



# Rapport JavaCard

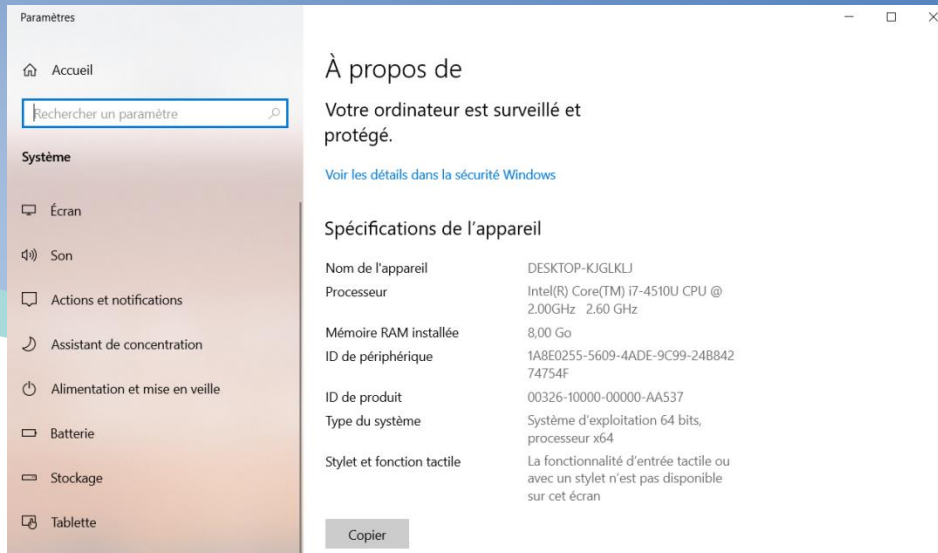
Préparé par :  
**Aya Zid & Nadine Mlayeh**

Encadré par :  
Mr. Ramzi Mahmoudi

# Environnement du développement Java Card

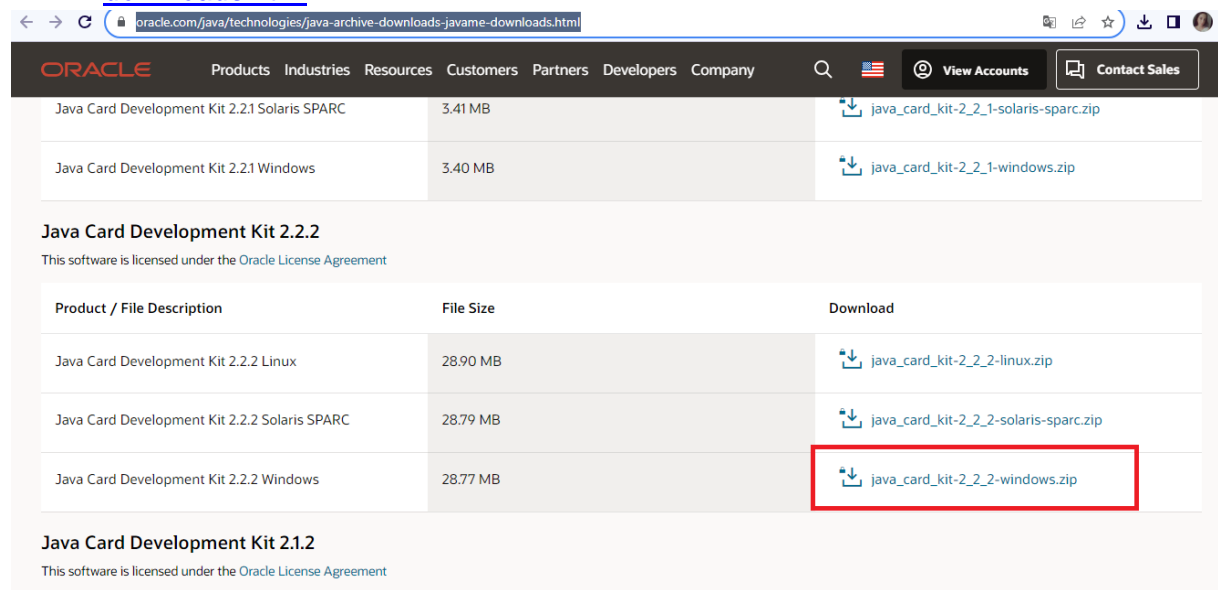
## I.1 Outils logiciels nécessaires :

1. Identifiez votre système d'exploitation ainsi que son type.















2. Le Java CardDevelopment Kit 2.2.2 : **Installer java carddevelopment kit 2.2.2 à partir de ce lien :**

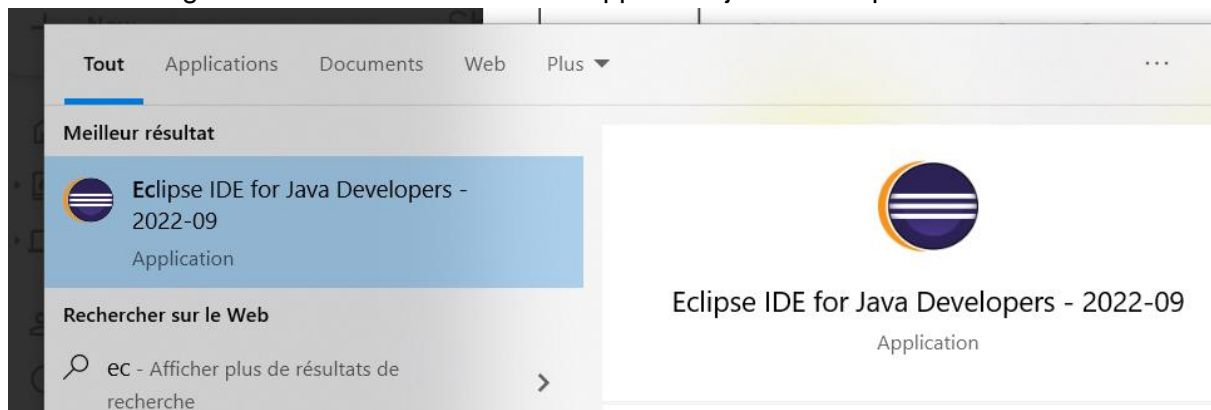
<https://www.oracle.com/java/technologies/java-archive-downloads-javame-downloads.html>



**Q3.** Télécharger un environnement standard édition de Java (Java SE) (JRE) :

Name	Owner	Last modified
 eclipse-inst-jre-win64.exe	 WORK (Ramz...	22 Nov 2022
 eclipse-jcde-0.2.zip	 WORK (Ramz...	22 Nov 2022
 java_card_kit-2_2_1-win-dom.zip	 WORK (Ramz...	22 Nov 2022
 java_card_kit-2_2_2-windows.zip	 WORK (Ramz...	22 Nov 2022
 jdk-18_windows-x64_bin.exe	 WORK (Ramz...	22 Nov 2022
 jre-8u341-windows-x64.exe	 WORK (Ramz...	22 Nov 2022

**Q4.** Télécharger un environnement de développement java : « eclipse IDE » :



## I.2 Instructions d'installation :

### I.2.1 Installation d'Eclipse sous Windows :

#### a. Installation du JDK

Installer Java Développement Kit à partir de ce lien :

<https://www.oracle.com/java/technologies/downloads/#jdk17-windows>

Oracle.com/java/technologies/downloads/#jdk17-windows

Java downloads Tools and resources Java archive

Learn about Java SE Subscription

JDK 21 **JDK 17** GraalVM for JDK 21 GraalVM for JDK 17

**JDK Development Kit 17.0.9 downloads**

JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions (NFTC).

JDK 17 will receive updates under the NFTC, until September 2024. Subsequent JDK 17 updates will be licensed under the Java SE OTN License (OTN) and production use beyond the limited free grants of the OTN license will require a fee.

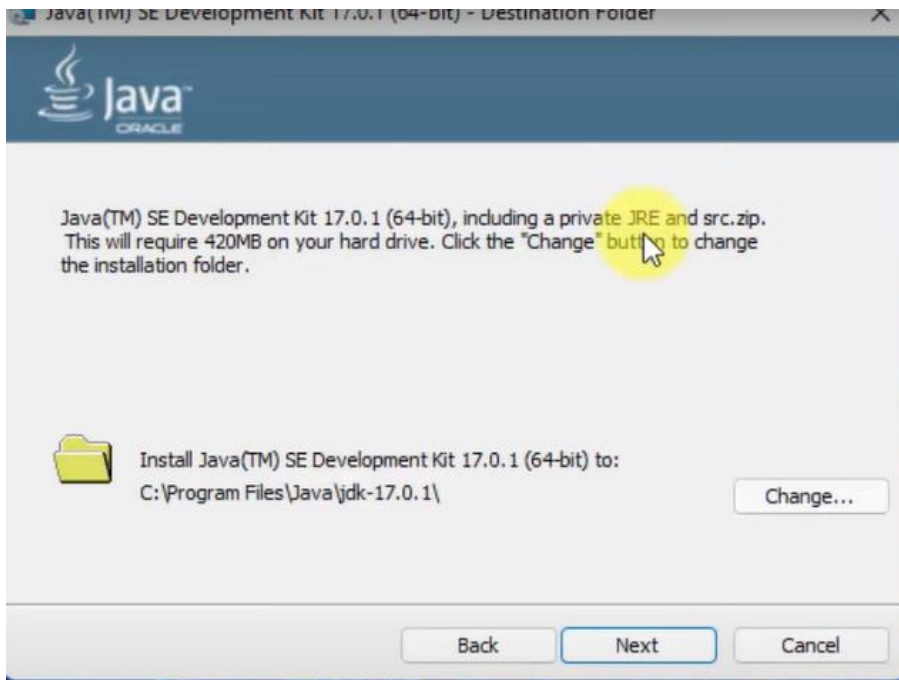
Linux macOS **Windows**

Product/file description	File size	Download
x64 Compressed Archive	172.42 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip">https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip</a> ( sha256)
x64 Installer	153.51 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe">https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe</a> ( sha256)
x64 MSI Installer	152.30 MB	<a href="https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi">https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi</a> ( sha256)

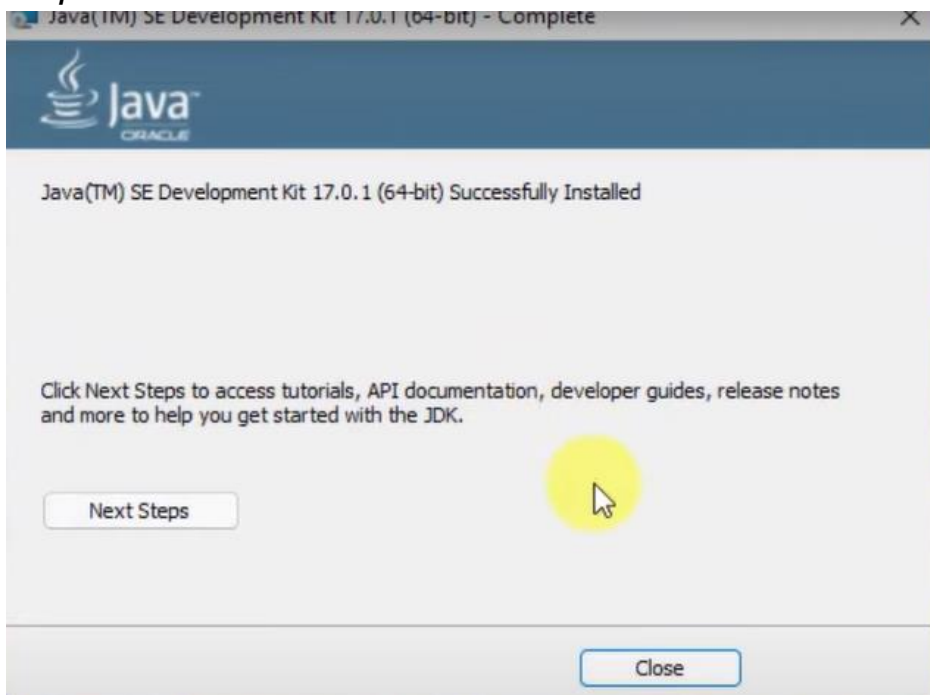
Une fois le téléchargement terminé , cliquer sur le fichier « jdk-17\_windows-x64\_bin » dans le répertoire Téléchargements pour terminer l'installation

Cliquer sur Next :

