**Institut Supérieur d’Informatique et de Mathématiques de Monastir**



**Niveau :**

**Cycle d’Ingénieur**

**en informatique**

**Groupe :**

**TD1**

**Réaliser par :**

**Nour elhouda salem**

**Ahmed Chebbi**

**Enseignant : Mr Ramzi Mahmoudi**



Table des matières

Table des matières iii

Chapitre 1. Environnement de développement JavaCard 4

I.1 Outils logiciels nécessaires 4

I.1.1 Identification de sys d'exploitation: 4

I.1.2 Telechargement de java card kit 2.2.2: 5

I.1.3 Telechargement de java SE:…………………………………………………………..5

I.1.4 Telechargement de ECLIPSE IDE:………………………………………………………………………….7

I.1.5 Telechargement de plugins d'integration:……………………………………………………….8

I.2 Instruction d'installation: 8

I.2.1 Installation d'ECLIPSE: 8

I.2.2 Installation du Java Card Development Kit 2 .2.2 : 13

I.2.3 Mise à jour des plugins Eclipse-JCDE: 17

I.2.4 Conclusion : 18

Chapitre II. Développement d’une application coté Serveur 19

II.1.1 Création d’un nouveau Projet JavaCard : 19

II.1.2 Création d’une applet Javacard: 21

II.2 Codage de notre applet : 22

II.3. Outils de Simulation : 24

II.3.1 JCWDE simulateur sans conservation d'etat: 24

II.3.2 CREF simulateur avec conservation d'etat: 28

Chapitre III. Programmation d’une application coté client 31

III.1 Création de l’application client sous Eclipse : 31

III.1.1 Création d’un nouveau projet Java : 31

III.1.2 Ajout de la librairie « apduio » dans le classpath : 32

III.1.3 Création de la classe principale : 33

III.2 Utilisation de l'application cliente avec un simulateur – JCWDE : 38

Chapitre IV. Réalisation du Mini-Projet 41

IV.1 Interface Graphique ( Partie Client ) : 41

IV.1.1 WindowBuilder Introduction : 41

IV.1.2 Installation de WindowBuilder : 42

IV.1.3 Comment utiliser WindowBuilder : 44

IV.1.4 Les interfaces de notre application Client : 49

IV.1.5 Implémentation des méthodes nécessaires : 57

IV.2 Partie Serveur (Back-End) : 59

IV.2.1 Déclaration des variables et des constantes : 59

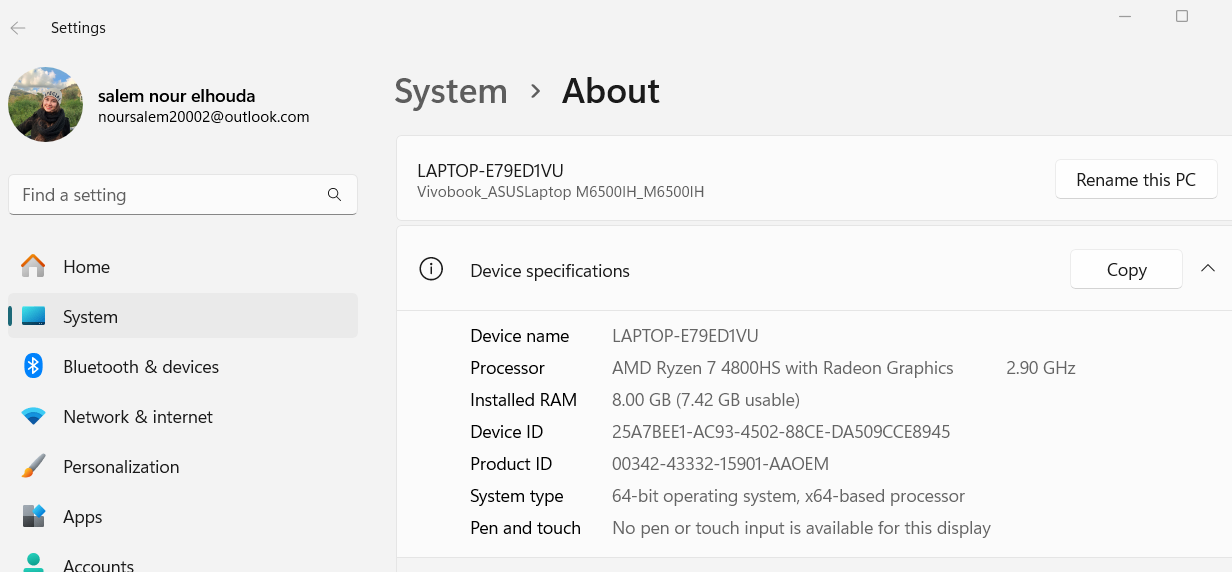
IV.2.2 Implémentations des méthodes nécessaires : 59

Chapitre 1. Environnement de

développement JavaCard

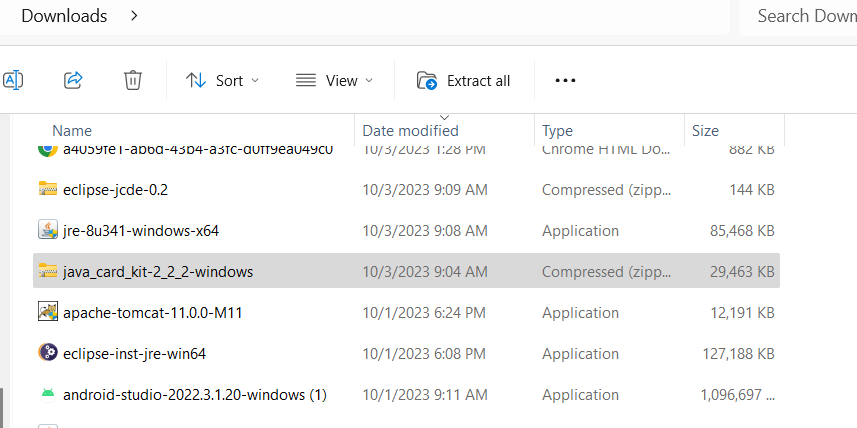
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I.1 Outils logiciels nécessaires :  
  
I.1.1**.** Identifiez votre système d’exploitation ainsi que son type



I.1.2 Telechargement Java Card Development Kit 2.2.2 :

Le Java Card Development Kit 2.2.2 : archive **java\_card\_kit-2\_2\_2-windows.zip** à partir du site d’Oracle : http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javame- 419430.html

