ciel Association Évolution | Logiciel 108€ en 2014 Maintenance + assistance téléphonique : 345€/an | Application monoposte qui intègre la partie comptable. |
| **Fontaine** | Musicol | Musicol | Musicol | Logiciel libre et gratuit Installation et maintenance assurées par la DSI. | Application web nécessite un serveur pour l'installation. |
| **Couëron** | Calc (Open Office) | Calc | Site web | Hébergement du site : 15€/an | Site web en cours de développement par des adhérents. |
| **Champtoceaux** | OpenAssos |  | OpenAssos | Abonnement annuel : 576 € Licence + formation : 786 € | Logiciel fonctionne en mode SaaS – Software as a Service - , nécessite uniquement un navigateur.  Créé et géré par l’entreprise 2iOPENservice |

# Extrait du script *donneesBase.sql (base de données du logiciel Musicol)*

|  |  |
| --- | --- |
| **INSERT INTO instruments values**  ('', 'Batterie'),  ('', 'Piano'); |  |
| **INSERT INTO cours VALUES**  ('', 'batterie', 8,NULL, NULL,1,1),  ('', 'piano', 8,NULL, NULL,2,1),  ('', 'atelier jazz', 12, NULL, 10, NULL, 2); |  |
| **INSERT INTO tranches VALUES**  (1,0),(2, 251),(3, 426),(4, 681),(5, 935),(6, 1801); |  |
| **INSERT INTO couter VALUES**  (1, 1, 60.00),(1, 2, 96.00),(1, 3, 126.00),(1, 4, 192.00),(1, 5, 282.00),(1, 6, 330.00),  (2, 1, 30.00),(2, 2, 48.00),(2, 3, 63.00),(2, 4, 96.00),(2, 5, 141.00),(2, 6, 165.00); | |
|  | |

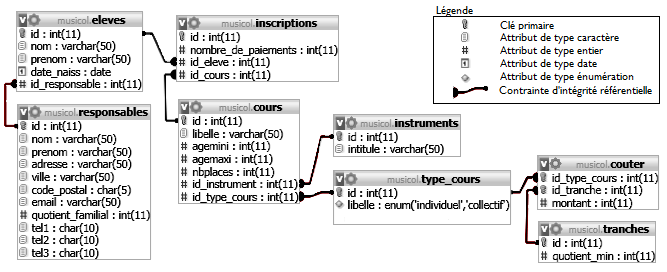
# Extrait du script s*tructureBase.sql (base de données du logiciel Musicol)*

1. **CREATE TABLE inscriptions**
2. (
3. id INTEGER(11) NOT NULL AUTO\_INCREMENT ,
4. nombre\_de\_paiements INTEGER(11) NOT NULL ,
5. id\_eleve INTEGER(11) NOT NULL ,
6. id\_cours INTEGER(11) NOT NULL ,
7. PRIMARY KEY (id) ,
8. FOREIGN KEY (id\_eleve) REFERENCES eleves (id) ,
9. FOREIGN KEY (id\_cours) REFERENCES cours (id)
10. );
11. **CREATE TRIGGER trigger\_inscriptions\_insert BEFORE INSERT ON inscriptions**
12. FOR EACH ROW
13. BEGIN
14. IF NEW.nombre\_de\_paiements < 1 or NEW.nombre\_de\_paiements > 3
15. THEN
16. signal sqlstate '16440' set message\_text = 'Insertion interdite';
17. END IF;
18. END;
19. **CREATE TABLE cours**
20. (
21. id INTEGER NOT NULL AUTO\_INCREMENT ,
22. libelle VARCHAR(50) NOT NULL ,
23. agemini INTEGER(11) NOT NULL ,
24. agemaxi INTEGER(11) NULL ,
25. nbplaces INTEGER(11) NULL ,
26. id\_instrument INTEGER(11) NULL,
27. id\_type\_cours INTEGER(11) NOT NULL,
28. PRIMARY KEY (id) ,
29. FOREIGN KEY (id\_instrument) REFERENCES instruments (id),
30. FOREIGN KEY (id\_type\_cours) REFERENCES type\_cours (id)
31. );