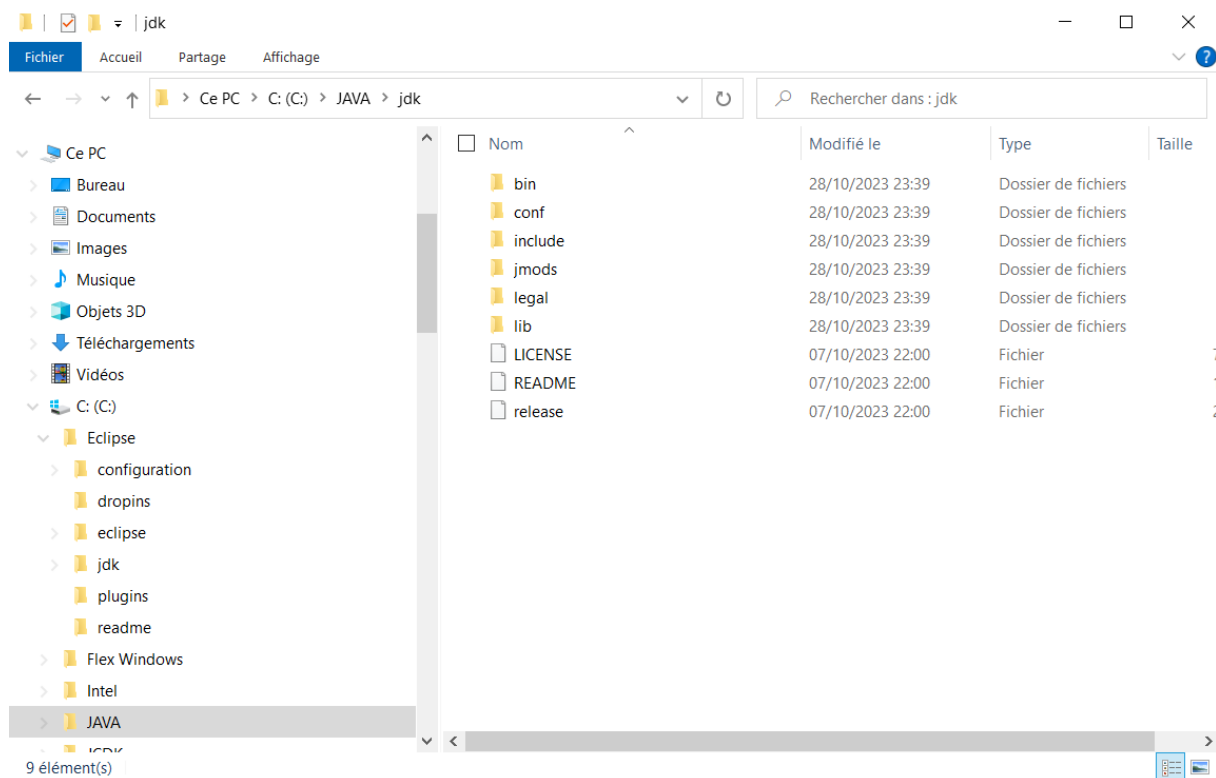




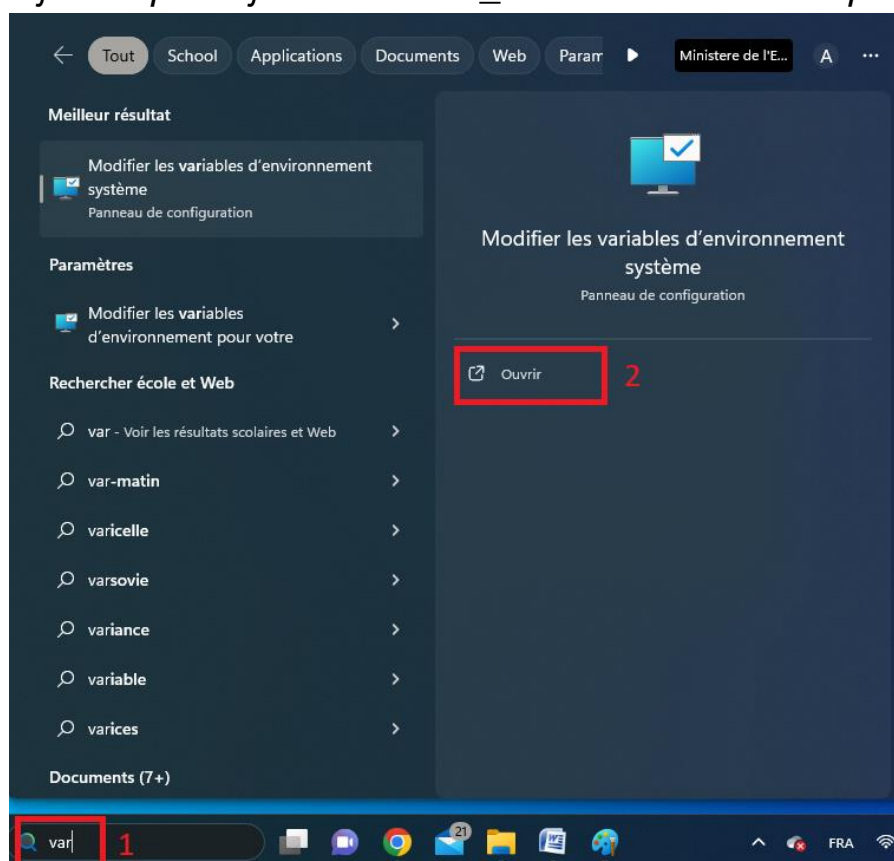
*Cliquez sur close :*

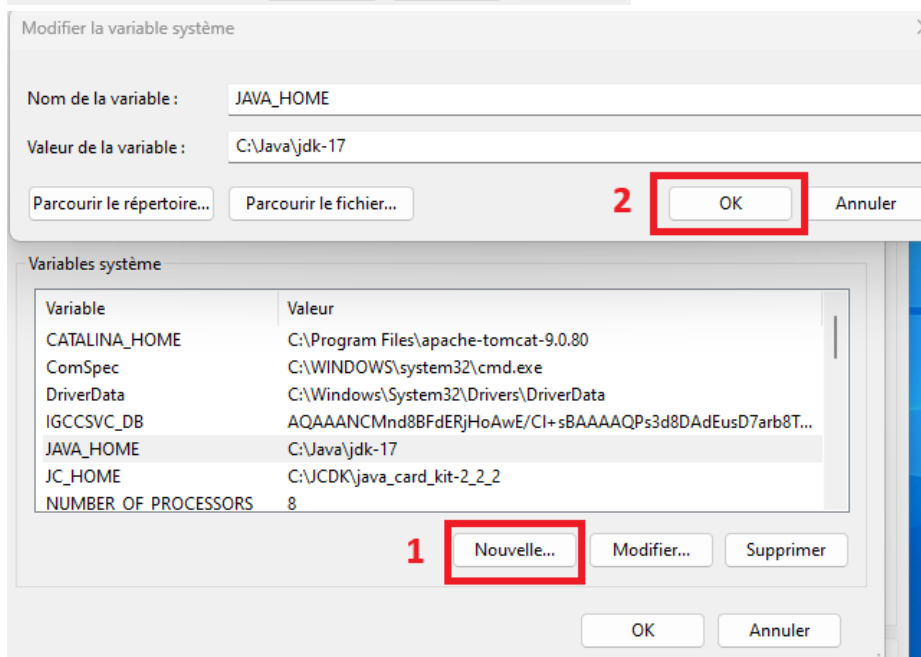
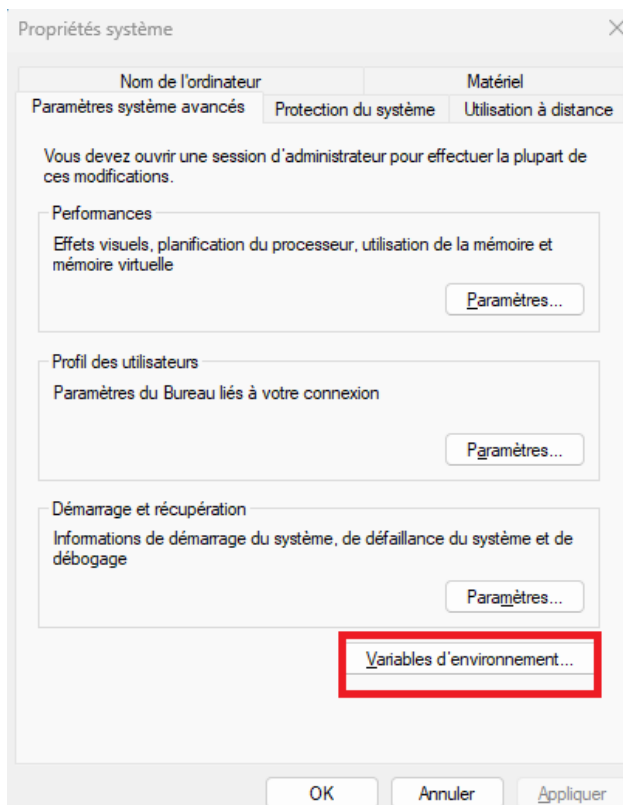


- Créer un nouveau répertoire nommé Java sous C:\ et recopier le contenu du répertoire C:\Program Files Java sous le répertoire C:\Java



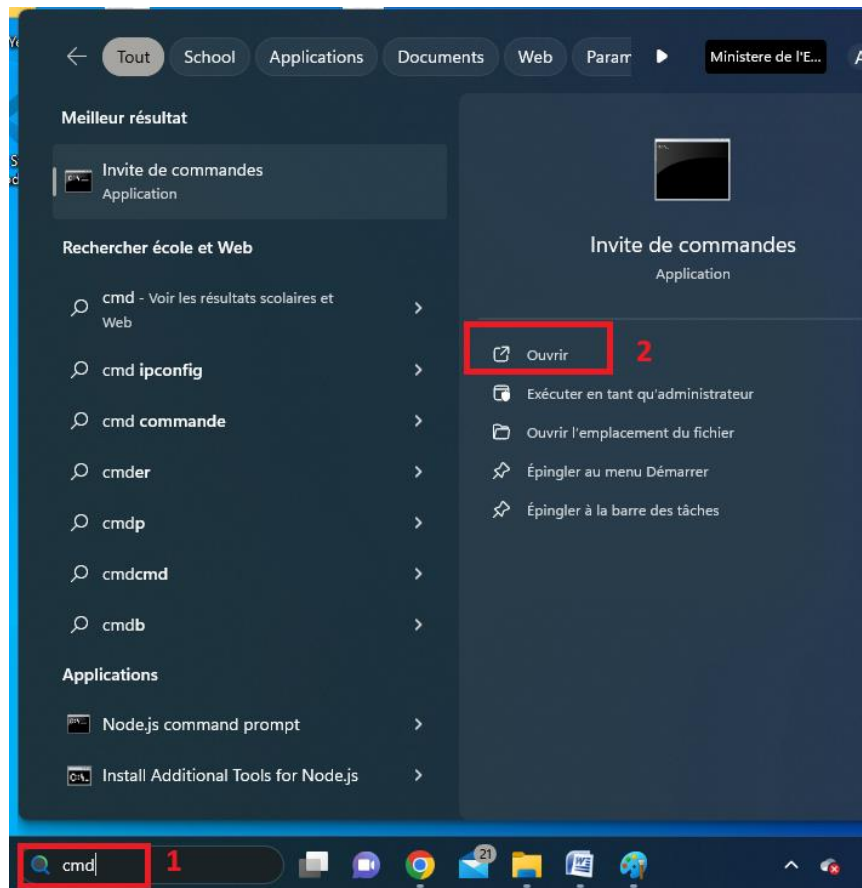
- Créer la variable d'environnement `JAVA_HOME` contenant le chemin `c:\Java\jdk-17` puis ajouter `%JAVA_HOME%\bin` dans le path :











```
Microsoft Windows [version 10.0.22621.2715]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HP>java -version
java version "17.0.8" 2023-07-18 LTS
Java(TM) SE Runtime Environment (build 17.0.8+9-LTS-211)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.8+9-LTS-211, mixed mode, sharing)

C:\Users\HP>
```

## b. Installation de l'IDE - Eclipse :

*Installer Eclipse à partir de ce lien :*

<https://www.eclipse.org/downloads/>

→ G eclipse.org/downloads/

Introducing Eclipse ThreadX

Eclipse ThreadX will be the world's first open source real-time operating system with multiple safety and security certifications governed under the stewardship of a vendor-neutral foundation.

[Read Announcement](#)

[Get Involved](#)

Install your favorite desktop IDE packages

[Learn More](#) [Download x86\\_64](#)

[Download Packages](#) | [Need Help](#)

TEMURIN

The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem.

[Learn More](#) [Download](#)

**ECLIPSE**  
FOUNDATION

[Projects](#) [Supporters](#) [Collaborations](#) [Resources](#) [Th](#)

Home ▶ Downloads ▶ Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

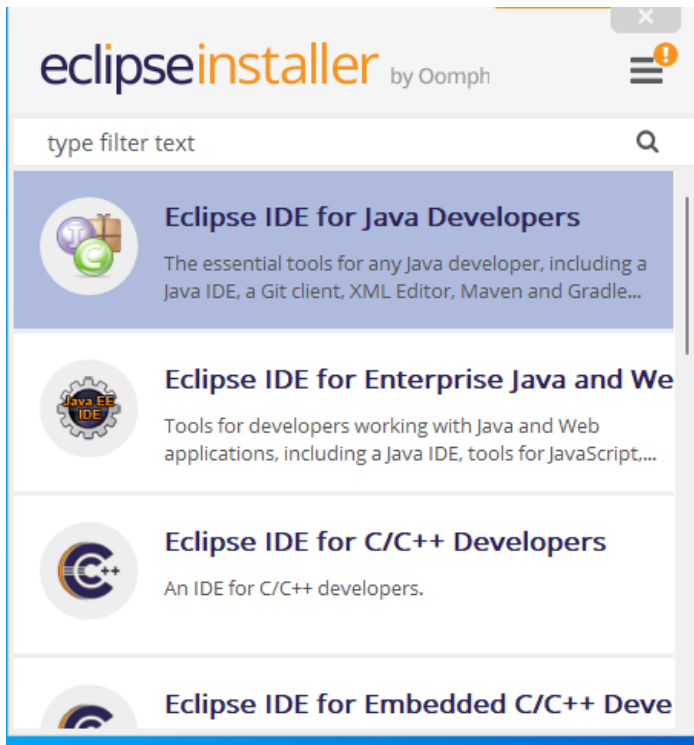
2 [Download](#)

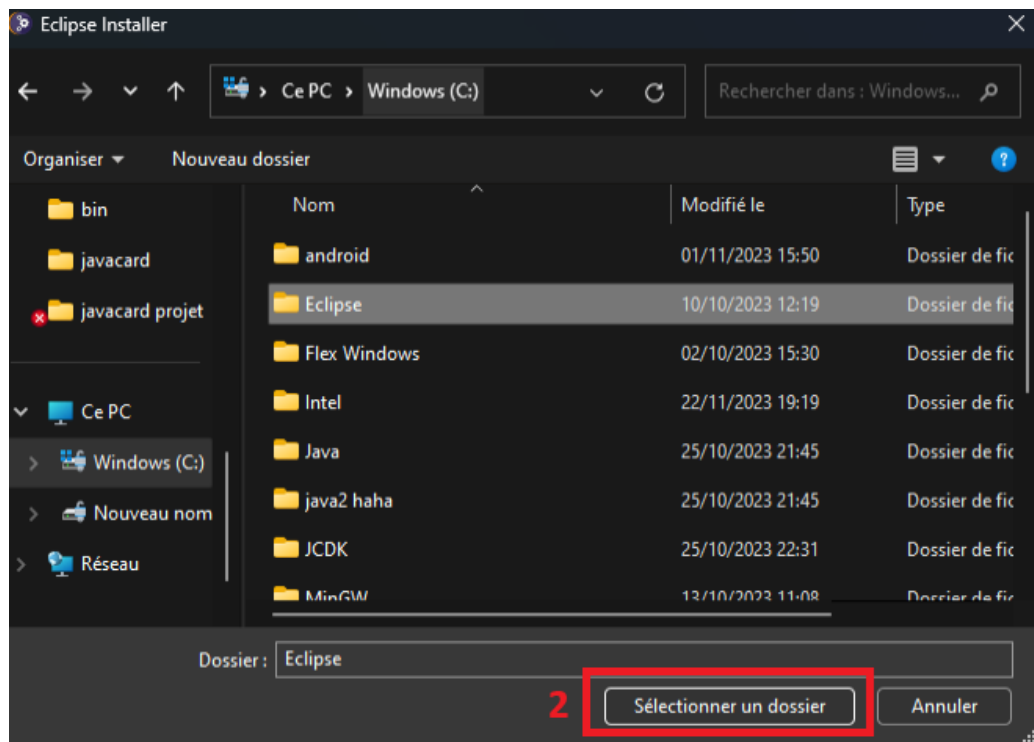
Download from: South Africa - Openserve (https)

File: [eclipse-inst-jre-win64.exe](#) [SHA-512](#)

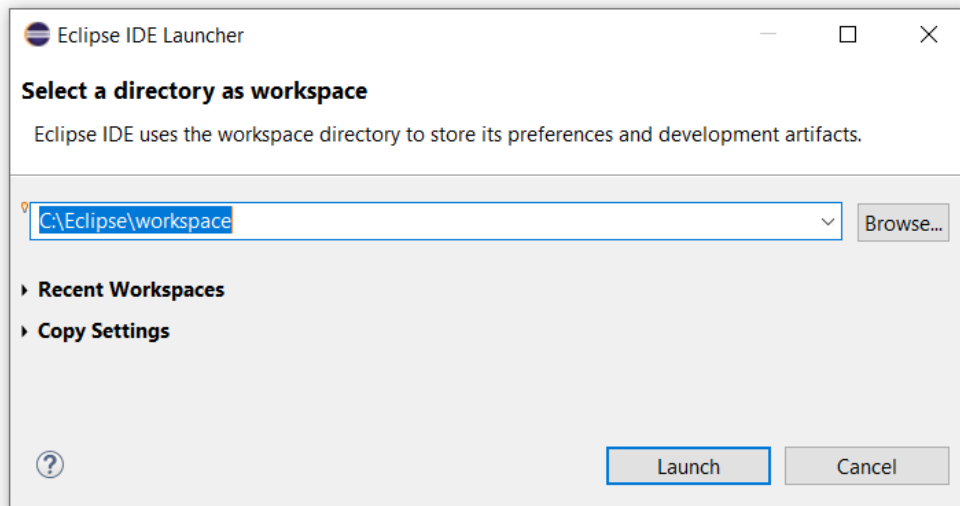
[>> Select Another Mirror](#)

- Créer un dossier nommé Eclipse sous le répertoire C:\
- Puis cliquer sur le fichier « eclipse-inst-jre-win64 » dans le répertoire Téléchargements
- Choisir Eclipse IDE for Java Developers





Le workspace est par défaut le répertoire C:\Eclipse\workspace :



### c. L'indispensable « Hello Word »

Création d'une application Java avec l'IDE Eclipse qui affiche à l'écran "Hello, World!":

## 1. Création d'un nouveau projet sous Eclipse:

New Java Project

**Create a Java Project**

Create a Java project in the workspace or in an external location.