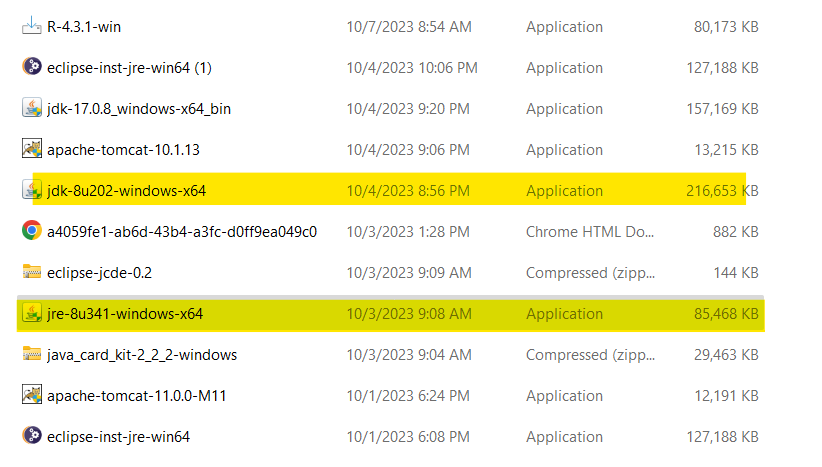


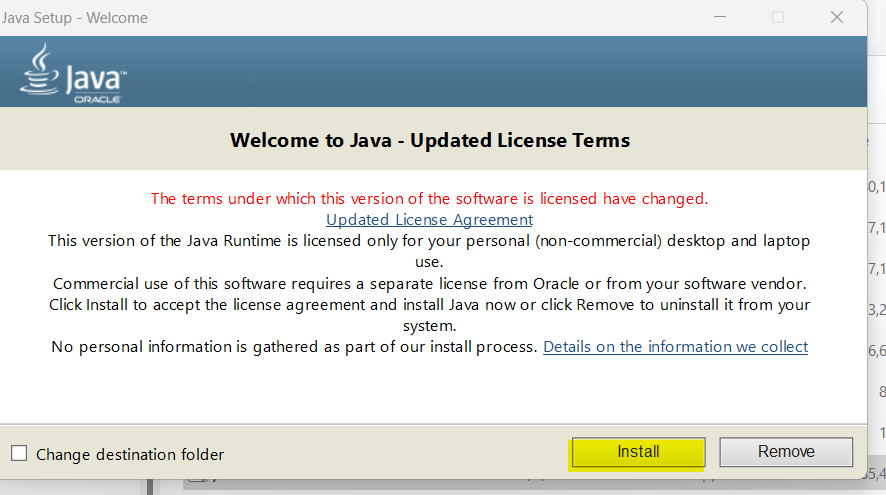
A screenshot of a computer

Description automatically generated

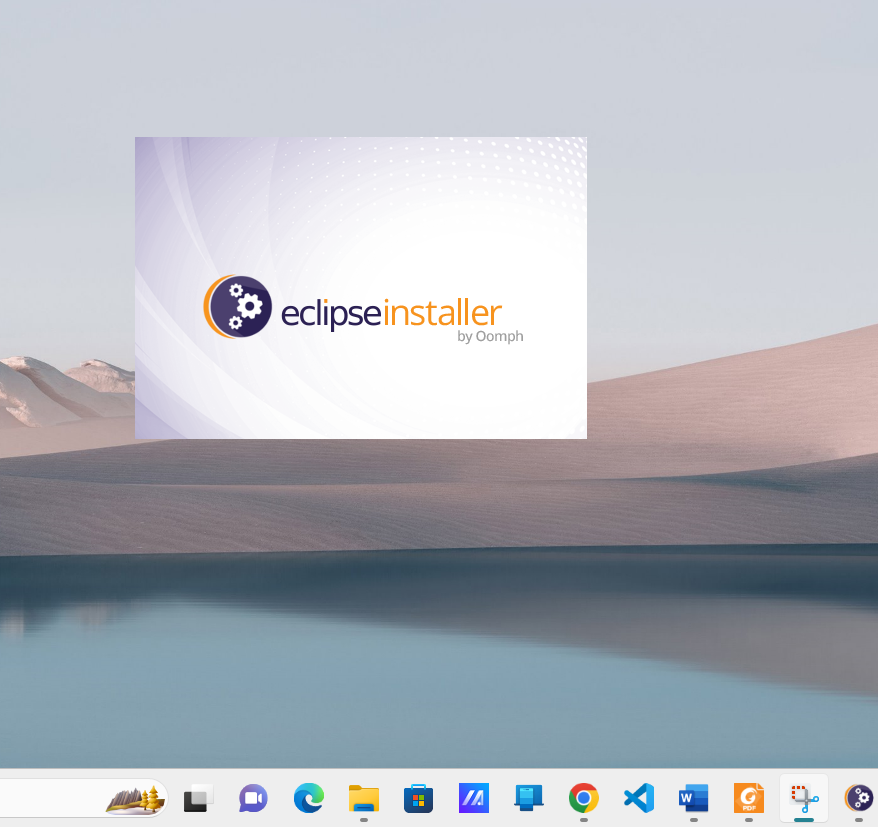
I.1.3 Téléchargement Java SE:

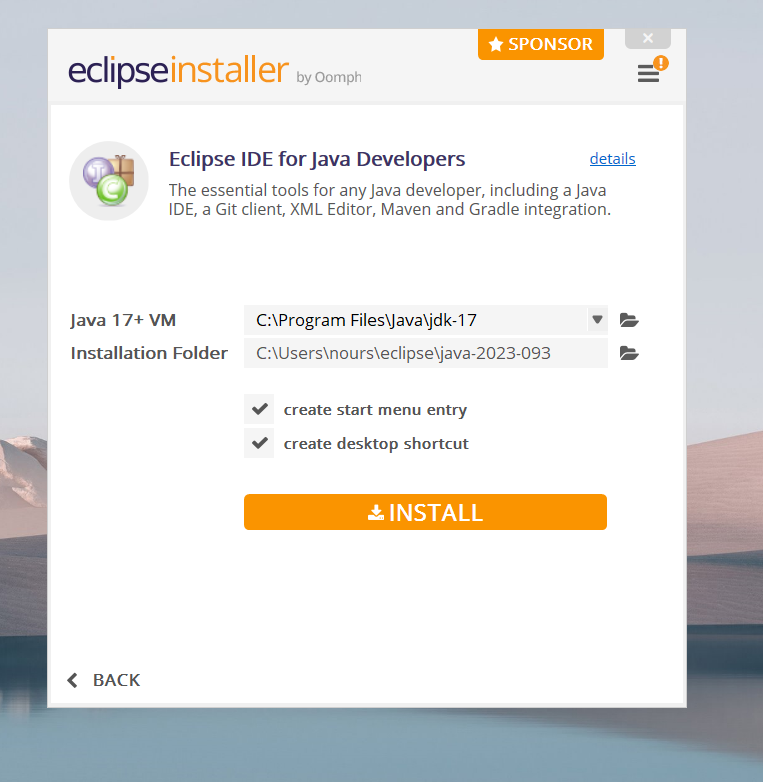
Télécharger un environnement standard édition de Java (Java SE) qui contient non seulement un **J**ava **R**untime **E**nvironnement (JRE) pour pouvoir tourner des applications java mais également un **J**ava **D**eveloppement **K**it pour pouvoir compiler vos codes sources. Nous proposons de télécharger Java SE 7u. **link :** http://www.oracle.com/





I.1.4 Téléchargement ECLIPSE IDE:

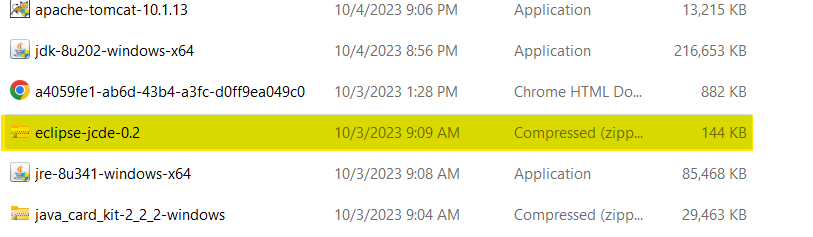
Suivant le type de votre système, télécharger un environnement de développement java pour pouvoir éditer votre code source. Nous proposons de télécharger ECLIPSE IDE. **link :** www.eclipse.org   




