

## MIF 13 – Programmation Web – Examen

Durée : 1 heure 00 – Documents autorisés – Ordinateurs, calculatrices et téléphones portables interdits

### Questions de cours (barème : 10 points)

Répondez aux questions ci-dessous en justifiant vos réponses.

1. À quel type de modèle MVC (push-based ou pull-based) correspond le fonctionnement suivant : « un navigateur envoie une requête à une application Web. Cette requête est transférée à une JSP qui utilise un bean pour générer sa réponse. » ?
2. Quel intérêt présente l'utilisation de la méthode `forward()` d'un `RequestDispatcher` en MVC push-based ?
3. Quel outil utiliseriez-vous pour reconcevoir en Web une application dont vous disposez d'une version en Java / Swing ?
4. En quoi le typage dynamique permet-il de mettre en place des fonctions façades en JavaScript ?
5. Pourquoi ne peut-on pas utiliser AJAX pour récupérer dans un client Web des événements provenant d'un serveur ?

### Étude de cas (barème : 10 points)

Comme chaque année, l'UCBL propose à ses usagers un service de cartes de vœux électroniques. Cette année, c'est vous qui êtes chargé(e) de concevoir cette application Web. Cette application fonctionne de la façon suivante : un utilisateur se connecte à l'application par le CAS. Une fois authentifié, il arrive sur la page d'accueil. De là, il peut visualiser l'aspect général de la carte de vœux et a accès à un lien lui permettant d'envoyer cette carte par messagerie électronique. S'il clique sur ce lien, il accède à un formulaire qui possède les champs suivants :

- Adresse de l'expéditeur (déjà remplie en fonction de l'adresse mail de l'utilisateur dans l'annuaire LDAP et non modifiable)
- Adresse du destinataire
- Titre de la carte
- Texte de la carte
- Choix de la langue de la carte (« français » ou « anglais »)
- Case à cocher pour recevoir ou non un message de confirmation
- Bouton envoyer

S'il clique sur le bouton envoyer, la carte est envoyée à l'adresse indiquée et l'utilisateur est renvoyé vers la page d'accueil. Cette page affiche la carte envoyée, plus un message de confirmation / erreur d'envoi. S'il a demandé une confirmation, il reçoit un message électronique lui indiquant le résultat de cette opération.

Pour réaliser cette application, vous disposez des composants suivants :

- Une classe `Carte` qui ne possède qu'une méthode statique nommée `getHtml(String titre, String texte, String langue)`. Cette méthode génère le code HTML de la carte de vœux à envoyer.
  - Une classe `Message` possédant les champs :
    - o nom de l'expéditeur
    - o adresse du destinataire
    - o titre du message (peut être le titre de la carte, mais pas nécessairement)
    - o texte (ou code HTML) du message
- La classe `Message` contient également une méthode `envoyer()` qui s'occupe de l'envoi par messagerie électronique.
- Une classe `User` qui représente l'utilisateur authentifié. Cette classe permet d'accéder à l'ensemble des champs de l'annuaire LDAP pour l'utilisateur. Pour l'instancier, il suffit d'appeler son constructeur, avec en paramètre le login de l'utilisateur. Les champs sont ensuite accessibles à l'aide d'accesseurs (« getters »).

Cette application ne s'appuiera pas sur un framework et mettra en œuvre un pattern Web **MVC pull-based**.

### Conception (barème : 4 points)

6. Listez les différents éléments nécessaires au fonctionnement de votre application, en indiquant leur nature : pages statiques, scripts, servlets, JSP, JavaBeans (précisez le scope)...
7. Décrivez, à l'aide de diagrammes UML appropriés, leurs communications lorsqu'un utilisateur authentifié clique sur le bouton de validation du formulaire, à partir de la réception des requêtes du client. Les éléments du SI externes à votre application seront modélisés comme des boîtes noires.

### Programmation (barème : 6 points)

8. Écrivez la page d'accueil de cette application.