# Tarifs des tranches de quotient familial

|  |  |  |  |
| --- | --- | --- | --- |
| **Q.F.** | **Tranche** | **Cours individuel** | **Cours collectif** |
| ***Résidents de la communauté de communes*** | | | |
| 0 à 250 | A | 60 | 30 |
| 251 à 425 | B | 96 | 48 |
| 426 à 680 | C | 126 | 63 |
| 681 à 934 | D | 192 | 96 |
| 935 à 1800 | E | 282 | 141 |
| + de 1800 | F | 330 | 165 |
| ***Résidents hors de la communauté de communes*** | | | |
|  | EXT | 417 | 209 |

# Schéma relationnel initial de la base de données Musicol

****

**responsables (**id, nom, prenom, adresse, ville, code\_postal, email, quotient\_familial, tel1, tel2, tel3**)**

*Clé primaire : id*

**eleves (**id, nom, prenom, date\_naiss, id\_responsable**)**

*Clé primaire : id*

*Clé étrangère : id\_*responsable *en référence à id de responsables*

**instruments** (id, intitule)

*Clé primaire : id*

**type\_cours** (id, libelle) *//libellé a pour valeur* ***individuel*** *ou* ***collectif***

*Clé primaire : id*

**cours** (id, libelle, agemini, agemaxi, nbplaces, id\_instrument, id\_type\_cours)

*Clé primaire : id*

*Clé étrangère : id\_instrument en référence à id de instruments*

*Clé étrangère : id\_type\_cours en référence à id de type\_cours*

**tranches** (id, quotient\_min)

*Clé primaire : id*

**couter** (id\_type\_cours, id\_tranche, montant)

*Clé primaire : id\_type\_cours, id\_tranche*

*Clé étrangère : id\_tranche en référence à id de tranches*

*Clé étrangère :* id\_type\_cours *en référence à id de type\_cours*

**inscriptions** (id, nombre\_de\_paiements, id\_eleve, id\_cours)

*Clé primaire : id*

*Clé étrangère : id\_eleve en référence à id de eleves*

*Clé étrangère : id\_cours en référence à id de cours*

# Écran de présentation de la grille tarifaire (produit par *grilleView.php*)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tarifs de l'année 2014**   |  |  |  | | --- | --- | --- | | **Quotient familial**  Cette indication doit être remplacée par EXT dans la nouvelle version de la page grilleView.php | **Cours individuel** | **Cours collectif** | | **-9999 à -1** | **417** | **209** | | **0 à 250** | **60** | **30** | | **251 à 425** | **96** | **48** | | **426 à 680** | **126** | **63** | | **681 à 935** | **192** | **96** | | **936 à 1800** | **282** | **141** | | **1801 et supérieur** | **330** | **165** | |

# Éléments de documentation PHP

Implémentation d’une collection d’objets :

$maCollection = array() ; *// Création de la collection*

$maCollection[] = $objetUn ; *// Ajout d’un objet à la collection*

$maCollection[] = $objetDeux ;

$maCollection[] = $objetTrois ;

foreach($maCollection as $unObjet) *// Parcours de la collection*

{

*// Utilisation de l’objet $unObjet*

echo $unObjet->GetLibelle() ; *// Appel d’une méthode de l’objet*

}

Implémentation d’un dictionnaire<clé, valeur> de <chaîne, objet> :

*Remarque : une particularité du langage PHP est de permettre de parcourir un dictionnaire en récupérant à la fois la clé et la valeur de chaque élément.*

$monDico = array() ; *// Création du dictionnaire, ou tableau associatif*

$monDico["un"] = $unObjet1 ; *// Ajout du couple <"un", unObjet1> au dictionnaire*

$monDico["deux"] = $unObjet2 ; *// Ajout du couple <"deux", unObjet2  > au dictionnaire*

$monDico["trois"] = $unObjet3 ; *// Ajout du couple <"trois", unObjet3  > au dictionnaire*

$unObjet = $monDico["deux"] ; *// $unObjet est affecté de unObjet2*

if (isset($monDico["deux"]))… *// Ce test retourne vrai, car la clé "deux" existe*

if (isset($monDico["quatre"]))… *// Ce test retourne faux, la clé "quatre" n’existe pas*

foreach($monDico as $laCle=>$unObjet) *// Parcours du dictionnaire*

{

// Utilisation de l’objet $unObjet

echo $laCle; *// Affichage de la clé*

echo $unObjet->UneMethode(); *// Appel d’une méthode de l’objet*

}

# Présentation et code de la page *grilleView*.*php*

La vue ***grilleView.php*** permet l’affichage de la grille tarifaire. Elle dispose de l’année courante et des 3 tableaux suivants :