Project name:

☒ Use default location

Location:  [Browse...](#)

JRE

☒ Use an execution environment JRE:  [Configure JREs...](#)

☐ Use a project specific JRE:

☐ Use default JRE 'jre' and workspace compiler preferences

Project layout

☐ Use project folder as root for sources and class files

☒ Create separate folders for sources and class files [Configure default...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets:  [Select...](#)

Module

☒ Create module-info.java file

Module name:

☒ Generate comments

module name will be "hello" (if no module is specified, then project name will be used as module name)

[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

## 2. Création d'une nouvelle classe :

- New -> Class :
- On indique le nom de la classe, Hello, en cochant la case Indiquant que l'on souhaite qu'elle contienne une méthode main :

New Java Class

**Java Class**

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

---

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static  
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass:

Interfaces:

Which method stubs would you like to create?

☒ public static void main(String[] args)

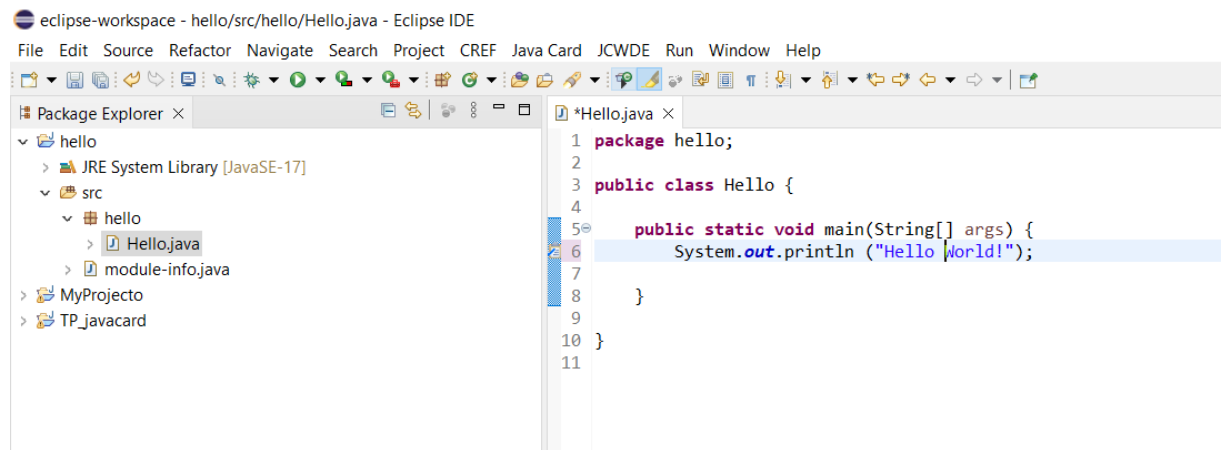
☐ Constructors from superclass

☒ Inherited abstract methods

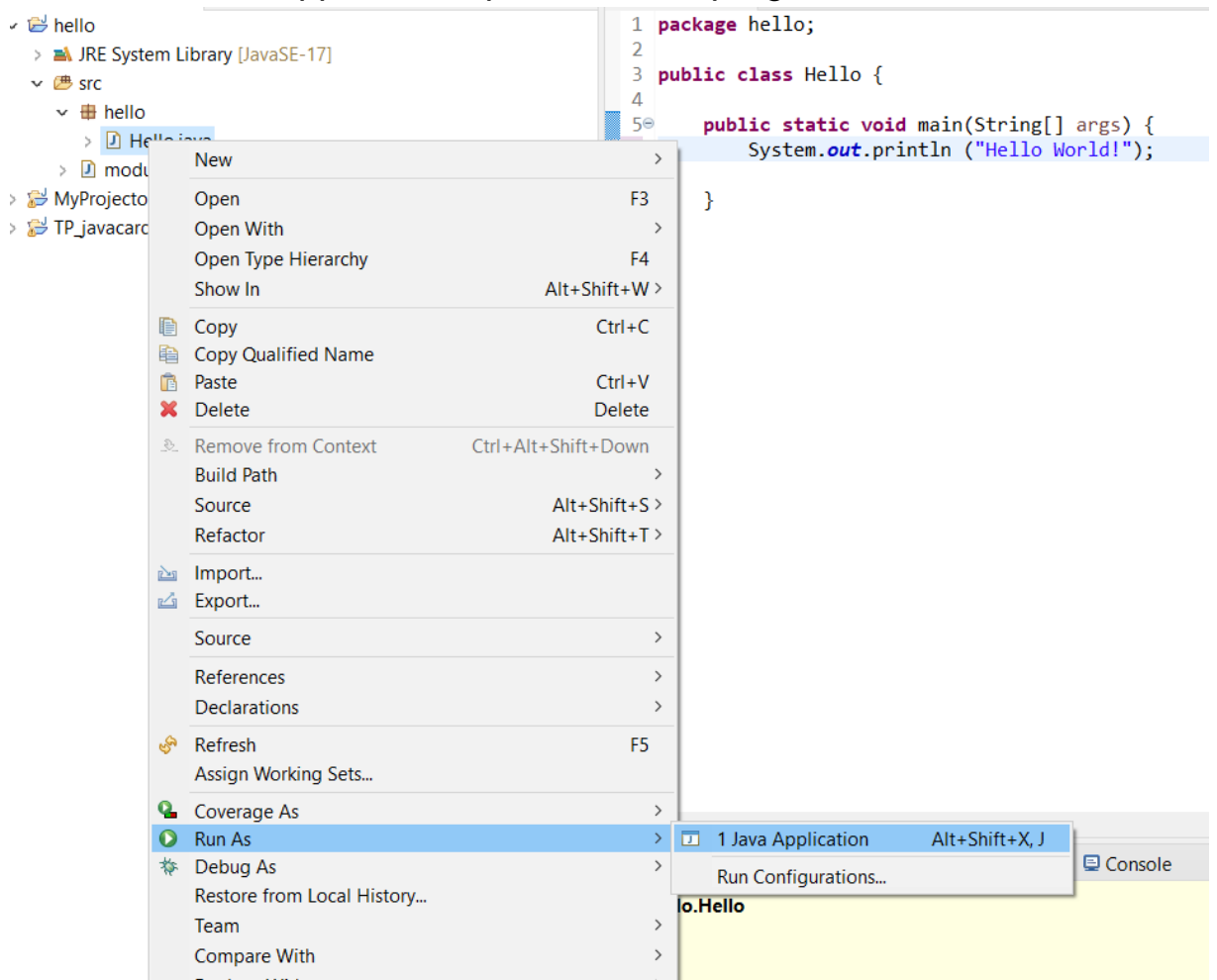
Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

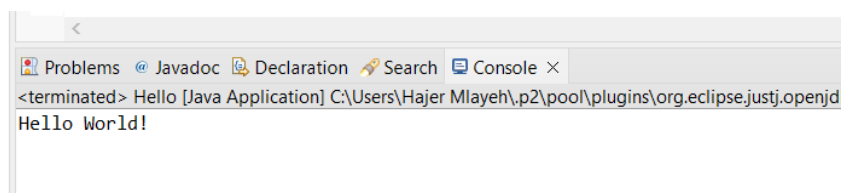
- Ajouter `System.out.println ("Hello World!");`;



- Run As -> JavaApplication : pour tester le programme .



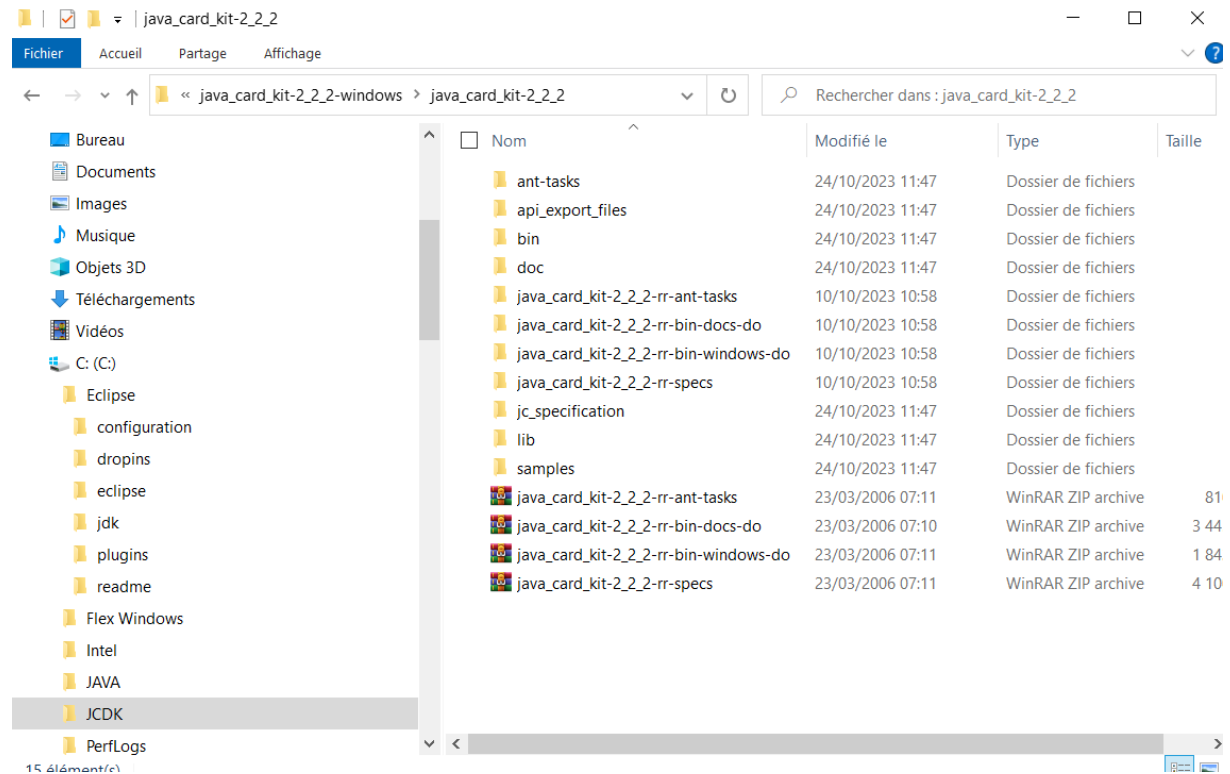
- L'affichage de l'exécution en "console" :





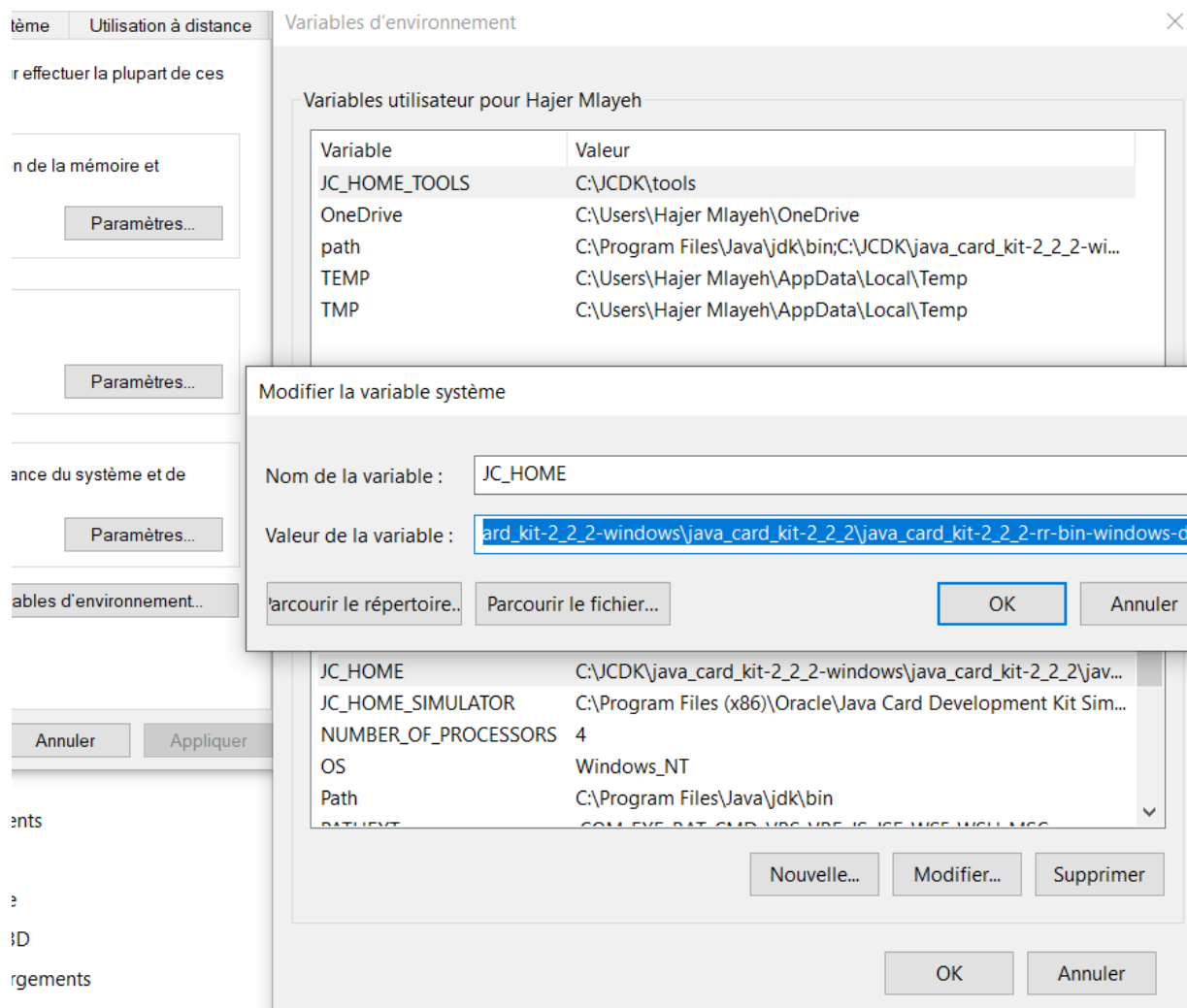
## I.2.2 Installation du Java CardDevelopment Kit 2 .2.2 :

- Création d'un nouveau répertoire C:\JCDK et décompression de l'archive java\_card\_kit-2\_2\_2-windows.zip .
- Dans le répertoire C:\JCDK \java\_card\_kit-2\_2\_2 ,affichagedes fichiers suivants :



1. Propriétés > Paramètres système Avancés > Variables d'environnement

2. Création d'une nouvelle variable d'environnement JC\_HOME contenant le chemin C:\JCDK\java\_card\_kit-2\_2\_2



#### 4. Vérification de la réussite de configuration en tapant « apdutil » dans la console

```

C:\Users\HP>apdutil
Microsoft Windows [version 10.0.22621.2715]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HP>apdutil
Java Card 2.2.2 APDU Tool, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Opening connection to localhost on port 9025.
java.net.ConnectException: Connection refused: connect

C:\Users\HP>

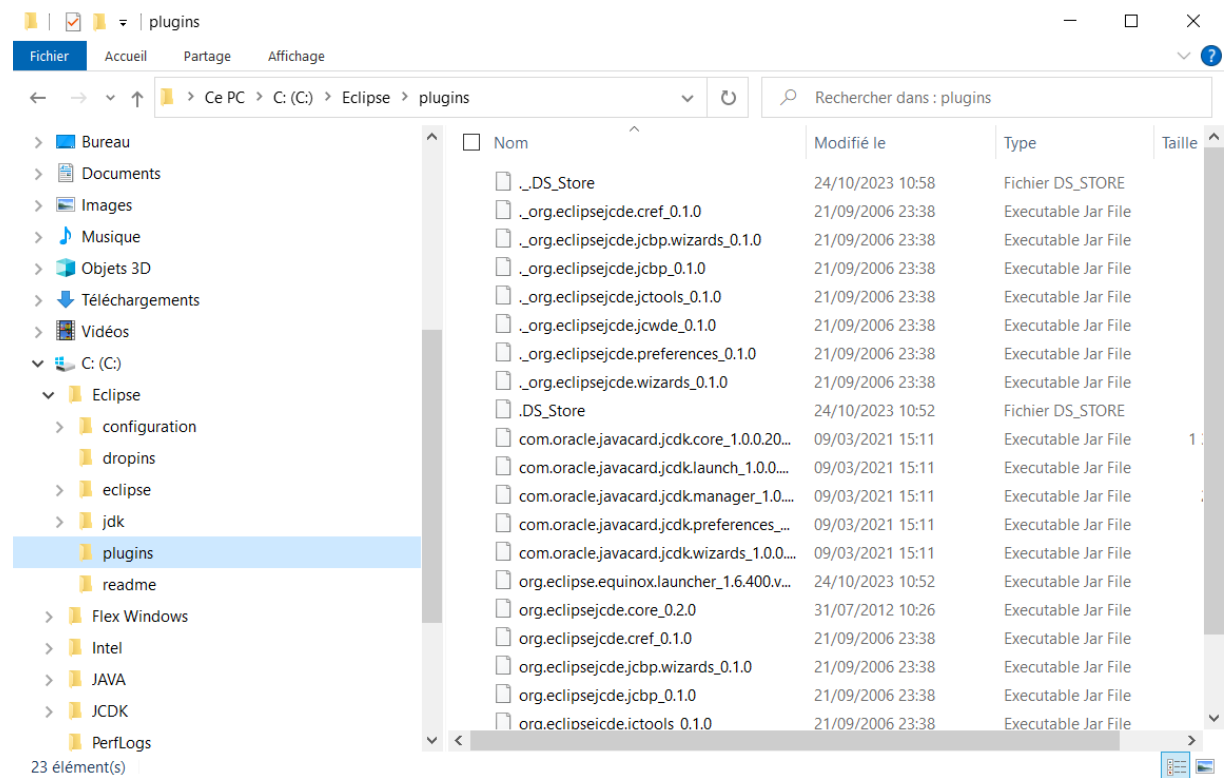
```

#### 1.2.3 Mise à jour des plugins Eclipse-JCDE :

- Télécharger eclipse-jcde-0.2.zip à partir de ce lien :

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

- . Décompression de l'archive eclipse-jcde-0.2.zip après l'avoir téléchargée dans le répertoire C:\Eclipse\eclipse\plugins pour assurer l'installation.
- . Recopier le contenu du sous-dossier "plugins " directement sous C:\Eclipse\eclipse\plugins.