I.1.5 Téléchargement du plugin d'intégration Eclipse-JCDE version 0.2 :

Le plugin d'intégration Eclipse-JCDE version 0.2 : archive **eclipse-jcde-0.2.zip** à partir du site de téléchargement sourceforge :

http://sourceforge.net/projects/eclipse-jcde/



**I.2 Instructions d'installation :**

**I.2.1 Installation d’Eclipse sous windows :**

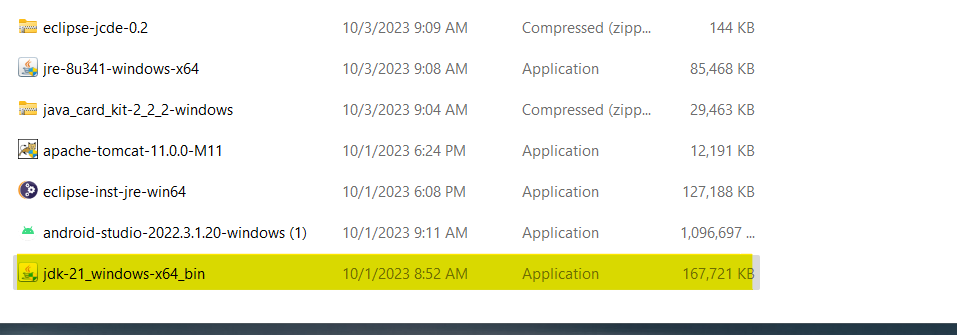
**Installation du JDK**

Remarque:

Il ne faut pas confondre le JDK (Java Development Kit) avec le JRE (Java Runtime Environment) :

✓ la JRE ne contient que les outils nécessaires pour exécuter des applications Java;

✓ le JDK permet également d'exécuter des applications Java mais il permet en plus de compiler du code source java pour en faire des applications compréhensible par la JRE, le JRE est inclut dans le JDK.



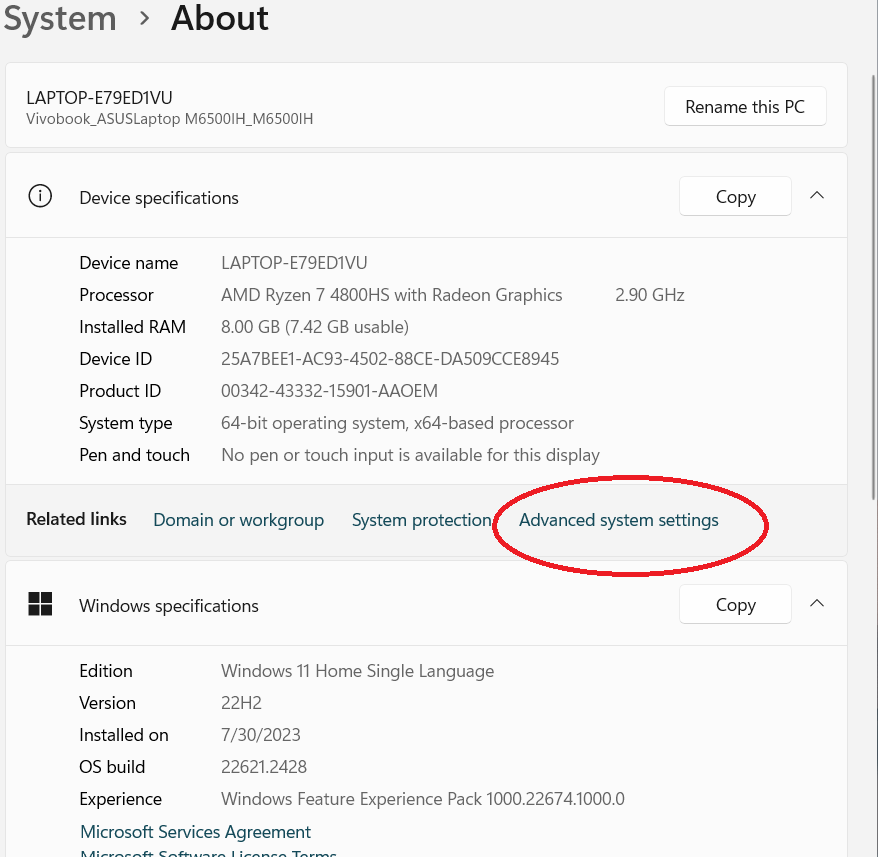
NB: software already installed

A screenshot of a computer error message

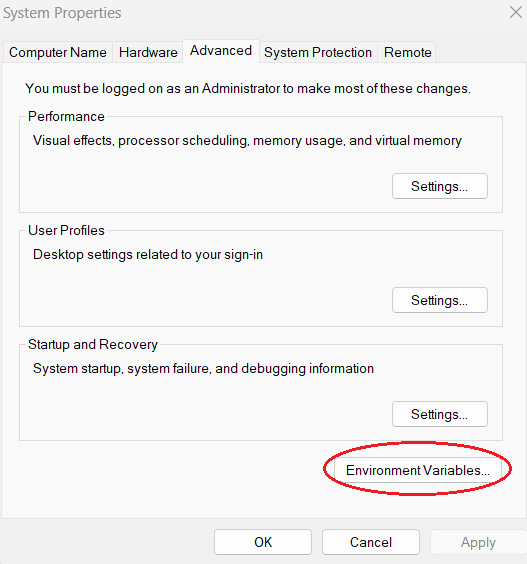
Description automatically generated

**Traitement des variables Environnement**

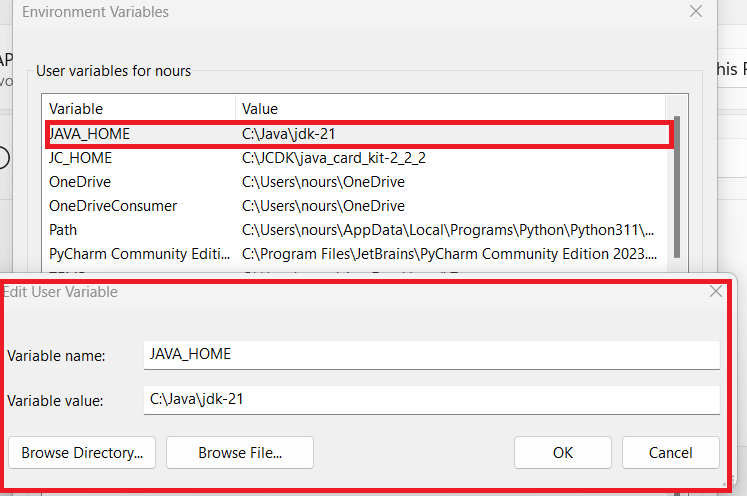
**1)**



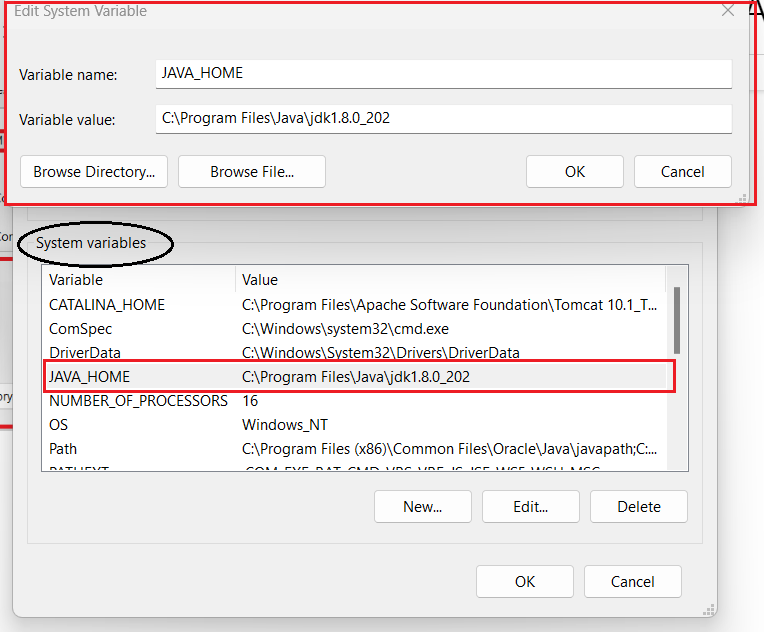
2)