* Le tableau associatif (ou dictionnaire) **$lesTranches** contient l’ensemble des tranches de quotient familial : l’indice est l’identifiant de tranche ; la valeur est un objet de la classe *Tranche* exposant deux méthodes publiques pour obtenir l’une le quotient minimum de la tranche, l’autre le quotient maximum de la tranche.
* Le tableau **$lesTarifs** contient l’ensemble des tarifs de l’école, présentant deux dimensions : la première est l’identifiant du type de cours, la seconde est l’identifiant de la tranche. La valeur contenue est le tarif associé. Par exemple : *$lesTarifs[1][2]* contient le tarif du type de cours 1 pour la tranche 2 (quotient compris entre 251 et 425).
* Le tableau associatif (ou dictionnaire) **$lesTypesCours** contient l'ensemble des types de cours : l'indice est l'identifiant de typeCours ; la valeur est un objet de la classe *TypeCours*.

<div class=**"tarif"**>

<h2>**Tarifs de l'année** <?php **echo** $annee ?></h2>

<table id=**"grille"**>

<thead>

<tr>

<th>**Quotient familial**</th>

<th>**Cours individuel**</th>

<th>**Cours collectif**</th>

</tr>

</thead>

<?php $i=0;

**foreach**($lesTranches **as** $tran\_id=>$tranche){

**if**($i % 2){ *// une ligne sur deux a le fond grisé*

**echo** "<tr class="altrow">";

}

**else**{

**echo** "<tr>";

}

$i++;

**echo** "<td>";

**if** ($i<**count**($lesTranches)) {

**echo** $tranche->getQuotientMin() .

" à ". $tranche->getQuotientMax();

}

**else** {

**echo** $tranche->getQuotientMin() . " et supérieur";

}

**echo** "</td>";

**foreach** ($lesTypesCours **as** $typeCours\_id=>$typeCours){

**echo** "<td>";

**echo** $lesTarifs[$typeCours\_id][$tran\_id];

**echo** "</td>";

}

**echo** "</tr>";

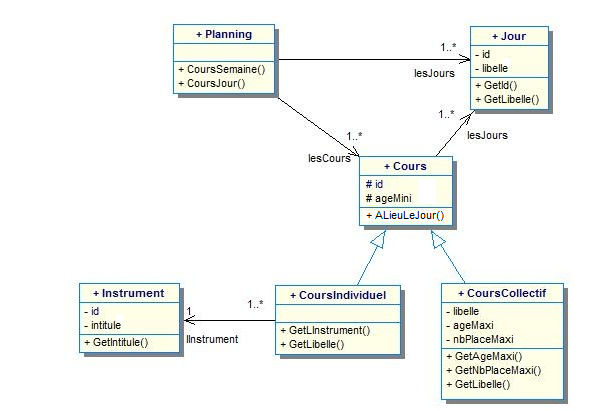
}

?>

</table>

</div>

# Diagramme de classes pour la planification des cours



# Code PHP des classes métier pour la planification des cours (extrait)

**<?php**

**class Jour{**

private $id;

private $libelle; *// libellé du jour*

public function \_\_construct($unId, $unLibelle){ *// constructeur*

$this->id=$unId;

$this->libelle=$unLibelle;

}

public function GetId(){

return $this->id;

}

public function GetLibelle(){

return $this->libelle;

}

}

**class Instrument{**

private $id;

private $intitule; *// intitulé de l’instrument*

public function \_\_construct($unId,$unIntitule){ *// constructeur*

$this->id=$unId;

$this->intitule=$unIntitule;

}

public function GetIntitule(){ *// retourne l'intitulé de l'instrument*

return $this->intitule;

}

}

**abstract class Cours{**

protected $id;

protected $ageMini;

protected $lesJours; *// collection des jours où a lieu le cours*

public function \_\_construct($unId,$unAgeMini){ *// constructeur*

$this->id=$unId;

$this->ageMini=$unAgeMini;

$this->lesJours=array();

