

---

## TD : CONCEPTION D'APPLICATIONS CARTE ET PROGRAMMATION BASICCARD

Concepteur : Didier DONSEZ

But : Ce TD vise à vous familiariser avec la programmation BasicCard et à la conception d'application Carte.

Durée : 3 heures (gérez votre temps !)

Conseil : Des nombreuses solutions à ce TD se trouvent dans les exemples !!!

Logiciels : BasicCard DK sous Windows

Documents <http://www-adele.imag.fr/~donsez/cours/basiccard.pdf>, Documentations BasicCard

**ATTENTION : Prenez la précaution de ne jamais mettre une carte dans l'état RUN : la carte ne pourrait plus jamais être reprogrammée. Cet état est prévu pour livrer la carte au porteur !**

---

## INSTALLATION DU KIT DE DEVELOPPEMENT BASICCARD

Chargez et installez le dernier Kit de Développement BasicCard.

Branchez votre lecteur sur le port COM1 (cf 5.4 Physical and Virtual Card Readers)

Testez les exemples fournis (echotest, debitcard, fileio, ...)

Testez la cryptographie avec echotest (cf. 8.9 et 3.17) et IDEA encryption

---

## UNE CARTE DE FIDELITE

Ecrivez une application carte Loyalty qui implémente une carte de fidélité simple.

Une première application terminal correspond à l'application de l'émetteur qui initialise la carte.

Une seconde application terminal correspond à l'application du commerçant qui crédite la carte. Le commerçant s'authentifie avec des moyens cryptographiques.

---

## UNE CARTE DE JEU (PARI)

La carte Gambling est une carte de pari qui permet de gagner (et perdre) des sommes qu'un joueur (gambler) peut parier. Dans ce pari, la carte joue le rôle de la banque au Casino. La carte possède un moteur de pari générique et qui s'adapte à tout jeu d'argent (bandit manchot, poker, pile ou face, roulette, boule, ...).

Comment peut-on réaliser la généricité du moteur de pari ? (Conseil : reprenez les règles de calcul des gains de chaque jeu)

Ecrivez l'application carte Gambling.

Une première application terminal correspond à l'application de l'émetteur (issu) qui initialise la carte avec le crédit initial, le taux de rapport des gains et le générateur de nombre aléatoire.

Une seconde application terminal correspond à l'application du joueur (gambler) qui permet au parieur de miser une certaine somme d'argent en débitant la somme de la carte et en la recreditant des gains gagnés (en cas de gain). Cette application permet également de recrediter sa carte de manière sécurisée et récupérer ses gains.

Une mauvaise programmation peut entraîner des pertes ou des créations d'argent ! Comment fiabiliser l'application carte gambler ?

Remarque : Un point de départ pour l'application Gambling est le Porte Monnaie Electronique.

---

## MANIPULATION DES FICHIERS

Manipulez les fichiers d'une carte avec les programmes terminal présents dans exemples\enhance (ZCDIR, ZCDEL, ZCCOPY, ...)

Modifiez ZCCOPY pour y ajouter un paramètre -b qui spécifie le facteur de blocage (taille de blocs d'octets à écrire ou lire) à utiliser pour la taille des lectures/écritures depuis/vers la carte (par défaut, le facteur est positionné à 64 octets dans l'exemple). Quel est le facteur de blocage maximum qui puisse être utilisé ?

Réalisez des mesures de débit d'écriture et de lecture d'un fichier (benchmark) en fonction du facteur de blocage.

Suggestion : Vous rajouterez un paramètre -t qui donne le temps passé à exécuter la copie au moyen de la

---

procédure `Sub GetDateTime (DT As DateTime)` et de la fonction `TimeInterval (StartTime As DateTime, EndTime As DateTime) As Long` (cf 6.8.1 Timing Functions).  
Donnez le graphe en compte rendu.

---

## **MANIPULATION DES FICHIERS**

Ecrivez une application terminal qui utilise une `BasicCard` pour implémenter une table hiérarchique de signets (*bookmark*) Web (Titre+URL)

---

## **QUELQUES MANIPULATIONS DEPUIS OCF (JAVA)**

Lancez l'outil `APDUTool` que vous avez développé en Java lors du TD sur OCF  
Envoyez des commandes APDU à votre `BasicCard` sur laquelle l'application `DébitCredit` est installée (Command `&H80 &H0A GetCardData (Amount As Long, PINCount, Customer$)`).  
Donnez dans le compte rendu la liste des commandes APDU !  
Envoyez également des commandes APDU à votre `BasicCard` pour consulter les fichiers présents.

---

## **INTERFACE GRAPHIQUE POUR GAMBLER**

Ecrivez un `CardService` OCF pour l'application `BasicCard` carte `Gambling`.  
Ecrivez une applet/application Java qui offre une interface graphique (AWT/Swing) pour les jeux (un par onglet) de pile-ou-face et de bandit manchot (utilisant le `GamblingCardService`)

---

## **MINI-PROJET : BASICCARDFILECARDSERVICE**

Ecrivez un `CardService` OCF pour accéder aux fichiers d'une `BasicCard`.

---

## **MINI-PROJET : BOOKMARKCARDSERVICE**

Ecrivez un `CardService` OCF pour accéder aux signets stockés dans les fichiers une `BasicCard`.  
Ecrivez une servlet qui affiche en HTML le contenu complet des signets stockés dans une `BasicCard`.

---

## **MINI-PROJET : GENERATEUR DE CARDSERVICE POUR APPLICATION BASICCARD**

Réalisez un générateur de `CardService` OCF à partir de la déclaration (fichier `.DEF`) des commandes d'une application `BasicCard`.  
Conseil : Utilisez `JavaCC`.

---