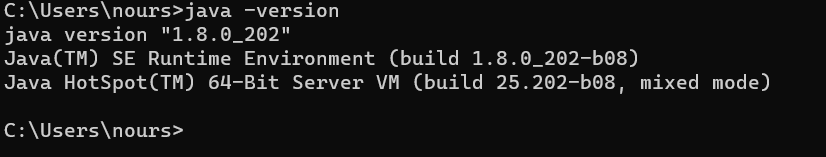
3)



4)

  
5) Pour vérifier le bon fonctionnement de votre JDK, il suffit de taper la commande « **java –**

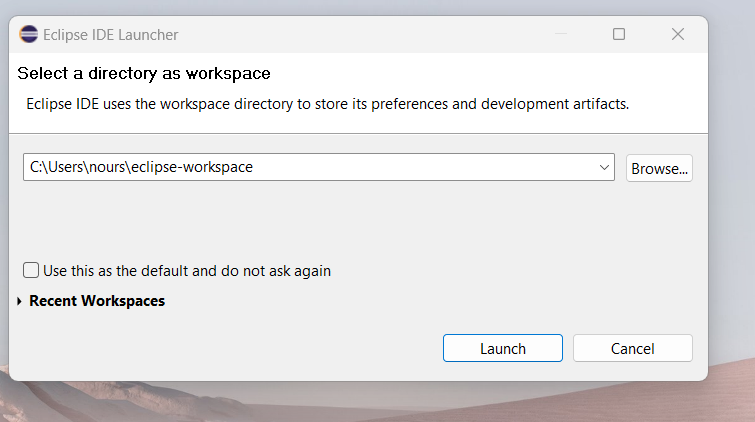
**version** » sous la console DOS :



**Installation de l’IDE - Eclipse :**



Remarque :

Eclipse vous demande de **choisir un Workspace** (Espace de travail). Notre Workspace sera par défaut le répertoire **C:\Eclipse\workspace** :   
  


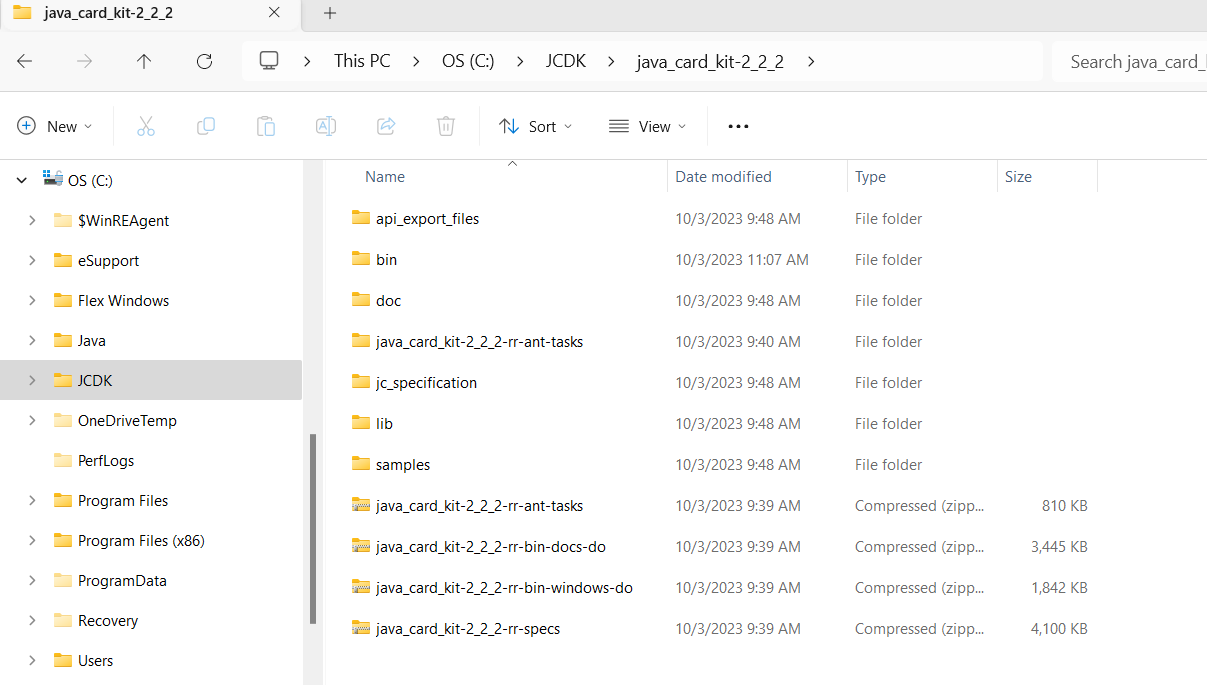
**I.2.2 Installation du Java Card Development Kit 2 .2.2 :**

Tout d'abord, Il faut créer un nouveau répertoire **C:\JCDK** :

▪ Décompresser l’archive **java\_card\_kit-2\_2\_2-windows.zip** après l’avoir télécharger dans

le répertoire **C:\JCDK**.

▪ Aller dans le répertoire C:\JCDK **\java\_card\_kit-2\_2\_2**

▪ Décompresser toutes les fichiers du répertoire courant C:\JCDK **\java\_card\_kit-2\_2\_2** :   
   