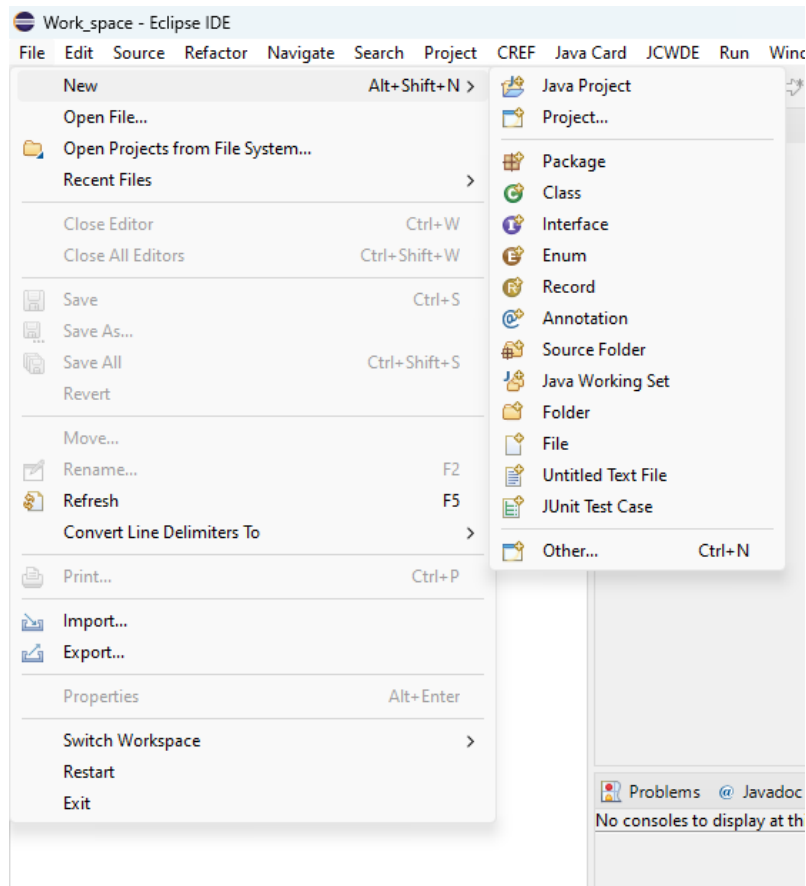


# Développement d'une application coté serveur

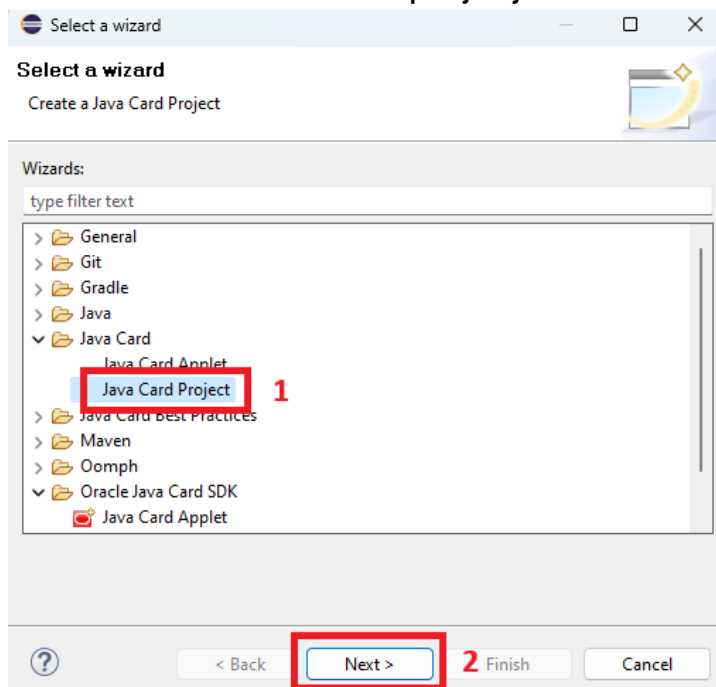
## I.1.1 Création de l'applet card sous Eclipse :

### 1. Création d'un nouveau projet :

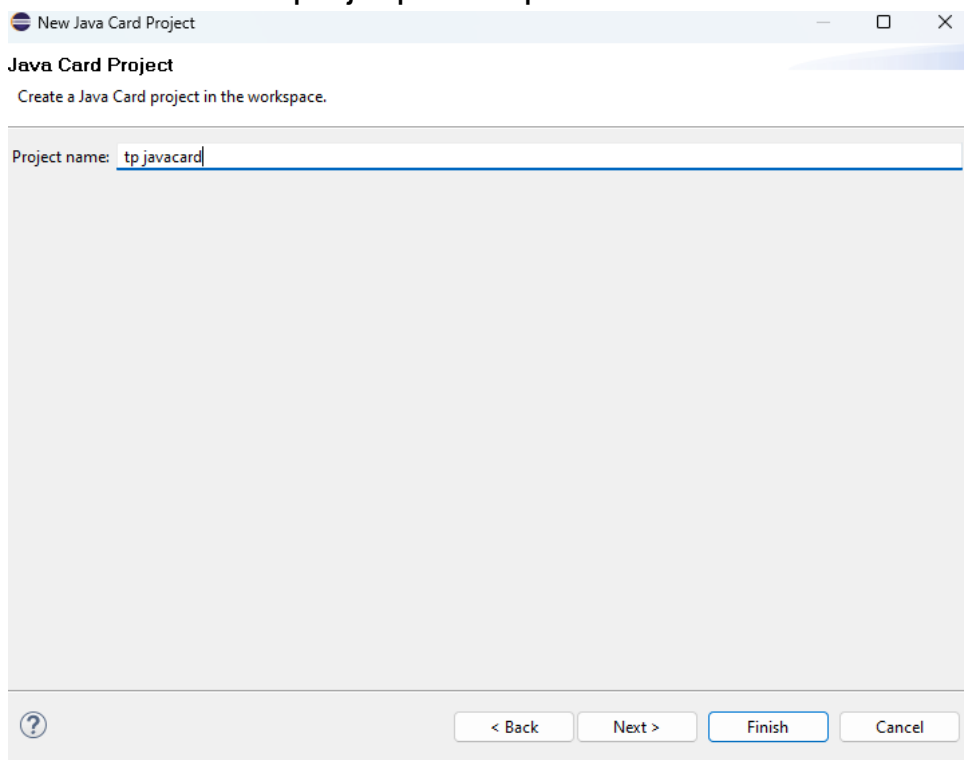
- Ouvrir Eclipse
- Aller dans file -> New ->Other.. :



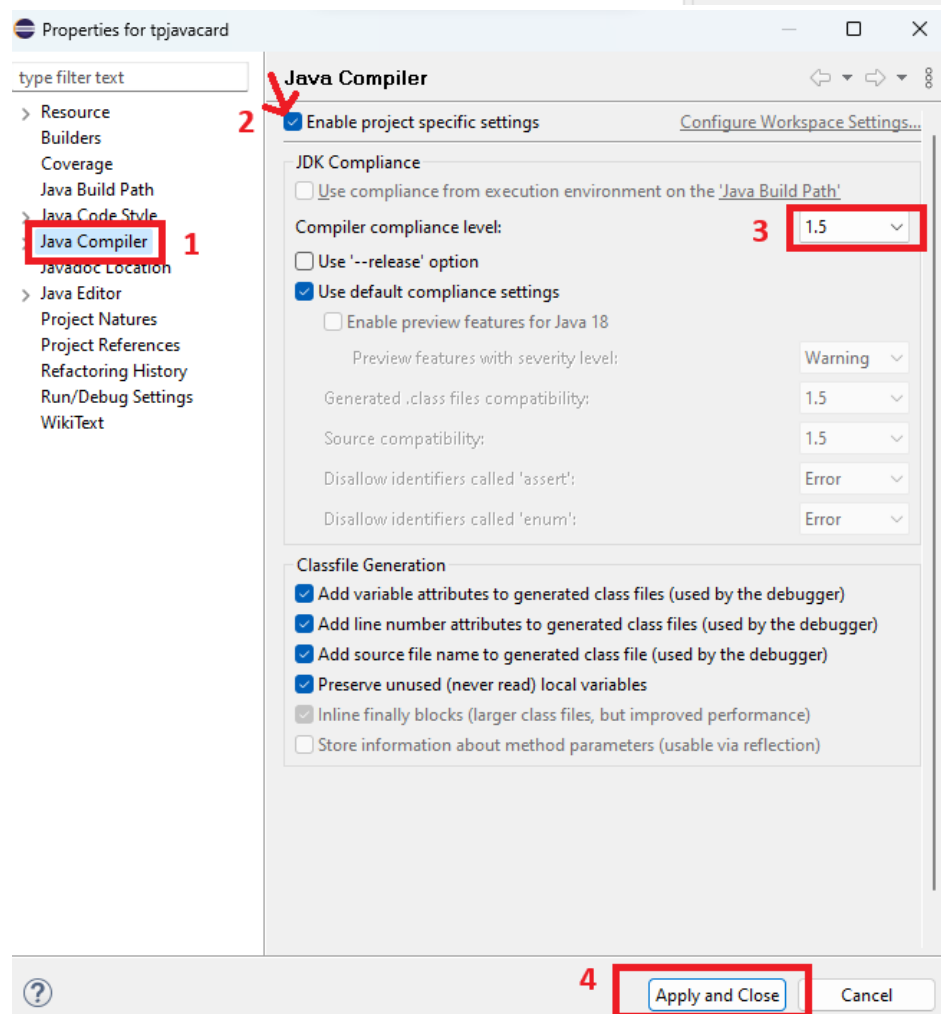
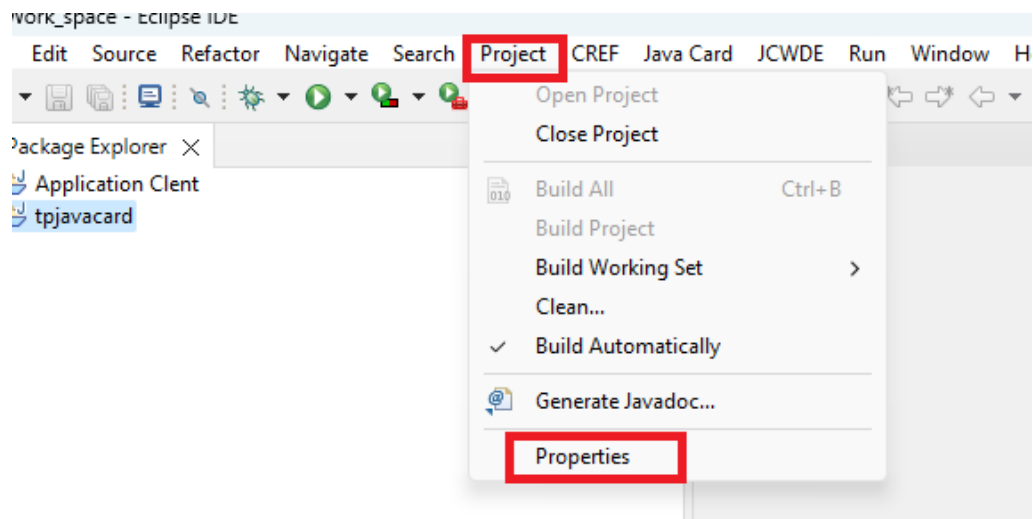
- Créer un nouveau projet java card :



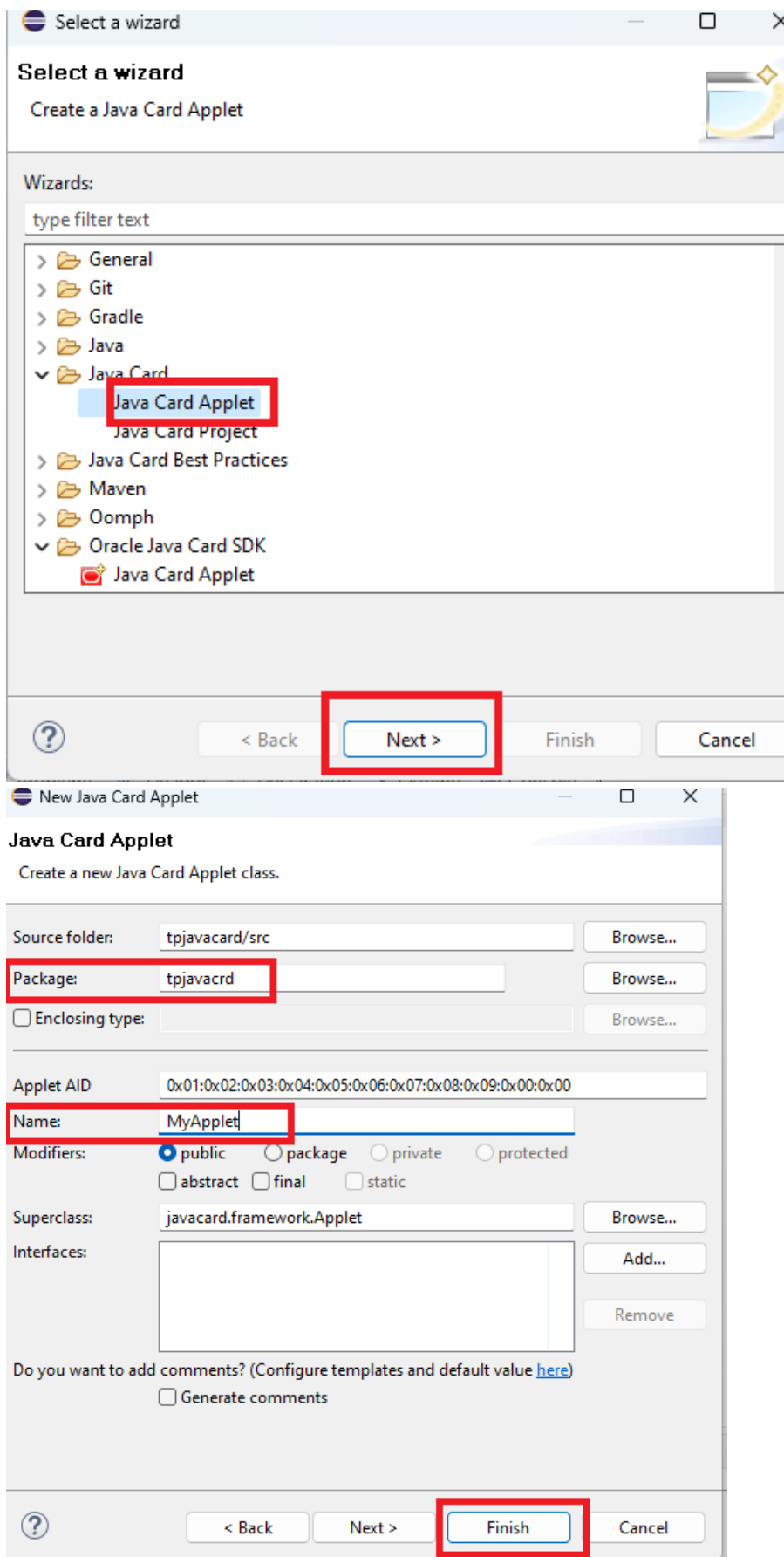
- Nommer le projet puis cliquer sur finish :



