**Réponse questions e-commerce (Partie complément)**

**Question 1:**

**a) Décrire en les comparant les trois protocoles de messagerie électronique SMTP, POP3 et IMAP.**

**b) Quelles différences IMAP apporte-t-il par rapport à POP3 et expliquer en termes d'agents le cheminement d'un message électronique depuis sa rédaction par l'auteur jusqu'à sa lecture par le destinataire.**

**c) Expliquer comment ces agents laissent des traces dans les headers des messages et comment on peut les retrouver en programmation Java pour un message reçu.**

**Question 2:**

**a) Dans les librairies javax.mail, décrire et expliquer les rôles des interfaces, classes abstraites et classes instanciables nécessaires à l'envoi d'un message sans pièces attachées.**

**b) Dresser le tableau synoptique de description synthétique des protocoles classiques SMTP, POP3, IMAP, HTTP.**

**Question 3:**

**a) Dans les librairies javax.activation, décrire et expliquer les rôles des interfaces, classes abstraites et classes instanciables nécessaires à l'envoi d'un message avec pièces attachées.**

**b) Donnez un exemple de programmation pour la récupération d’un mail contenant plusieurs pièces attachées (fichiers texte, image).**

**c) Expliquer en quoi l'examen des types MIME et de la "disposition" des pièces attachées intervient dans la programmation de cette réception de messages composites.**

**Question 4:**

**Décrire et expliquer le contenu d'un fichier apk. Quels sont les outils qui interviennent dans sa construction ?**

**En particulier, expliquer le contenu et le rôle de la classe R.java.**

**Question 5:**

**Décrire l'architecture Android à 5(6) couches en en décrivant les éléments principaux et décrire le bytecode particulier utilisé par les JVMs d'Android.**

**Expliquer les caractéristiques des JVM Dalvik et ART et ce qui les différencie de la JVM classique.**

**Question 6:**

**Décrire la structure des répertoires (et leur principal contenu) qui correspondent à une application Android comportant une activité classique avec images, couleurs, menus, textes personnalisés selon la langue et la manière d'en faire usage dans le code Java d'une activité.**

**Question 7:**

**Sous Android, décrire à partir d'un exemple d'application ne comportant qu'une seule activité, les diverses manières de programmer la réponse à un événement graphique et la manière de gérer un composant comme une ListView selon le modèle MVC.**

**Question 8:**

**Sous Android, décrire à partir d'un exemple d'application ne comportant qu'une seule activité l'utilisation d'un AsyncTask qui accède à un serveur vendant des actions.**

**Expliquer le fonctionnement du modèle MVC pour les composants graphiques d'Android à partir de l'affichage par l'activité des actions achetées.**

**Question 9:**

**Expliquer comment une communication réseau est sécurisée par SSL/TLS (SSL-Record), une fois le SSL-Handshake réalisé (donc sans le décrire).**

**Pourquoi parle-t-on de "sous-protocole(s)" ?**

**Qu'entend-on par "variables de session" et "variable de connexion".**

**Question 10:**

**Décrire en détails le SSL-Handshake.**

**En particulier, expliquer en quoi on peut considérer qu'il est adaptable à différentes situations.**

**Pourquoi parle-t-on de "sous-protocoles" ?**

**Question 11:**

**Expliquer le code Java utilisant SSL à substituer au code classique d'un serveur TCP/IP dont on désire sécuriser les communications.**

**En particulier, décrire les keystores nécessaires au bon fonctionnement de ce code et leur utilisation, côté serveur et côté client.**

**Pourquoi voit-on intervenir dans javax.net.\* un si grand nombre d'interfaces et de classes abstraites ?**

**Question 12:**

**Expliquer ce qu'est un « intent » et les différents mécanismes qui permettent de les mettre en œuvre.**

**Question 13:**

**Expliquez et illustrez les différences (améliorations, mécanisme, …) entre les versions TLS précédentes la 1.3 et TLS 1.3**

**Question 14:**

**Termes à définir brièvement, remettre dans son contexte, acronyme de … , utilité(s), rôle(s), exemple d’application.**

**1) MFS**

**2) RFC822**

**3) SMTP**

**4) POP3**

**5) IMAP**

**6) APK**

**7) RMTA**

**8) FMTA**

**9) RMDA**

**10) MRA**

**11)**

**12) RMUA**

**13) SMUA**

**14) R.java**

**15) .arsc**

**16) JVM/ART**

**17) JVM/ART**

**18) FindViewById**

**19) Intent**

**20) AsyncTask**

**21) SSL-Record**

**22) SSL-Handshake**

**23) Keystore**