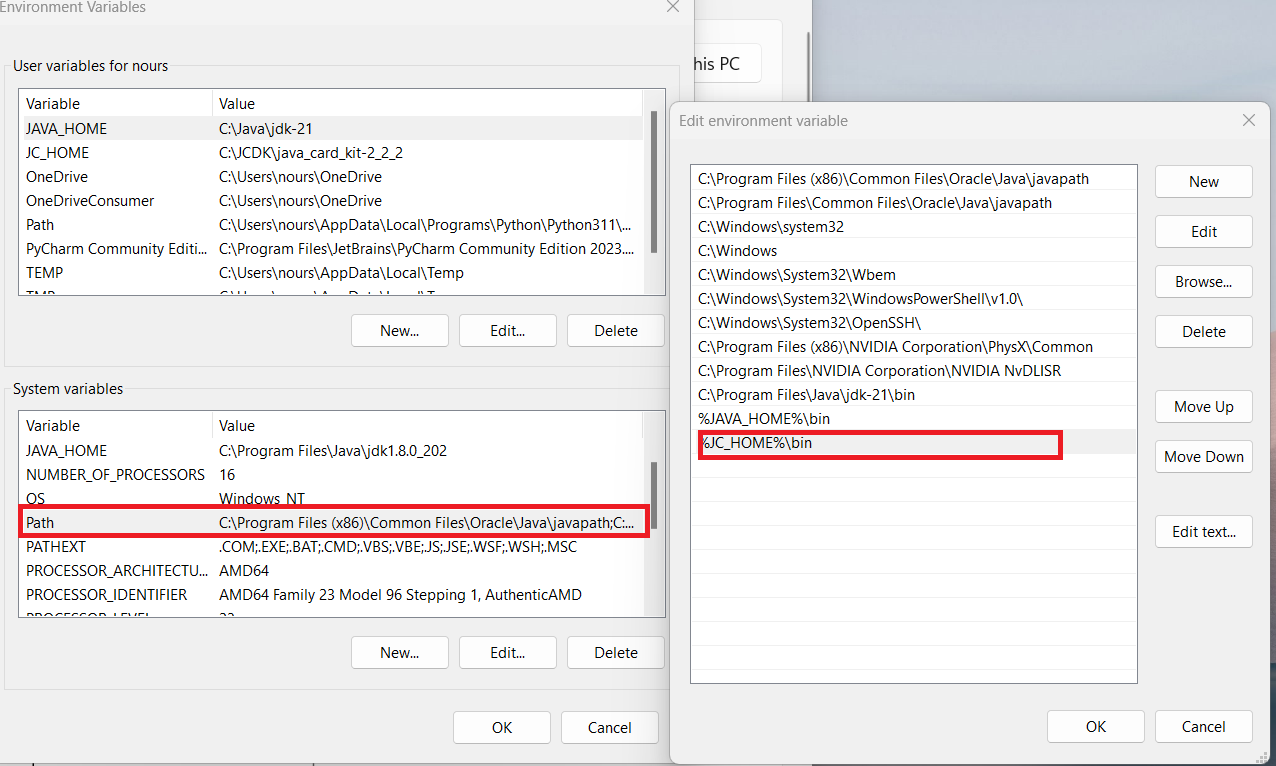
Remarque :

- **C:\JCDK\java\_card\_kit-2\_2\_2\bin** : contient les outils du JCDK (simulateur, ...)

- **C:\JCDK\java\_card\_kit-2\_2\_2\lib** : contient les librairies du JCDK

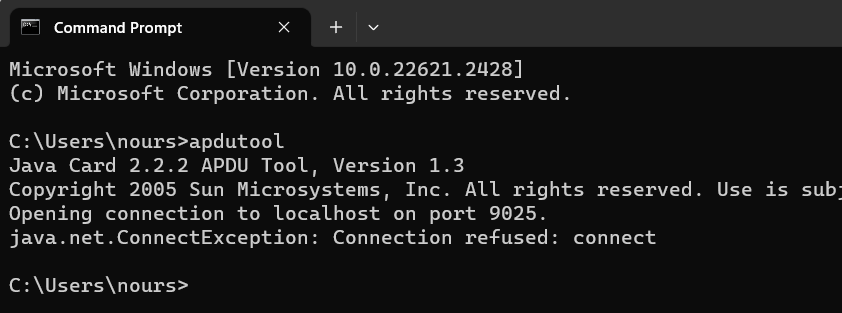
- **C:\JCDK\java\_card\_kit-2\_2\_2\jc\_specifications\specs\api\html** : contient l'API du

JCDK (très utile)   
Une fois l'installation terminée, il faut ajouter la variable d’environnement **JC\_HOME** contenant le chemin d'accès au JCDK dans la définition de la variable **Path** :



Vérifiez la réussite de configuration en tapant « **apdutool** » dans une console de

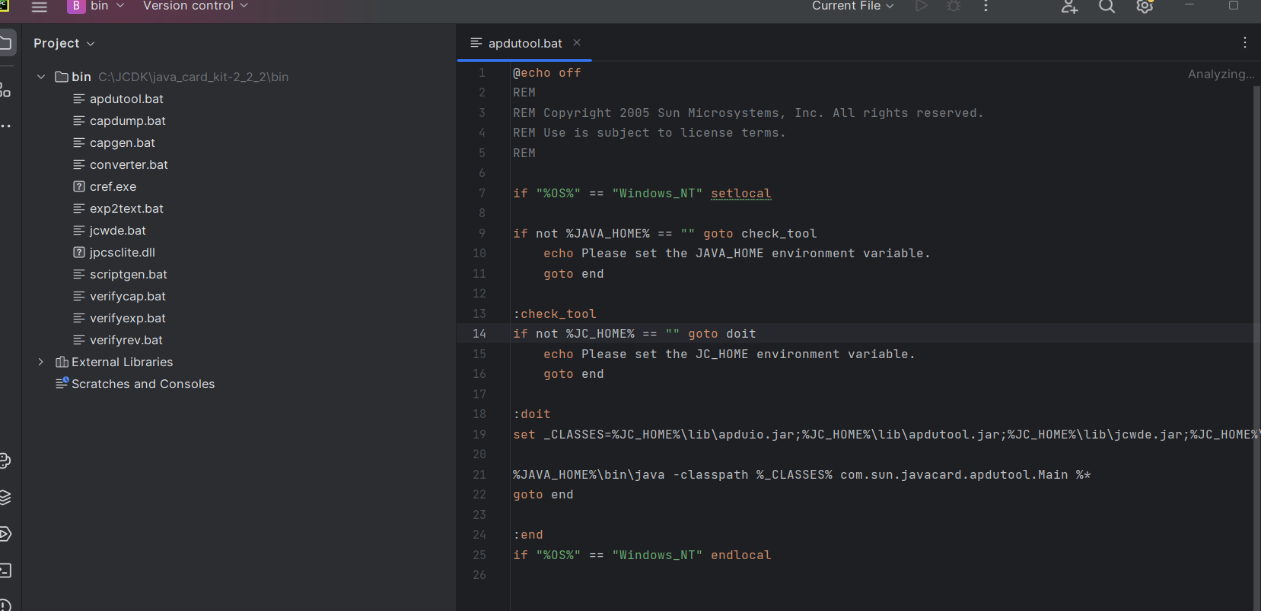
commandes :



En cas ou la commande apdutool n’a pas fonctionner

On peut réaliser ses étapes suivantes :

1. Accéder au fichier bin situé dans l’arboraisance suivante C:\JCDK\java\_card\_kit-2\_2\_2\bin
2. Ouvrir le fichier apdutool.bat en mode edit
3. Vérifier que le code est écrit comme ce dessus :



Mise a jour des plugins:

A screenshot of a computer

Description automatically generated

C’est un ensemble des méthodologies, des outils et des concepts informels imposés par les normes adaptées à l'environnement d'utilisation afin de favoriser la production et la maintenance de composants logiciels de qualité.

Le génie logiciel est généralement réservé aux logiciels complexes de grande envergure et non aux applications ou programmes simples. Le développement n'est toutefois qu'une phase du processus. Les ingénieurs logiciels sont responsables de la conception des systèmes, alors que les programmeurs sont chargés du codage permettant leur implémentation.