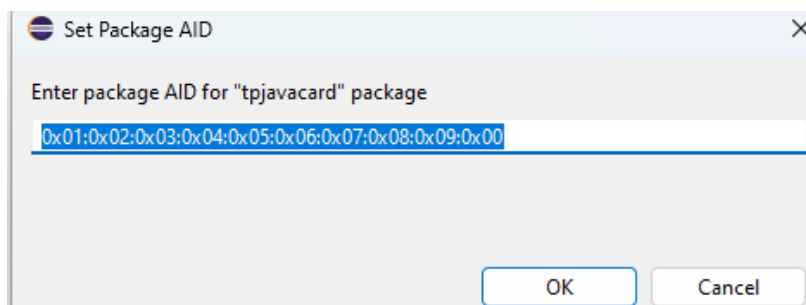
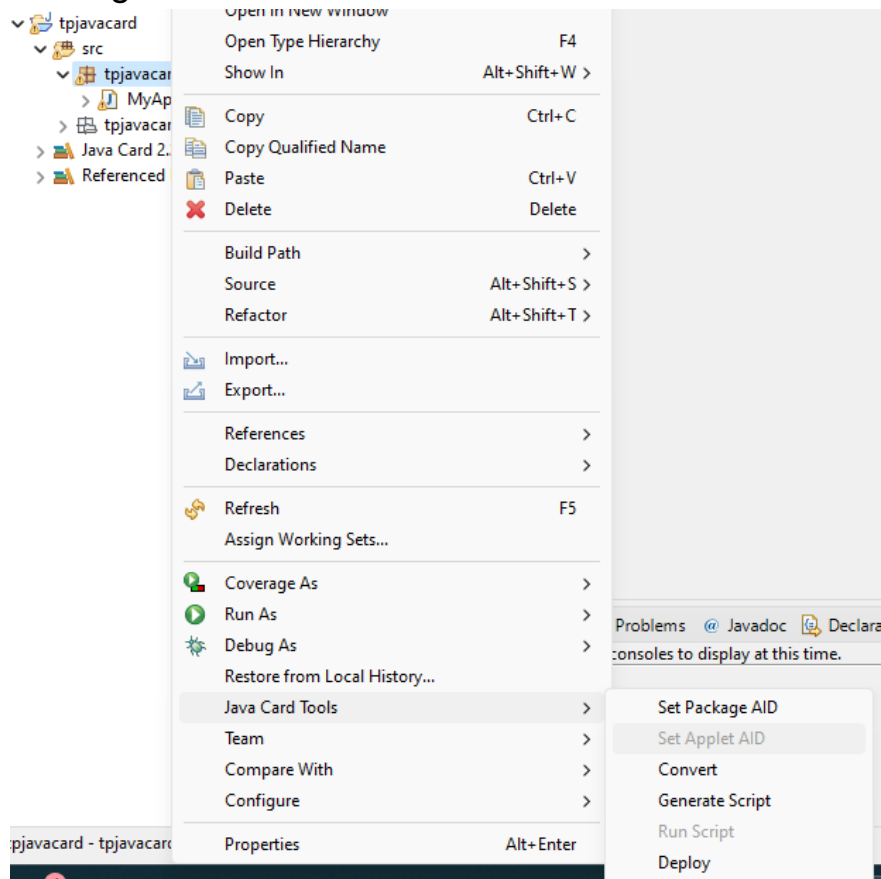
- Changer le compiler compliancelevel pour éviter les erreurs d'incompatibilité entre le « Java Card Class File Converter tool » et le « class file format version » en suivant ces étapes :



## Création d'une applet Javacard:



- set the package AID :  
unclicdroitsur le package tpjavacard -> Java Card Tools -> Set Package AID -> OK



### I.1.2 Codage de l'applet :

- Définir les constantes, le constructeur et les méthodes install et process :



```

1 package tpjavacard;
2
3 import javacard.framework.APDU;
4
5
6
7
8 public class MyApplet extends Applet {
9
10     public static final byte CLA_MONAPPLET = (byte) 0xB0;
11
12     public static final byte INS_INCREMENTER_COMPTEUR = 0x00;
13     public static final byte INS_DECREMENTER_COMPTEUR = 0x01;
14     public static final byte INS_INTERROGER_COMPTEUR = 0x02;
15     public static final byte INS_INITIALISER_COMPTEUR = 0x03;
16
17     /* Attributs */
18     private byte compteur;
19
20     private MyApplet() {
21         compteur = 0;
22     }
23
24     public static void install(byte bArray[], short bOffset, byte bLength) throws ISOException {
25         new MyApplet().register();
26     }
27
28     // @Override
29     public void process(APDU apdu) throws ISOException {
30         // TODO Auto-generated method stub
31         byte[] buffer = apdu.getBuffer();
32         if (this.selectingApplet()) return;
33

```

```

    public void process(APDU apdu) throws ISOException {
        // TODO Auto-generated method stub
        byte[] buffer = apdu.getBuffer();
        if (this.selectingApplet()) return;

        if (buffer[ISO7816.OFFSET_CLA] != CLA_MONAPPLET) {
            ISOException.throwIt(ISO7816.SW_CLA_NOT_SUPPORTED);
        }
        switch (buffer[ISO7816.OFFSET_INS]) {
            case INS_INCREMENTER_COMPTEUR:
                compteur++;
                break;

            case INS_DECREMENTER_COMPTEUR:
                compteur--;
                break;

            case INS_INTERROGER_COMPTEUR:
                buffer[0] = compteur;
                apdu.setOutgoingAndSend((short) 0, (short) 1);
                break;

            case INS_INITIALISER_COMPTEUR:
                apdu.setIncomingAndReceive();
                compteur = buffer[ISO7816.OFFSET_CDATA];
                break;

            default:
                ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
        }
    }
}

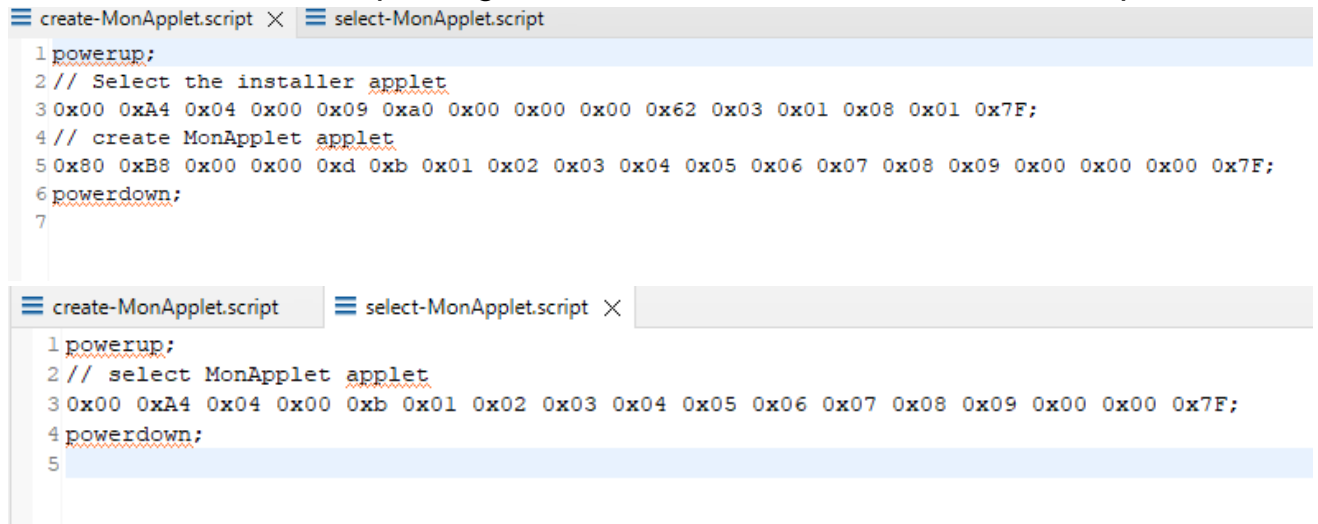
```

### I.1.3 Outils de simulation :

#### 1- JCWDE : Simulateur sans conservation d'état :

- Générer le script :

Click droit sur le package -> Java Card Tools ->Generate Script

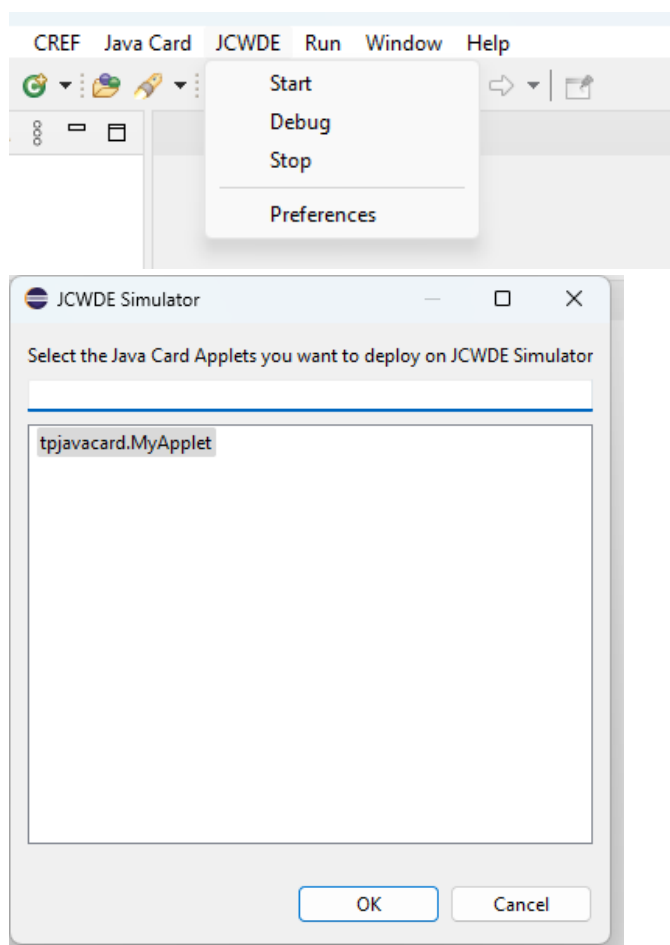


```
1 powerup;
2 // Select the installer applet
3 0x00 0xA4 0x04 0x00 0x09 0xA0 0x00 0x00 0x00 0x62 0x03 0x01 0x08 0x01 0x7F;
4 // create MonApplet applet
5 0x80 0xB8 0x00 0x00 0xd 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x00 0x7F;
6 powerdown;
7

1 powerup;
2 // select MonApplet applet
3 0x00 0xA4 0x04 0x00 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x7F;
4 powerdown;
5
```

- Démarrer JCWDE (simulator) :

JCWDE -> Start -> OK



- Taper apdutooldanscmd

```

C:\Users\HP>apdutooldanscmd
Microsoft Windows [version 10.0.22621.2715]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HP>apdutoold
Java Card 2.2.2 APDU Tool, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Opening connection to localhost on port 9025.
Connected.

```

- Créer l'applet en copiant l'APDU de script create-MonApplet sauf powerdown ;
- sélectionner l'applet en recopiant l'APDU du script select-MonApplet (bien sur sans powerdown ;) )

```

C:\Users\HP>apdutoold
Java Card 2.2.2 APDU Tool, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Opening connection to localhost on port 9025.
Connected.
powerup;
// Select the installer applet
0x00 0xA4 0x04 0x00 0x09 0xA0 0x00 0x00 0x62 0x03 0x01 0x08 0x01 0x7F;
// create MonApplet applet
Received ATR = 0x3b 0xf0 0x11 0x00 0xff 0x00
CLA: 00, INS: a4, P1: 04, P2: 00, Lc: 09, a0, 00, 00, 00, 62, 03, 01, 08, 01, Le: 00, SW1: 90, SW2: 00
0x80 0xB8 0x00 0x00 0xd 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x7F;
CLA: 80, INS: b8, P1: 00, P2: 00, Lc: 0d, 0b, 01, 02, 03, 04, 05, 06, 07, 08, 09, 00, 00, 00, Le: 0b, 01, 02, 03, 04, 05,
06, 07, 08, 09, 00, 00, SW1: 90, SW2: 00
powerup;
// select MonApplet applet
Received ATR = 0x3b 0xf0 0x11 0x00 0xff 0x00
0x00 0xA4 0x04 0x00 0xb 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x00 0x00 0x7F;
CLA: 00, INS: a4, P1: 04, P2: 00, Lc: 0b, 01, 02, 03, 04, 05, 06, 07, 08, 09, 00, 00, Le: 00, SW1: 90, SW2: 00

```

Nous voyons que la carte répond positivement avec un statusword 0x9000

- Tester l'applet :
  - 1- Interroger le compteur
  - 2- Incrémenter le compteur
  - 3- Interroger de nouveau
  - 4- Initialiser le compteur a 0x4A
  - 5- Décrémenter le compteur
  - 6- Interroger de nouveau

```

0xB0 0x02 0x00 0x00 0x00 0x7F;
CLA: b0, INS: 02, P1: 00, P2: 00, Lc: 00, Le: 01, 00, SW1: 90, SW2: 00

0xB0 0x00 0x00 0x00 0x00 0x7F;
CLA: b0, INS: 00, P1: 00, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00

0xB0 0x02 0x00 0x00 0x00 0x7F;
CLA: b0, INS: 02, P1: 00, P2: 00, Lc: 00, Le: 01, 01, SW1: 90, SW2: 00

0xB0 0x03 0x00 0x00 0x01 0x4A 0x7F;
CLA: b0, INS: 03, P1: 00, P2: 00, Lc: 01, 4a, Le: 00, SW1: 90, SW2: 00

0xB0 0x01 0x00 0x00 0x00 0x7F;
CLA: b0, INS: 01, P1: 00, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00

0xB0 0x02 0x00 0x00 0x00 0x7F;
CLA: b0, INS: 02, P1: 00, P2: 00, Lc: 00, Le: 01, 49, SW1: 90, SW2: 00

```

- Déconnecter le simulateur en tapant powerdown; dans apdutool ce qui provoque la fermeture de JCWDE dans la console d'eclipse :

```

Problems  @ Javadoc  Declaration  Outline  Console x
<terminated> JCWDE [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831
Java Card 2.2.2 Workstation Development Environment, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
jcwde is listening for T=1 Adu's on TCP/IP port 90025.
jcwde exiting on receipt of power down command.