}

public function ALieuLeJour(Jour $unJour){ }

*//renvoie vrai si le cours a lieu le jour passé en paramètre*

}

**class CoursCollectif extends Cours{**

private $libelle; *// libellé du cours collectif*

private $ageMaxi;

private $nbPlacesMaxi;

*// constructeur*

public function \_\_construct($unId,$unAgeMini,$unLibelle,$unAgeMaxi,$unNbPlacesMaxi){

parent::\_\_construct($unId,$unAgeMini);

$this->libelle=$unLibelle;

$this->ageMaxi=$unAgeMaxi;

$this->nbPlacesMaxi=$unNbPlacesMaxi;

}

public function GetLibelle(){

*// retourne le libellé du cours collectif*

}

public function GetAgeMaxi(){

return $this->ageMaxi;

}

public function GetNbPlacesMaxi(){

return $this->nbPlacesMaxi;

}

}

**class CoursIndividuel extends Cours{**

private $lInstrument; *//instrument concerné par le cours individuel*

public function \_\_construct($unId,$unAgeMini,$unInstrument){ *// constructeur*

parent::\_\_construct($unId,$unAgeMini);

$this->lInstrument=$unInstrument;

}

**public function GetLInstrument()**{

return $this->lInstrument ;

}

**public function GetLibelle(){**

*// retourne le libellé du cours individuel qui est en fait l’intitulé de l’instrument*

}

}

**class Planning**{

public $lesJours; *// collection des jours de la semaine*

public $lesCours; *// collection des cours*

public function \_\_construct(){ *//constructeur*

$this->lesJours=array();

$this->lesCours=array();

}

**public function CoursJour(Jour $unJour){**

*// retourne une collection d’objets Cours contenant les cours du jour $unJour*

}

**public function CoursSemaine(){**

$lePlanningComplet=array();

foreach($this->lesJours as $leJour){

$lePlanningComplet[$leJour->GetLibelle()] =$this->CoursJour($leJour);

}

return $lePlanningComplet;

}

?>

# Fiche de renseignement d'un instrument

**bas-2** (Basse électrique)



Date d'achat : 06/06/2011

Prix d'achat : 380 €

Marque : Squier Modèle : Jazz Bass

Numéro de série : Q 07219

Couleur dominante : Noire

Utilisation : prêt

**Classe d'instruments :** Instruments amplifiés

**Interventions**

**N° Date début Date fin Professionnel Travaux effectués Prix**

1 12/09/2013 14/09/2013 Hall de musique Réglage du manche 30€

24 rue Amiral Ronarc'h

29000 Quimper

2 12/09/2014 12/09/2014 Ets Poënot Réparation de

Port Rhu la prise jack 10€

29100 Douarnenez

3 14/09/2014 15/09/2014 Hall de musique Réglage du manche 32€

24 rue Amiral Ronarc'h

29000 Quimper

# Fiche de prêts d'un instrument

**bas-2** (Basse électrique)



Accessoires fournis : housse souple, jack, ampli 10W

**Prêts**

**Elève Date début Date fin N° intervention quotité**

**suite à un prêt famille**

Horellou Simon 14/09/2011 07/09/2012

15, rue Ar Ster

29100 Pouldergat

Queffelec Annaël 15/09/2012 08/09/2013 1 0%

31 rue Édouard Branly

29100 Douarnenez

Le Boulch Thibault 16/09/2013 09/09/2014 2 100%

56 rue Victor Hugo 3 0%

29100 Douarnenez

**Remarque** : Le prêt à l'élève Simon Horellou n'a nécessité aucune intervention.

# Liste des instruments enseignés et leur classe

|  |  |
| --- | --- |
| **Classes d'instruments** | **Instruments** |
| Claviers | Orgue |
| Piano |
| Instruments  amplifiés | Clavier amplifié |
| Guitare électrique |
| Basse électrique |
| Bois | Saxophone |
| Clarinette |
| Flute traversière |
| Cuivres | Trombone |
| Trompette |
| Tuba |
| Cordes | Violon |
| Violoncelle |
| Harpe celtique |
| Percussions | Batterie |

# Modélisation conceptuelle de la base de données modifiée Musicol

***La modélisation conceptuelle est volontairement fournie sous les deux représentations les plus courantes.***

|  |
| --- |
| **Schéma entité-association** |
| **MCD initial V15.jpg** |

|  |
| --- |
| **Diagramme de classes** |
|  |