1. Télécharger le zip **eclipse-jcde-0.2.zip** à partir du lien suivant :

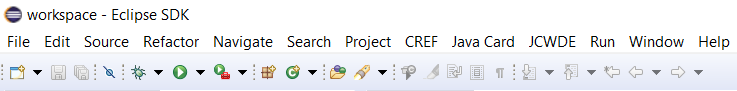
<https://sourceforge.net/projects/eclipse-jcde/files/eclipse-jcde/eclipse-jcde-0.2.zip/download>

1. Décompresser l’archive **eclipse-jcde-0.2.zip** après l’avoir téléchargé dans le répertoire **C:\Eclipse\eclipse\**

A screenshot of a computer

Description automatically generated

1. Ouvrer Eclipse, vous remarquez que les plugins sont ajoutés dans la barre de menu.



1. Maintenant allez sur **Help>Install New Software…**

A screenshot of a computer program

Description automatically generated

1. Cliquez sur **Manage** puis **JavaCard.**

A screenshot of a computer program

Description automatically generated

1. Ajouter le chemin de Java Card Home **C:\JCDK\java\_card\_kit-2\_2\_2** puis cliquer sur **Apply and Close.**

A screenshot of a computer program

Description automatically generated

1. Conclusion :

A ce stade, nous disposons d'un environnement de développement complet permettant de créer des Applet Javacard, de les simuler et de créer des applications clientes.

Chapitre II. Développement d’une

application coté Serveur

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 1 )Creation de l’applet card sous Eclipse :

1.1. Création d’un nouveau projet :

▪ Lancer Eclipse, dans le menu File, faire New puis Other et

sélectionner "*Java Card Project* ":

A screenshot of a computer program

Description automatically generated

▪ Cliquer alors sur le bouton **Next**. Donner alors un nom au projet. Nous l'appellerons "**TP Javacard**"

A screenshot of a computer

Description automatically generated

▪ Cliquer alors sur le bouton **Finish**. Vous devriez alors voir le message d'erreur

Entrer alors le dossier où a été installé le Java Card Development Kit 2.2.2, puis cliquer sur le bouton **OK :**

A screenshot of a computer program

Description automatically generated

1.2. Création d’une applet Javacard:

- Aller dans le menu **File**, faire **New** puis **Other**... Sélectionner alors "*Java Card Applet"* puis cliquer sur le bouton **Next**.

Donner alors un nom au package ainsi qu'à l'applet. Nous les appellerons respectivement "monpackage" et "MonApplet" :

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Cliquer alors sur le bouton **Finish**. Eclipse vient de générer automatiquement le squelette de notre Applet :

A screenshot of a computer program

Description automatically generated

▪ Clic droit sur le package monpackage, **Java Card Tools**, **Set Package AID** :

A box with text and numbers

Description automatically generated

II.2 Codage de notre applet :

Nous allons créer une applet **compteur**. L'applet comportera 4 fonctions :

- incrémenter le compteur

- décrémenter le compteur

- interroger le compteur

- initialiser le compteur à une valeur donnée

**Étape 1. Ajouter API JavaCard :**

A screenshot of a computer program

Description automatically generated

**Étape 2. Déclarer les attributs et les constantes :**

A screenshot of a computer code

Description automatically generated

**Étape 3. Définition des méthodes publiques qu'elle doit obligatoirement implémenter**

On définit tout d'abord :

1) **la méthode install () : création et enregistrement de l'objet Applet**

A computer screen shot of a computer code

Description automatically generated

2) **la méthode process () : Traitement des commandes APDU**

A screenshot of a computer program

Description automatically generated

**III.3 Outils de simulation :**

**3.1.JCWDE** : simulateur sans conservation d'état

On fait un clic droit sur le package, sélectionner Java Card Tools puis Generate Script. Ceci a pour conséquence de générer automatiquement les APDU nécessaires à l'upload, l'instanciation (installation) et la sélection de l'applet sur une Javacard. testing.javacard contient alors 3 scripts :

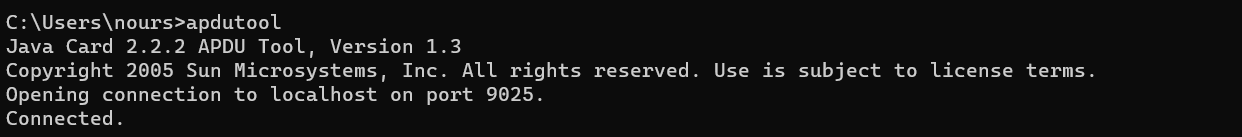
* A screenshot of a computer program

  Description automatically generatedcap-downloadscript
* create-MonAppletscript
* select-MonApplet.script

A screenshot of a computer

Description automatically generated

lancer l'outil **APDUTOOL:**



* Installons notre applet en recopiant les APDU contenus dans le script **createMonApplet.script**
* sélectionner notre applet en recopiant l'APDU contenu dans **selectMonApplet.script**

A screenshot of a computer

Description automatically generated

* tester l’applet. Commençons par **interroger le compteur (INS = 0x02)** en envoyant l'APDU:

**0xB0 0x02 0x00 0x00 0x00 0x7F**



* **incrémenter le compteur (INS = 0x00)** en envoyant l'APDU:

**0xB0 0x00 0x00 0x00 0x00 0x7F;**



* **Interrogeons de nouveau** le compteur **(INS = 0x02)** en envoyant l'APDU:

**0xB0 0x02 0x00 0x00 0x00 0x7F;**



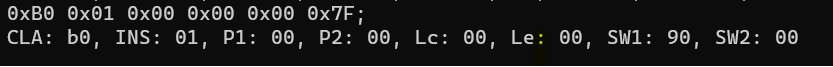
* **Initialisons** maintenant **(INS = 0x03)** le compteur à **0x4A** en envoyant l'APDU suivante

**0xB0 0x03 0x00 0x00 0x01 0x4A 0x7F;**



* **Décrémentons le compteur (INS = 0x01)** en envoyant l'APDU suivante :

**0xB0 0x01 0x00 0x00 0x00 0x7F;**



* **Interrogeons de nouveau** le compteur **(INS = 0x02)** en envoyant l'APDU suivante :



* Nous allons nous déconnecter du simulateur en tapant "**powerdown;"** dans **apdutool**, ce

qui provoque **la fermeture de JCWDE** dans la console d'Eclipse :

**powerdown;**

A screenshot of a computer

Description automatically generated

**3.2 CREF :** simulateur avec conservation d'état:

* Nous pouvons lancer **CREF** et créer notre fichier image. Pour cela, ouvrir une invite de

commandes, lancer le script **setvars.bat**, puis taper " **cref -o monapplet.eeprom** " et

valider :

A screenshot of a computer program

Description automatically generated

* **Uploadons maintenant notre applet**. Pour cela, clic droit sur le package monpackage, **Java Card Tools**, **Deploy** :

A screenshot of a computer

Description automatically generated

* **Relançons CREF** afin d'installer notre applet, en prenant soin de recharger notre fichier image (option **–i**) :

A screenshot of a computer program

Description automatically generated

* **Installons notre applet**. Pour cela, dans Eclipse, clic droit sur le **script** *createMonApplet.script,* **Java Card Tools**, **Run Script** :

A screenshot of a computer

Description automatically generated

* Puis, dans un autre terminal, **lançons apdutool**, sélectionnons notre applet, après quoi nous pouvons envoyer des APDU à notre applet :

A screenshot of a computer program

Description automatically generated

Chapitre III. Programmation d’une

application coté client

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**III.1 Création de l’application client sous Eclipse :**

III.1.1 Création d’un nouveau projet :

- Créons un **nouveau projet.** Pour cela, dans Eclipse, aller dans le menu **File,**

faire **New** puis **Project...**

- Choisir **Java Project** puis confirmer en cliquant sur **Next.** Donner un nom au nouveau projet (**Application client** par exemple) puis cliquer sur **Finish** :

A screenshot of a computer program

Description automatically generated

III.1.2 Ajout de la librairie « **apduio** » dans le classpath

Afin de pouvoir utiliser les classes servant à communiquer avec notre Javacard, il faut ajouter la bibliothèque **apduio.jar** (présente dans le répertoire **C:\JCDK \java\_card\_kit-2\_2\_2\lib**).