```

## 2- CREF :simulateur avec conservation d'état

- Lancer CREF et créer le fichier image :

```

C:\Users\HP>cref -o myapplet.eeprom
Java Card 2.2.2 C Reference Implementation Simulator (version 0.41)
32-bit Address Space implementation - with cryptography support
T=1 / T=CL Dual interface APDU protocol (ISO 7816-3)
Copyright 2005 Sun Microsystems, Inc. All rights reserved.

Memory configuration
  Type   Base   Size   Max Addr
  ----   -
  RAM    0x0     0x1000 0xffff
  ROM    0x2000 0xe000 0xffff
  E2P    0x10020 0xffe0 0x1ffff

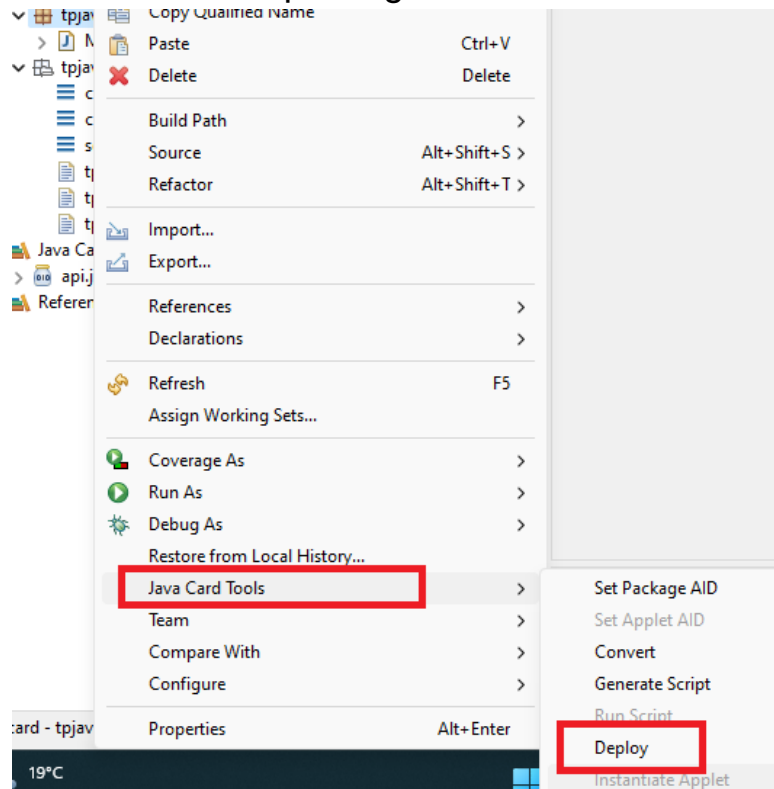
  ROM Mask size =          0xce64 =      52836 bytes
  Highest ROM address in mask = 0xee63 =      61027 bytes
  Space available in ROM =    0x119c =      4508 bytes

EEPROM will be saved in file "myapplet.eeprom"
Mask has now been initialized for use

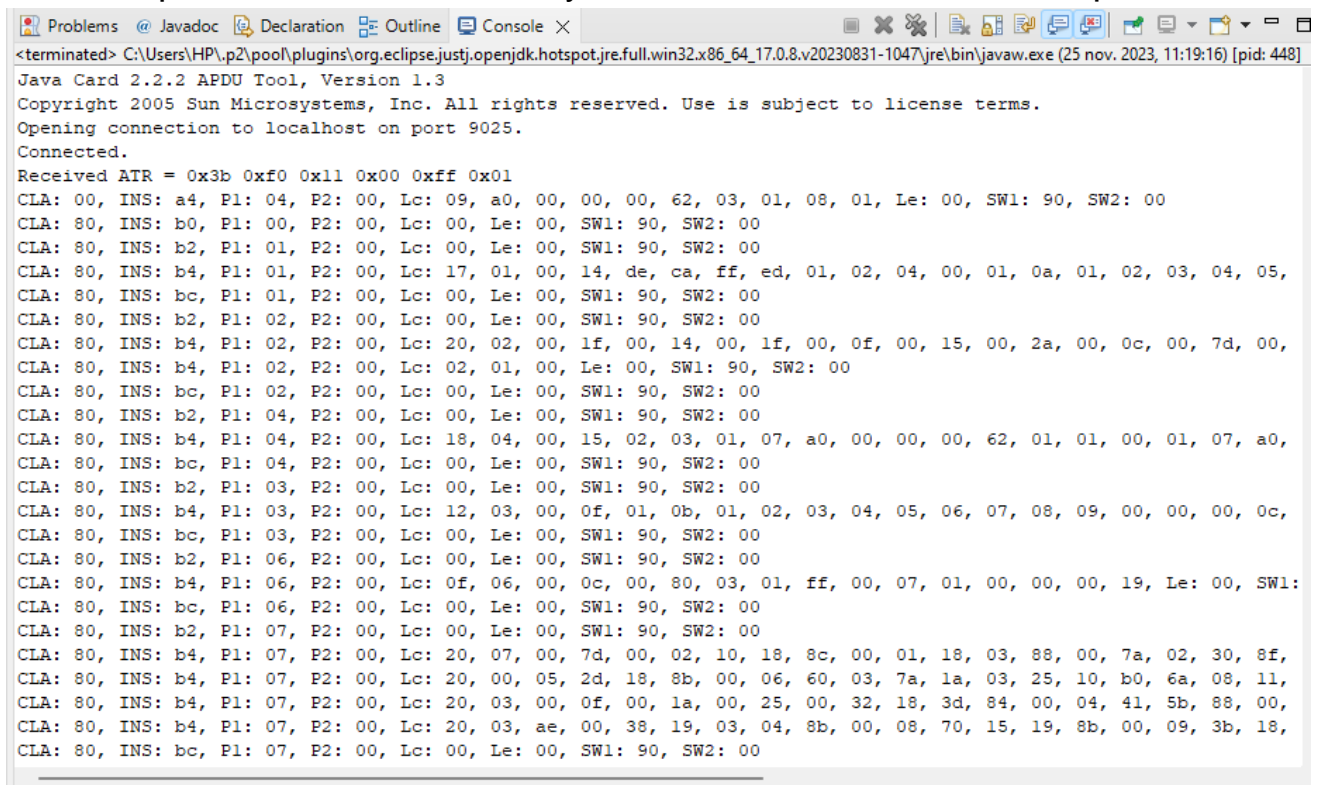
```

- Uploader l'applet :

Click droit sur le package -> Java Card Tools ->Deploy :



- On peut voir les APDU envoyés dans la zone Console d'Eclipse :



- Relancer CREF et recharger le fichier image :

```

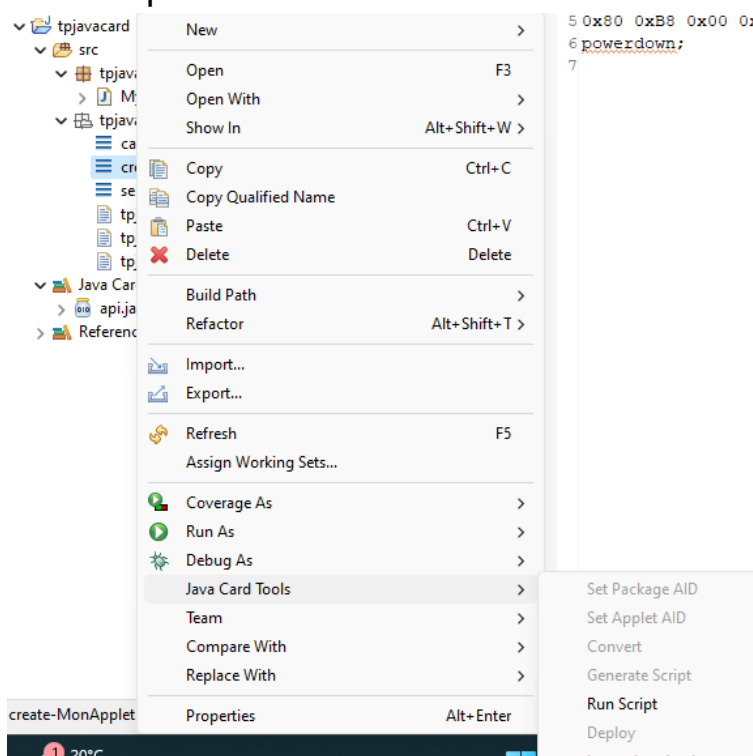
C:\Users\HP>cref -i myapplet.eeprom -o myapplet.eeprom
Java Card 2.2.2 C Reference Implementation Simulator (version 0.41)
32-bit Address Space implementation - with cryptography support
T=1 / T=CL Dual interface APDU protocol (ISO 7816-3)
Copyright 2005 Sun Microsystems, Inc. All rights reserved.

Memory configuration
  Type   Base   Size   Max Addr
  RAM    0x0    0x1000 0xffff
  ROM    0x2000 0xe000 0xffff
  E2P    0x10020 0xffe0 0x1ffff

  ROM Mask size =          0xce64 =      52836 bytes
  Highest ROM address in mask = 0xee63 =    61027 bytes
  Space available in ROM =    0x119c =    4508 bytes
EEPROM (0xffe0 bytes) restored from file "myapplet.eeprom"
Using a pre-initialized Mask

```

- Installer l'applet :  
Click droit sur script create-MonApplet.script -> Java Card Tools -> Run Script



```

<terminated> C:\Users\HP\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe (25 nov. 2023, 11:56:30) [pid: 84]
Java Card 2.2.2 APDU Tool, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Opening connection to localhost on port 9025.
Connected.
Received ATR = 0x3b 0xf0 0x11 0x00 0xff 0x01
CLA: 00, INS: a4, P1: 04, P2: 00, Lc: 09, a0: 00, 00, 00, 62, 03, 01, 08, 01, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: b8, P1: 00, P2: 00, Lc: 0d, 0b, 01, 02, 03, 04, 05, 06, 07, 08, 09, 00, 00, 00, Le: 0b, 01, 02, 03,

```

- Relancer CREF :

```

C:\Users\HP>cref -i myapplet.eeprom -o myapplet.eeprom
Microsoft Windows [version 10.0.22621.2715]
(c) Microsoft Corporation. Tous droits réservés.

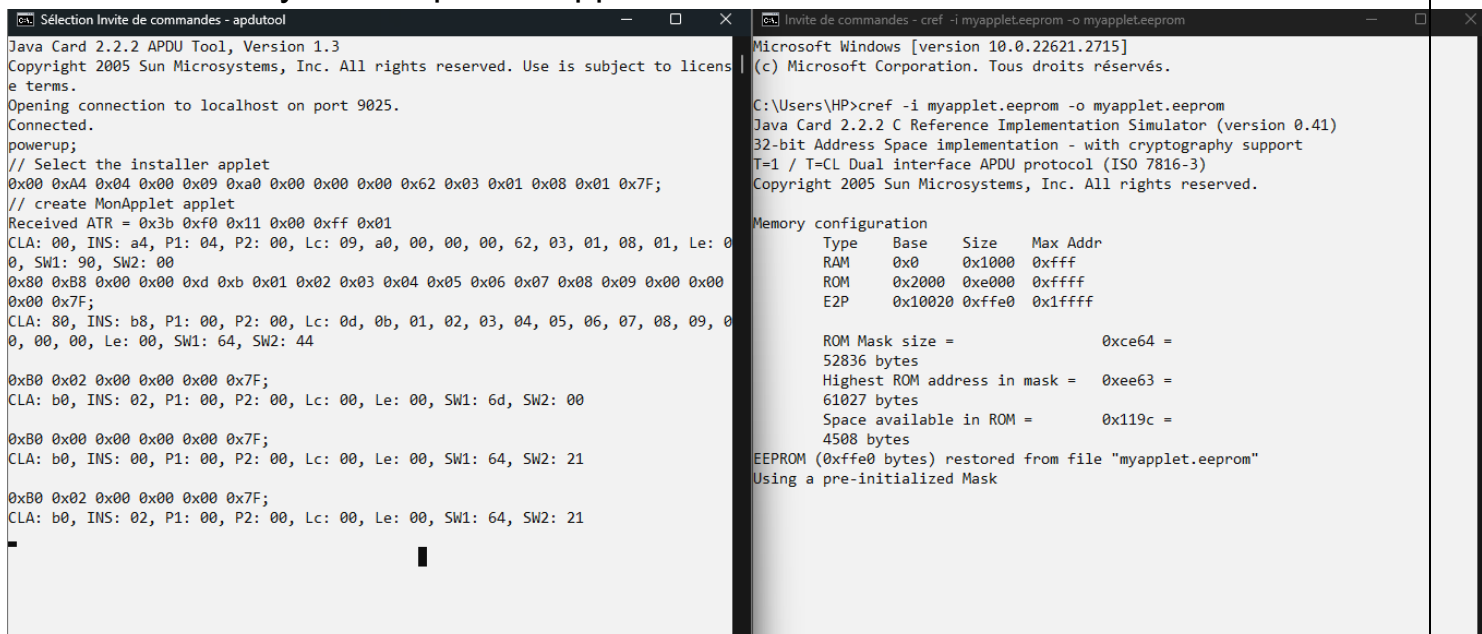
C:\Users\HP>cref -i myapplet.eeprom -o myapplet.eeprom
Java Card 2.2.2 C Reference Implementation Simulator (version 0.41)
32-bit Address Space implementation - with cryptography support
T=1 / T=CL Dual interface APDU protocol (ISO 7816-3)
Copyright 2005 Sun Microsystems, Inc. All rights reserved.

Memory configuration
  Type   Base   Size   Max Addr
  RAM    0x0    0x1000 0xffff
  ROM    0x2000 0xe000 0xffff
  E2P    0x10020 0xffe0 0x1ffff

  ROM Mask size =          0xce64 =
  52836 bytes
  Highest ROM address in mask = 0xee63 =
  61027 bytes
  Space available in ROM =    0x119c =
  4508 bytes
EEPROM (0xffe0 bytes) restored from file "myapplet.eeprom"
Using a pre-initialized Mask

```

- Ouvrir une autre fenêtre de commande
- Lancer apdutool
- Sélectionner l'applet
- Envoyer des apdu à l'applet