- Pour cela, faire un clic droit sur notre projet **Application Client** puis **Propriétés**...

- Dans la section **Java Build Path,** sélectionner l'onglet **Librairies** et cliquer sur le bouton **Add External Jars** ...

- Sélectionner alors le **fichier apduio.jar**, valider et appuyer sur le bouton **OK**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

III.1.3 Création de la classe principale :

- Tout d'abord, créons un **nouveau package**. Pour cela, faire un clic droit sur notre projet, puis **New et Package**

- Donner un **nom** au package (**monpackageclient** par exemple ), puis valider à l'aide du bouton **Finish**

A screenshot of a computer program

Description automatically generated

- Créons maintenant la classe principale de notre application. Pour cela, faire un clic droit sur le package créé puis **New** et **Class**

- Donnons un **nom** à notre nouvelle classe **(Maclasse** par exemple ), cocher la case **public static void main** puis cliquer sur **Finish** :

A screenshot of a computer program

Description automatically generated

L'application cliente se trouve sur le terminal qui communique avec le serveur (applet carte), on peut séparer l’écriture de notre application en plusieurs étapes :

**Étape 1 - Connexion :**

La connexion au simulateur se fait via une socket**.** Le simulateur écoute par défaut sur le **port 9025**. La classe que nous utiliserons pour les échanges de données est **CadT1Client**.

A screen shot of a computer code

Description automatically generated

**Etape 2 - Sélection**

La **sélection d'applet** se fait en envoyant la commade SELECT APDU (voir annexe A)

A computer screen shot of a computer code

Description automatically generated

**Etape 3 - Invocation des services implémentés**

Il suffit pour chaque opération **d'initialiser** correctement **une instance de l'objet APDU et de l'envoyer à la carte** via l'instance de la classe **CadT1Client**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**Etape 4 - Mise hors tension**

La déconnexion de la Javacard se fera via la méthode **powerDown()** de la classe **CadT1Client** :

**A screen shot of a computer

Description automatically generated**

**III.2 Utilisation de l'application cliente avec un simulateur - JCWDE**

Afin de pouvoir **lancer le simulateur** de notre applet en ligne de commande, nous allons **créer un fichier "de configuration**". Il permet de lister les applets Javacard à installer pour la simulation et de spécifier leurs AID respectifs.

▪ Créons notre fichier que nous appellerons **"monapplet.app"** (répertoire parent du package contenant le fichier class de l'applet card « .\workspace\TP Javacard\bin ») **.** Il contiendra la ligne suivante :

**monpackage.MonApplet 0x01:0x02:0x03:0x04:0x05:0x06:0x07:0x08:0x09:0x00:0x00**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

A black screen with white text

Description automatically generated

Maintenant que notre simulateur est lancé, **lançons notre application cliente** :

Commençons par **interroger le compteur** : tapons **1** puis validons :

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

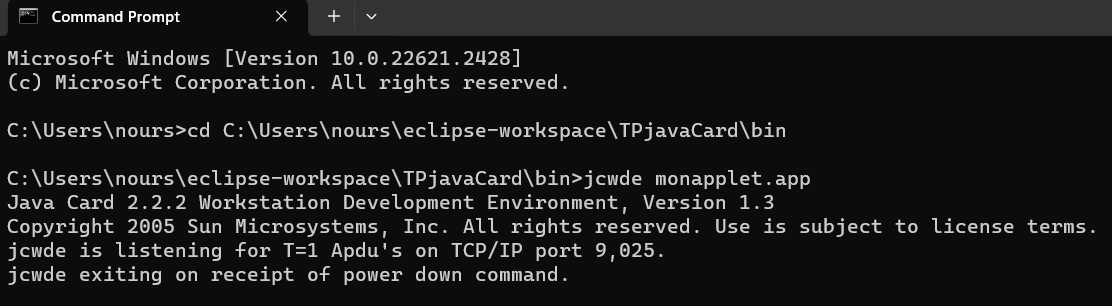
A screenshot of a computer

Description automatically generated

**Quittons maintenant notre application cliente** (commande **5**). Nous pouvons voir que le **simulateur se termine automatiquement** à la réception de la commande "**powerdown"** :

A screenshot of a computer

Description automatically generated



Chapitre IV. Réalisation du Mini-Projet

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IV.1 Interface Graphique ( Partie Client ) :

Pour réaliser l’interface graphique du Partie Client, on va utiliser Java Swing.

Il existe deux méthodes pour la réalisation du design de cette interface :

* Code pure
* En utilisant la palette (Drag and Drop)
* Dans ce projet on va construire l’interface en utilisant la palette. Pour cela on va installer le plugin **WindowBuilder** dans eclipse.

IV.1.1 WindowBuilder Introduction :

Le plugin Eclipse WindowBuilder est un concepteur Java GUI visuel, puissant et facile à utiliser permettant la création d'applications GUI Java sans vous casser la tête à écrire du code pour afficher des objets graphiques simples comme fenêtres, bouton de commandes, champs de textes... Avec WindowBuilder, vous pouvez créer des fenêtres compliquées en quelques minutes, il suffit d’utiliser le concepteur visuel et le code Java sera automatiquement généré pour vous. Vous pouvez facilement ajouter des contrôles à l'aide de glisser-déposer, gérer les événements de vos contrôles, modifier diverses propriétés des contrôles à l'aide d'un éditeur de propriétés et bien plus encore. Le code généré ne nécessite aucune bibliothèque personnalisée supplémentaire pour compiler et exécuter : l'ensemble du code généré peut être utilisé sans installer WindowBuilder.

L'éditeur est doté des principaux composants interface utilisateur (user interface ) suivants:  
**•    Design View** - la zone de présentation visuelle principale.  
•    **Source View**- code d'écriture et analyse du code généré  
•    **Structure View**- composée de l'arbre de composant et du volet Propriété.  
•    **Component Tree**- montre la relation hiérarchique entre tous les composants.  
•    **Property Pane**- affiche les propriétés et les événements des composants sélectionnés.  
•    **Palette** - permet un accès rapide aux composants spécifiques à une trousse d'outils.  
•     **ToolBar** - permet d'accéder aux commandes couramment utilisées.  
•     **Context** Menu - permet d'accéder aux commandes couramment utilisées.

IV.1.3 Installation de WindowBuilder :

Le plugin Eclipse WindowBuilder est un concepteur Java GUI visuel, puissant et facile à utiliser permettant la création d'applications GUI Java sans vous casser la tête à écrire du code pour afficher des objets graphiques simples comme fenêtres, bouton de commandes, champs de textes... Avec WindowBuilder, vous pouvez créer des fenêtres compliquées en quelques minutes, il suffit d’utiliser le concepteur.