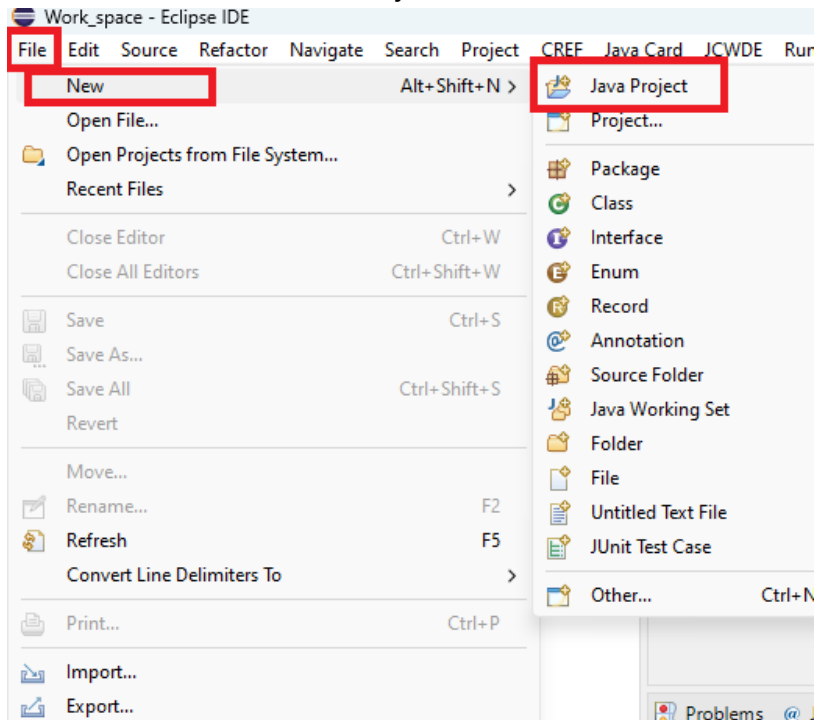


# Programmation d'une application coté client

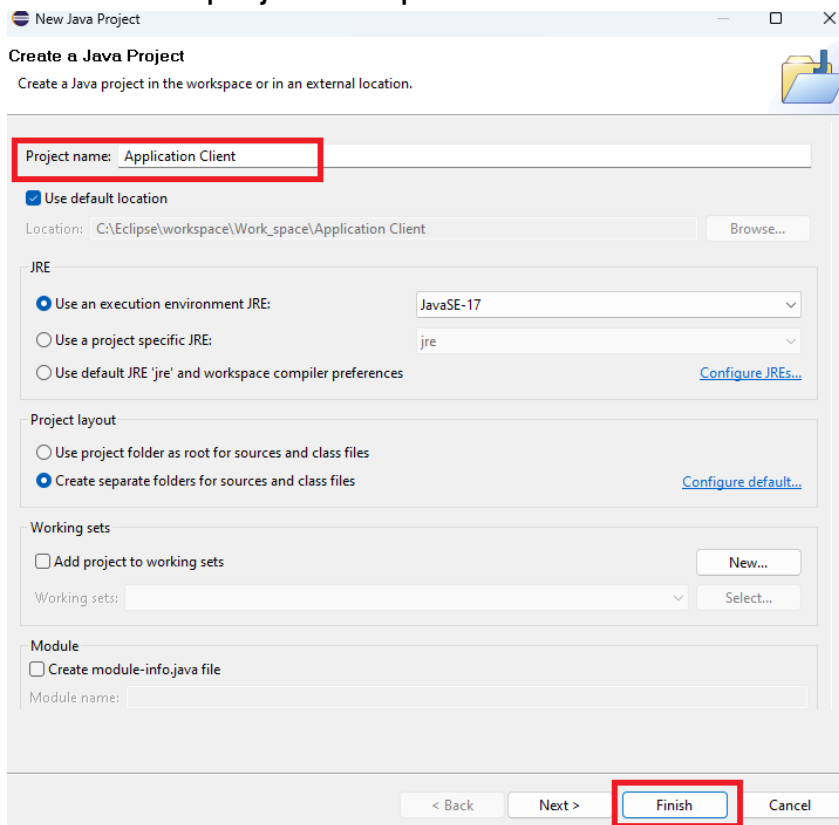
## I.1Création de l'application client sous Eclipse :

1- Créer un nouveau java projet :

File -> New -> Java Project

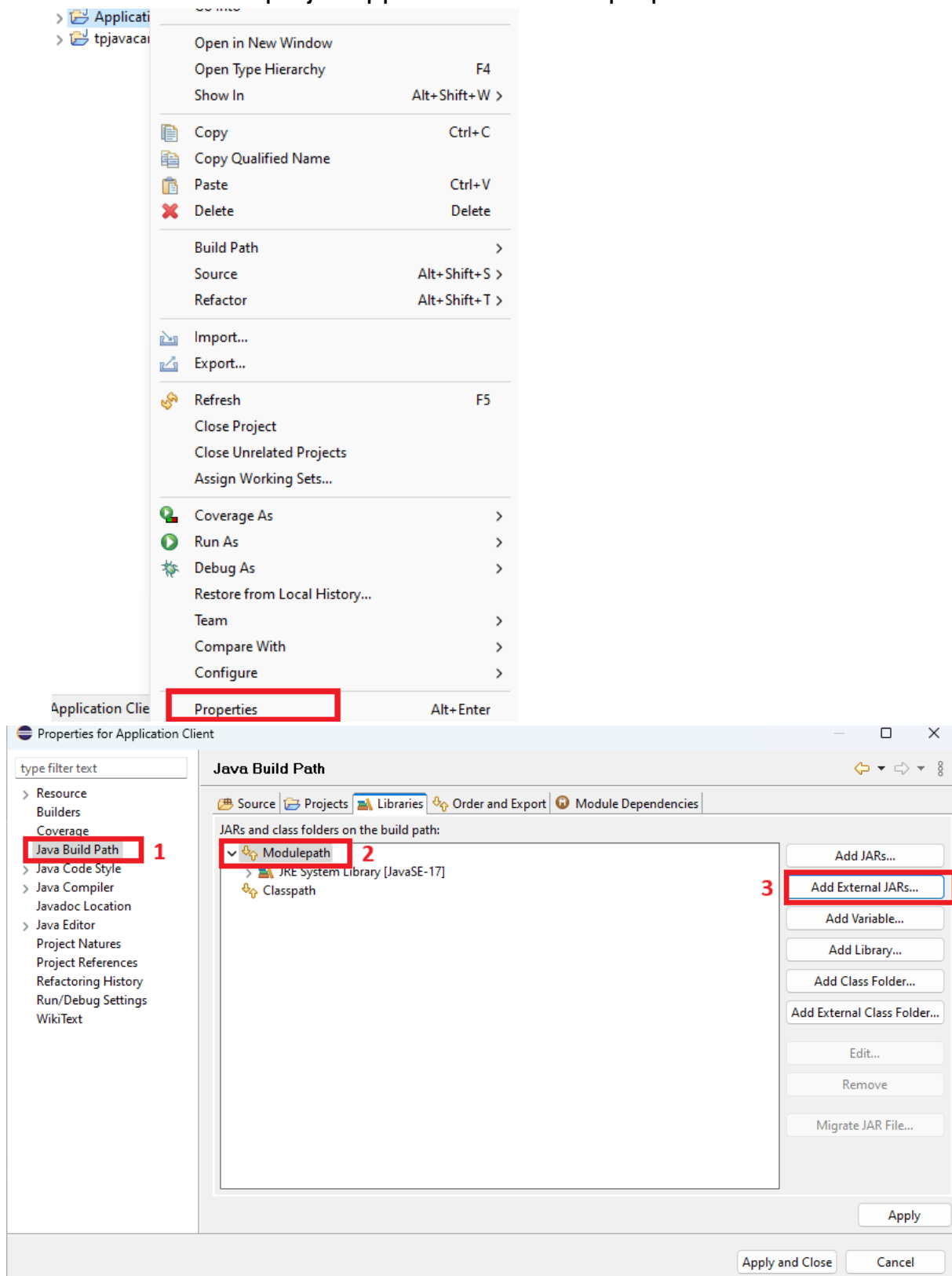


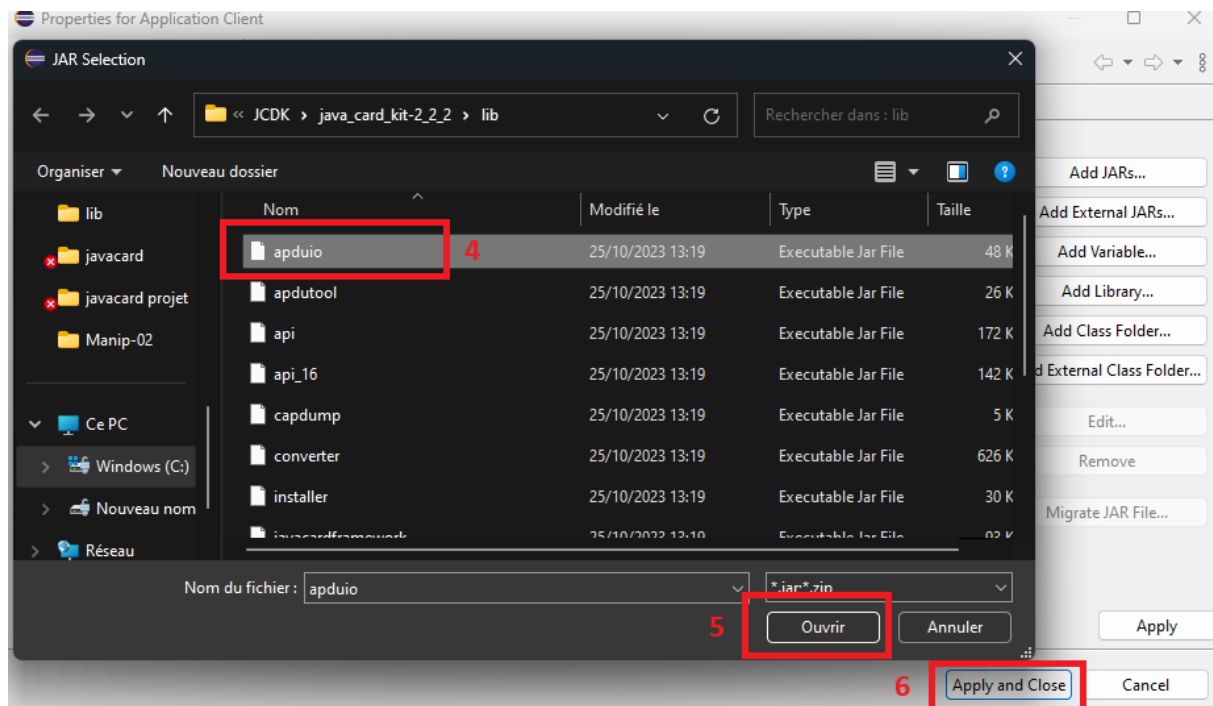
Nommer le projet et cliquer sur Finish :



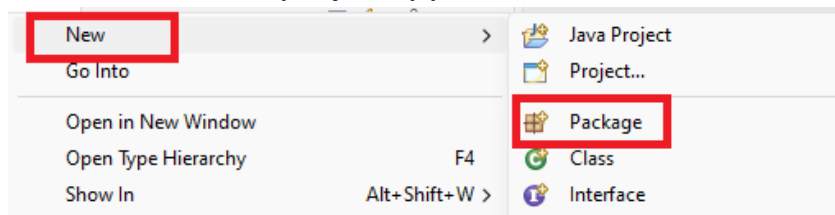


- Ajouter la librairie apduio dans le class path :  
Click droit sur le projet Application Client ->properties :

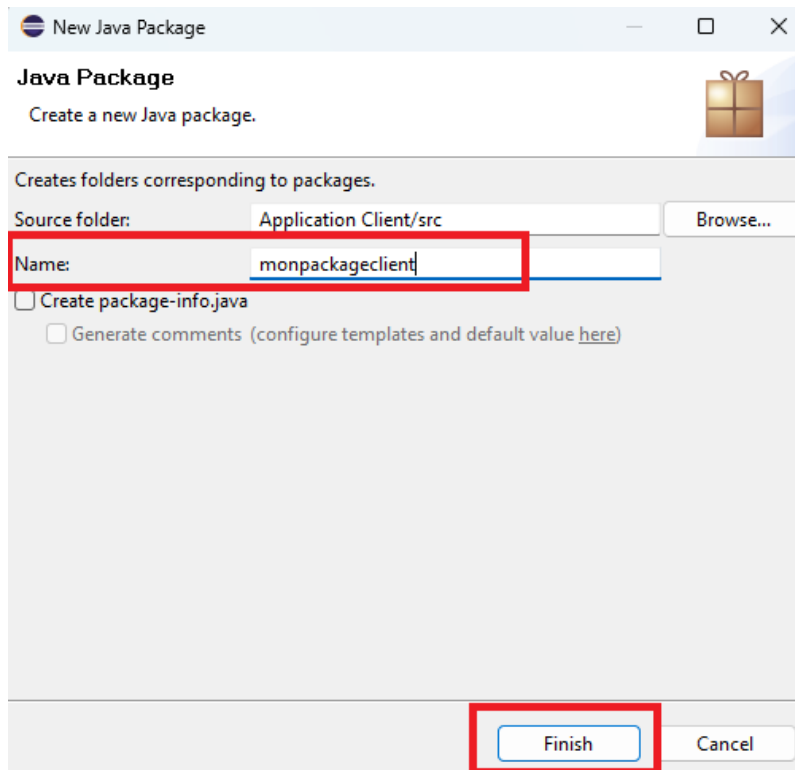




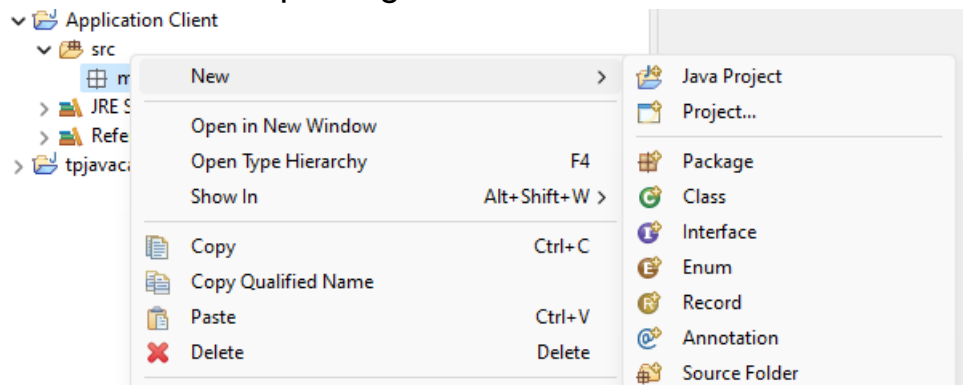
- Créer un nouveau package :  
Click droit sur le projet Application Client -> New -> Package

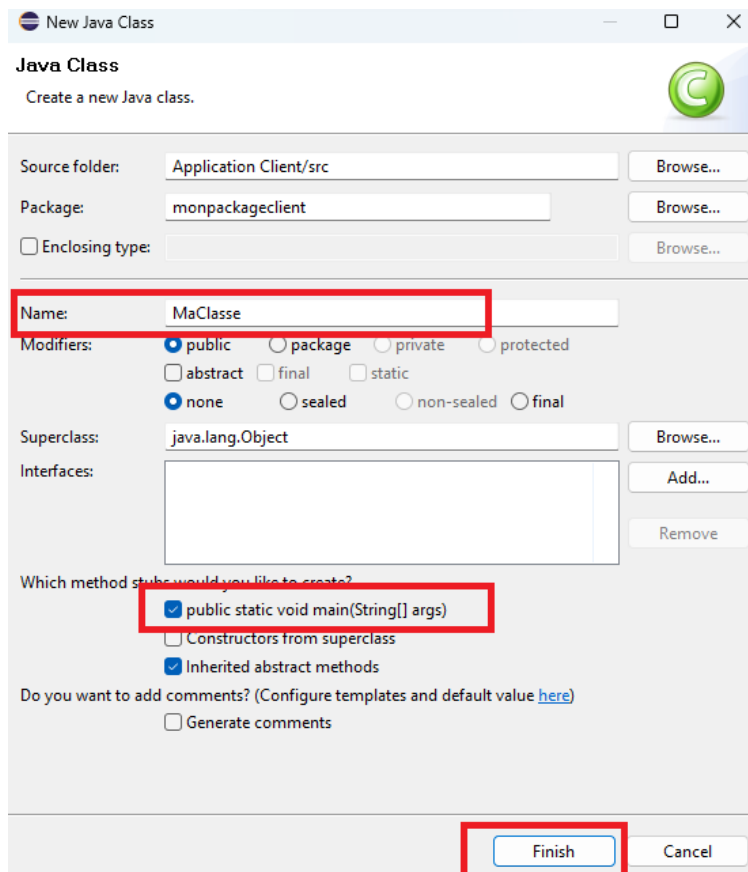


Nommer le package et cliquer sur finish :



- Créer la classe principale de l'application :  
Click droit sur le package -> New -> Class





- On peut séparer l'écriture de notre application en plusieurs étapes :