1. Lancer Eclipse puis Depuis le menu Help d’Eclipse, choisissez **Help > Install new Software...**

A screenshot of a computer

Description automatically generated

1. Dans la première fenêtre qui apparait cliquez sur **le bouton Add** en suite, dans la boîte de dialogue apparue, dans le champ Name, saisissez un nom descriptif (comme "**WindowBuilder**") et collez le lien ci-dessus dans le champ Emplacement.  Cliquez ensuite sur le bouton **OK** :

**Lien** : <https://download.eclipse.org/windowbuilder/latest/>

A screenshot of a computer

Description automatically generated

1

1. Sélectionnez toutes les **cases à cocher** qui vont apparaître, puis cliquez sur **Suivant** pour installer **WindowBuilder** :

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

IV.1.3 Comment utiliser WindowBuilder :

Le plugin Eclipse WindowBuilder est un concepteur

Pour utiliser WindowBuilder :

1. Créer un nouveau projet. Cliquez sur **File > New > Java Prosject**

A screenshot of a computer program

Description automatically generated

2.Saisir le nom de votre Projet (Ici **Interface\_graphique**) puis cliquer sur **Finish**

A screenshot of a computer program

Description automatically generated

3. Après avoir créer votre Projet Java. Aller sur le dossier clique droit **New**> **Other**

A screenshot of a computer program

Description automatically generated

4.Choisir ensuite comme type de projet : **WindowBuilder** -> **Application Window** puis cliquer sur **Next**

A screenshot of a computer program

Description automatically generated

5. Choisir alors le Nom de votre Projet et le package (Ici Name : **Interface\_graphique** Package **Interface\_graphique**).

A screenshot of a computer application

Description automatically generated

6. Après avoir cliquez sur le bouton  **Finish** vous obtenez votre **projet WindowBuilder** :

A screen shot of a computer program

Description automatically generated

7. Cliquez maintenant sur l'onglet Design pour afficher la fenêtre crée automatiquement par **WindowBuilder** , ainsi que les autres outils graphiques WYSIWYG : Containers, Layouts, Compoasants Swing et Awt ...

A screenshot of a computer

Description automatically generated

**Les outils graphiques de windowBuilderWindowBuilder**

**phiques de WindowBuilderraphiques de WindowBuilder**

* Maintenant, vous pouvez commencer à construire votre fenêtre et le code sera automatiquement générer dans la partie **code.**

IV.1.4 Les interfaces de notre application Client :

1. La première interface qui s’affiche lors de l’exécution. Il faut saisir le code pin dans la zone du Texte puis valider. Un apdu sera envoyer au Serveur pour confirmer si le code pin est correct ou non.

A screenshot of a computer

Description automatically generated

* 1er Cas:

si l’utilisateur saisie un input de longueur **n’est pas égale à 4** un message d’erreur sera affiché: Wrong length code!

A screenshot of a computer

Description automatically generated ou A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

* 2eme Cas:

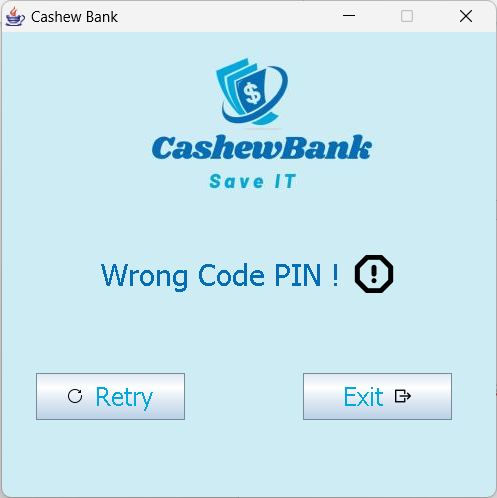
Si l’input contient **un caractére alphabétique ou plus** un message d’erreur s’affiche : Wrong Input

A screenshot of a computer

Description automatically generated

* **3éme cas:**

Si le code Pin saisi est **incorrect** alors une fenêtre d’erreur s’affiche: Wrong Code Pin



* Vous pouvez cliquer sur **Retry** pour ressaisir votre code PIN ou bien **Exit** pour fermer le programme.
* **4éme Cas :**

Si vous avez saisi un code PIN incorrect 3 fois, votre carte sera bloquée:

A screenshot of a computer

Description automatically generated

* **5éme Cas :**

Si vous avez saisi le code PIN **correctemen**t :

A screenshot of a computer

Description automatically generated

2. On se retrouve alors dans la fenêtre principale qui contient 4 boutons :

* **Check Balance** : Permet de consulter le solde .
* **Retrieve Money** : Permet de débiter un montant .
* **Cash deposit :** Permet de créditer un montant .
* **A propos** : Contient nos informations

**Check Balance** :

A screenshot of a computer

Description automatically generated

**Retrieve Money**

A screenshot of a computer

Description automatically generated

**1ére Cas :**

Si le montant qui vous avez choisi est valide et votre solde est suffisant pour le retrait d’argent.

A screenshot of a computer

Description automatically generated

**2ére Cas :**

Si vous n’avez pas le solde suffisant pour le retrait de l’argent.

A screenshot of a computer

Description automatically generated

**Cash deposit**

A screenshot of a computer

Description automatically generated

**1ére Cas :**

Si le montant qui vous avez choisi est valide et votre balance ne dépasse pas le montant autorisé (le plafond du compte :10000):

A screenshot of a computer

Description automatically generated

**2ére Cas :**

Si en ajoutant le montant que vous avez choisi votre balance dépasse le montant autorisé une erreur s’affiche :

A screenshot of a computer

Description automatically generated

**A propos**

A screenshot of a computer

Description automatically generated

Pour plus d’information cliquer sur les boutons contact :

A screenshot of a computer

Description automatically generated A screenshot of a phone

Description automatically generated

IV.1.5 Implémentation des méthodes nécessaires :

1.Déclaration des variables :

A black background with white text

Description automatically generated

2.Implémentation de la methode Msg() :

Cette méthode permet l’envoi des Apdus.

A screen shot of a computer

Description automatically generated

3.Implémentation de la methode Connect() :

Cette méthode permet la connexion avec la carte.

A screen shot of a computer code

Description automatically generated

**4.Implémentation de la methode Select() :**

Cette méthode permet la sélection de notre applet dans la carte.

A screen shot of a computer

Description automatically generated

**5. Implémentation de la methode Deselect() :**

A black screen with blue and green text

Description automatically generated

IV.2.Partie Serveur (Back-End) :

1.Declaration des variables et des constantes :

A screenshot of a computer program

Description automatically generated

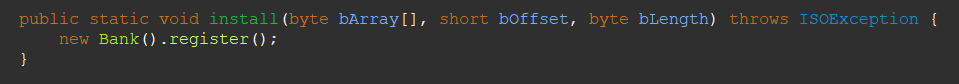
2.Implementations des méthodes nécessaires :

2.1 constructeur :

A computer code on a black background

Description automatically generated

2.2 Implementation de la methode install() :



2.3 Implementation de la methode show\_balance() :

A screen shot of a computer code

Description automatically generated

2.4 Implementation de la methode check() :

A screen shot of a computer

Description automatically generated

2.5 rédéfintion de la methode process() :

A screen shot of a computer program

Description automatically generated

2.6 Implementation de la methode add() :

A computer screen with text

Description automatically generated

2.7 implementation de la methode subtract() :

A screen shot of a computer program

Description automatically generated

**2.8 Rédéfintion de la methode PIN\_CODE() :**

A computer screen shot of a program code

Description automatically generated