## Étape 0—Addition des importations , et déclaration des attributs statiques :

```

2 import java.net.Socket;
3 import java.io.*;
4 import com.sun.javacard.apduio.Apdu;
5 import com.sun.javacard.apduio.*;
6 import com.sun.javacard.apduio.CadT1Client;
7 import com.sun.javacard.apduio.CadTransportException;
8
9
10
11 public class MaClasse {
12     public static final byte CLA_MONAPPLET = (byte) 0xB0;
13     public static final byte INS_INCREMENTER_COMPTEUR = 0x00;
14     public static final byte INS_DECREMENTER_COMPTEUR = 0x01;
15     public static final byte INS_INTERROGER_COMPTEUR = 0x02;
16     public static final byte INS_INITIALISER_COMPTEUR = 0x03;

```

## 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.

```

public static void main(String[] args) throws IOException, CadTransportException {
    /* Connexion - Javacard */
    CadT1Client cad;
    Socket sckCarte;
    try {
        sckCarte = new Socket("localhost", 9025);
        sckCarte.setTcpNoDelay(true);
        BufferedInputStream input = new BufferedInputStream(sckCarte.getInputStream());
        BufferedOutputStream output = new BufferedOutputStream(sckCarte.getOutputStream());
        cad = new CadT1Client(input, output);
    } catch (Exception e) {
        System.out.println("Erreur : impossible de se connecter a la Javacard");
        return;
    }
    /* Mise sous tension de la carte */
    try {
        cad.powerUp();
    }
    catch (Exception e)
    {
        System.out.println("Erreur lors de l'envoi de la Powerup a la Javacard");
        return;
    }
}

```

## Etape 2 -Sélection

La sélection d'applet se fait en envoyant la commande SELECT APDU

```

/* Sélection de l'applet */
Apu apdu = new Apdu();
apdu.command[Apu.CLA] = 0x00;
apdu.command[Apu.INS] = (byte) 0xA4;
apdu.command[Apu.P1] = 0x04;
apdu.command[Apu.P2] = 0x00;
byte[] appletAID = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x00, 0x00 };
apdu.setDataIn(appletAID);
cad.exchangeApu(apdu);
if (apdu.getStatus() != 0x9000) {
    System.out.println("Erreur lors de la sélection de l'applet");
    System.exit(1);
}

```

## 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.

```

/* Menu principal */
boolean fin = false;
while (!fin) {
    System.out.println();
    System.out.println("Application cliente Javacard");
    System.out.println("-----");
    System.out.println();
    System.out.println("1 - Interroger le compteur");
    System.out.println("2 - Incrémenter le compteur");
    System.out.println("3 - Décrémenter le compteur");
    System.out.println("4 - Réinitialiser le compteur");
    System.out.println("5 - Quitter");
    System.out.println();
    System.out.println("Votre choix ?");
    int choix = System.in.read();
    while (!(choix >= '1' && choix <= '5')) {
        choix = System.in.read();
    }
    apdu = new Adu();
    apdu.command[Adu.CLA] = MaClasse.CLA_MONAPPLET;
    apdu.command[Adu.P1] = 0x00;
    apdu.command[Adu.P2] = 0x00;
    apdu.setLe(0x7f);
    switch (choix) {
        case '1':
            apdu.command[Adu.INS] = MaClasse.INS_INTERROGER_COMPTEUR;
            cad.exchangeAdu(apdu);
            if (apdu.getStatus() != 0x9000) {
                System.out.println("Erreur : status word different de 0x9000");
                System.out.println(apdu.getStatus());
            } else {
                System.out.println("Valeur du compteur : " + apdu.dataOut[0]);
            }
            break;
        case '2':
            apdu.command[Adu.INS] = MaClasse.INS_INCREMENTER_COMPTEUR;
            cad.exchangeAdu(apdu);
            if (apdu.getStatus() != 0x9000) {
                System.out.println("Erreur : status word different de 0x9000");
            } else {
                System.out.println("OK");
            }
            break;
        case '3':
            apdu.command[Adu.INS] = MaClasse.INS_DECREMENTER_COMPTEUR;
            cad.exchangeAdu(apdu);
            if (apdu.getStatus() != 0x9000) {
                System.out.println("Erreur : status word different de 0x9000");
            } else {
                System.out.println("OK");
            }
            break;
        case '4':
            apdu.command[Adu.INS] = MaClasse.INS_INITIALISER_COMPTEUR;
            byte[] donnees = new byte[1];
            donnees[0] = 0;
            apdu.setDataIn(donnees);
            cad.exchangeAdu(apdu);
            if (apdu.getStatus() != 0x9000) {
                System.out.println("Erreur : status word different de 0x9000");
            }
    }
}

```

```

        case '4':
            apdu.command[Apdu.INS] = MaClasse.INS_INITIALISER_COMPTEUR;
            byte[] donnees = new byte[1];
            donnees[0] = 0;
            apdu.setDataIn(donnees);
            cad.exchangeApu(apdu);
            if (apdu.getStatus() != 0x9000) {
                System.out.println("Erreur : status word different de 0x9000");
            } else {
                System.out.println("OK");
            }
            break;
        case '5':
            fin = true;
            break;
    }
}

```

## Etape 4 - Mise hors tension

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

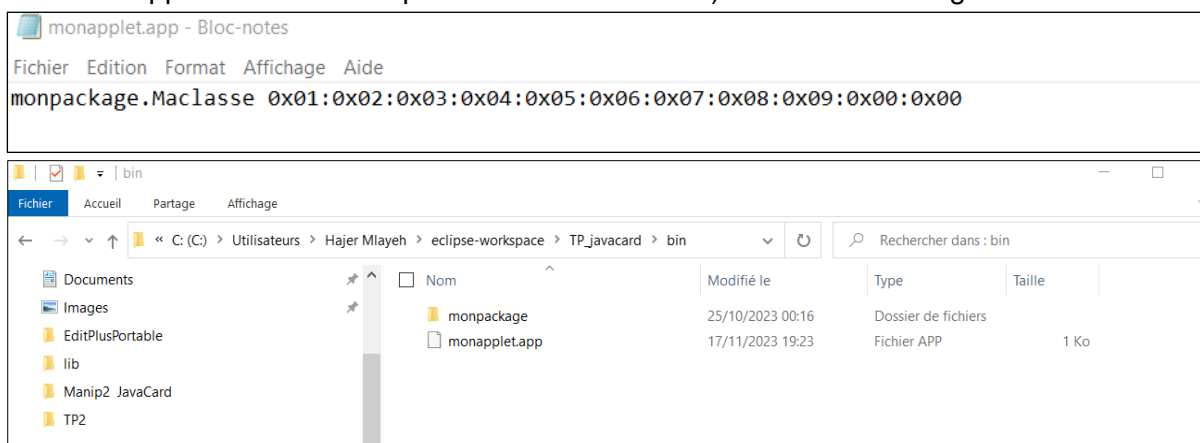
```

    }
    /* Mise hors tension de la carte */
    try {
        cad.powerDown();
    } catch (Exception e) {
        System.out.println("Erreur lors de l'envoi de la commande Powerdown a la Javacard");
        return;
    }
}

```

## I.2 Utilisation de l'application cliente avec un simulateur – JCWDE

- Création du fichier "monapplet.app" (répertoire parent du package contenant le fichier class de l'applet card « `.\workspace\TP Javacard\bin` ») .Il contiendra la lignesuivante :



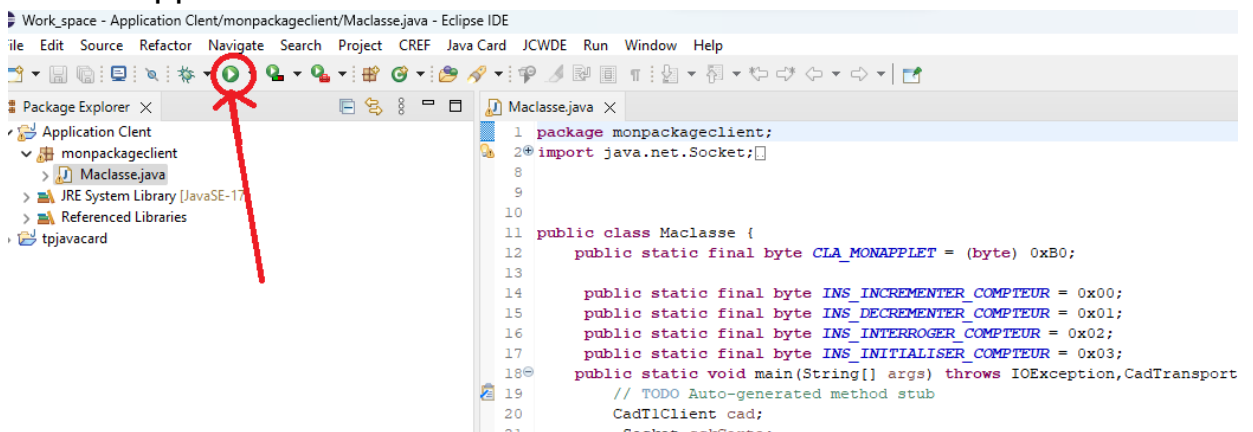
- Lancer le simulateur en prenant soin de se placer dans le bon repertoire :

```
Invite de commandes - jcwde monapplet.app
Microsoft Windows [version 10.0.22621.2715]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HP>cd C:\Eclipse\workspace\Work_space\tpjavacard\bin

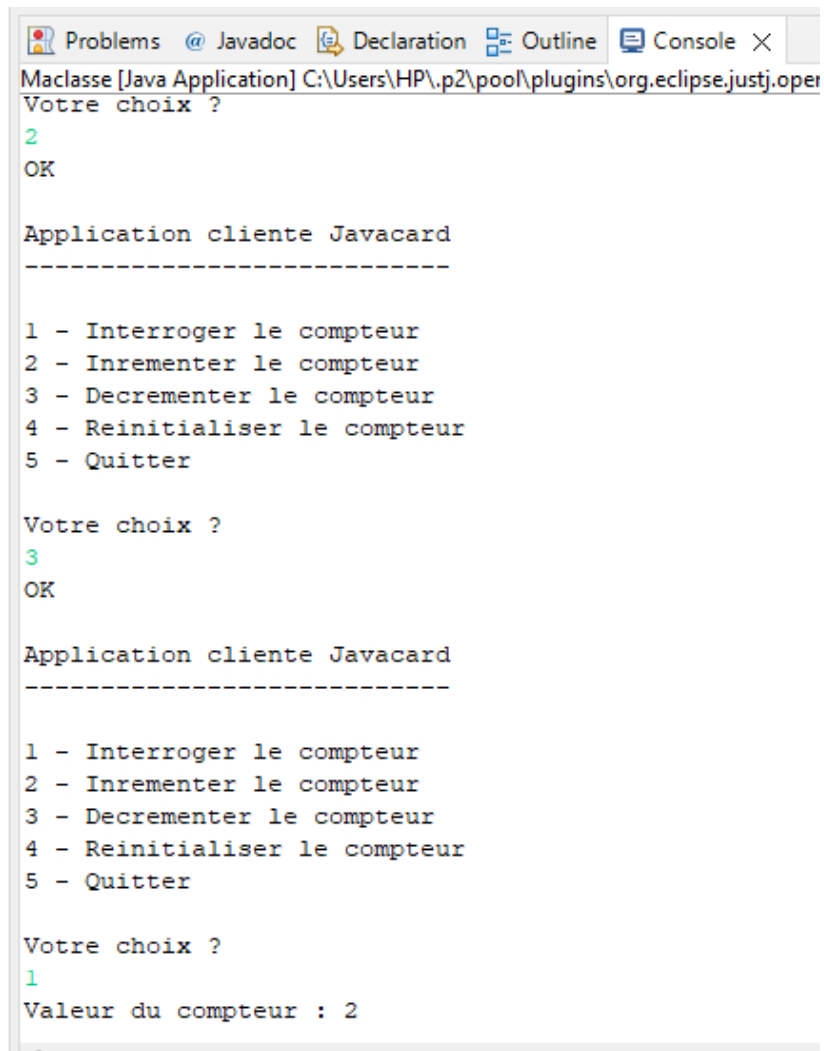
C:\Eclipse\workspace\Work_space\tpjavacard\bin>jcwde monapplet.app
Java Card 2.2.2 Workstation Development Environment, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
jcwde is listening for T=1 Adu's on TCP/IP port 9?025.
```

- lancer l'application Client :



- interroger le compteur, l'incrémenter 3 fois , le décrémenter une fois et l'interroger de nouveau :





```
Problems @ Javadoc Declaration Outline Console X
Maclasse [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.oper
Votre choix ?
2
OK

Application cliente Javacard
-----

1 - Interroger le compteur
2 - Inrementer le compteur
3 - Decrementer le compteur
4 - Reinitialiser le compteur
5 - Quitter

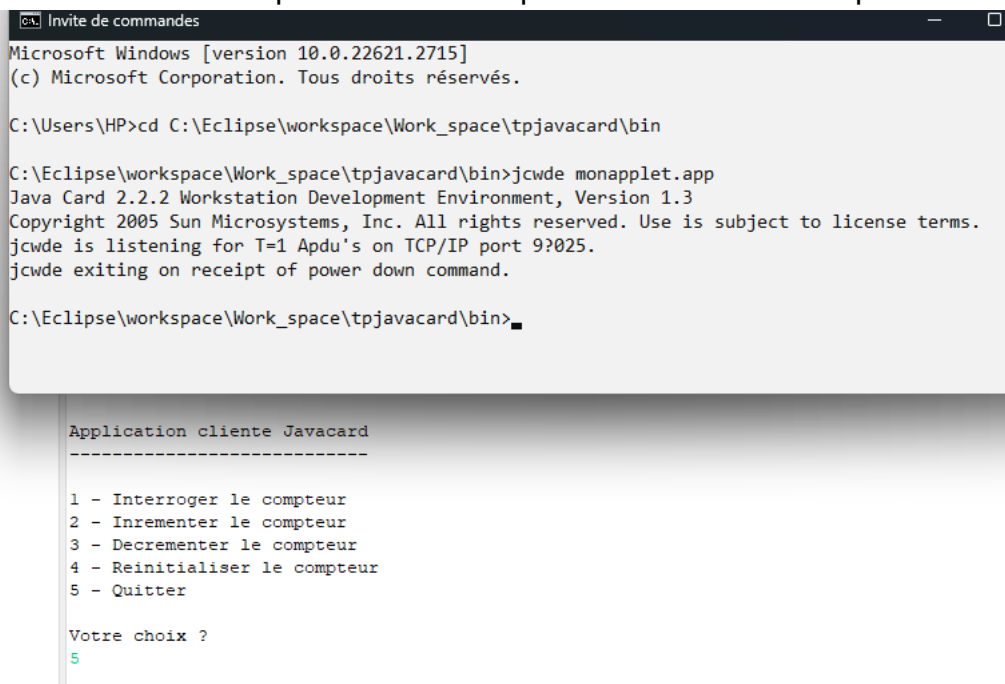
Votre choix ?
3
OK

Application cliente Javacard
-----

1 - Interroger le compteur
2 - Inrementer le compteur
3 - Decrementer le compteur
4 - Reinitialiser le compteur
5 - Quitter

Votre choix ?
1
Valeur du compteur : 2
```

- En Quittant l'application cliente (commande 5). On peut voir que le simulateur se termine automatiquement à la réception de la commande "powerdown" :



```
Invite de commandes
Microsoft Windows [version 10.0.22621.2715]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HP>cd C:\Eclipse\workspace\Work_space\tpjavacard\bin

C:\Eclipse\workspace\Work_space\tpjavacard\bin>jcwde monapplet.app
Java Card 2.2.2 Workstation Development Environment, Version 1.3
Copyright 2005 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
jcwde is listening for T=1 Adu's on TCP/IP port 9025.
jcwde exiting on receipt of power down command.

C:\Eclipse\workspace\Work_space\tpjavacard\bin>


Application cliente Javacard
-----

1 - Interroger le compteur
2 - Inrementer le compteur
3 - Decrementer le compteur
4 - Reinitialiser le compteur
5 - Quitter

Votre choix ?
5
```

# Interface Graphique

## 1-Insertion du code PIN :

 AUTOMATED TELLER MACHINE

— □ ×

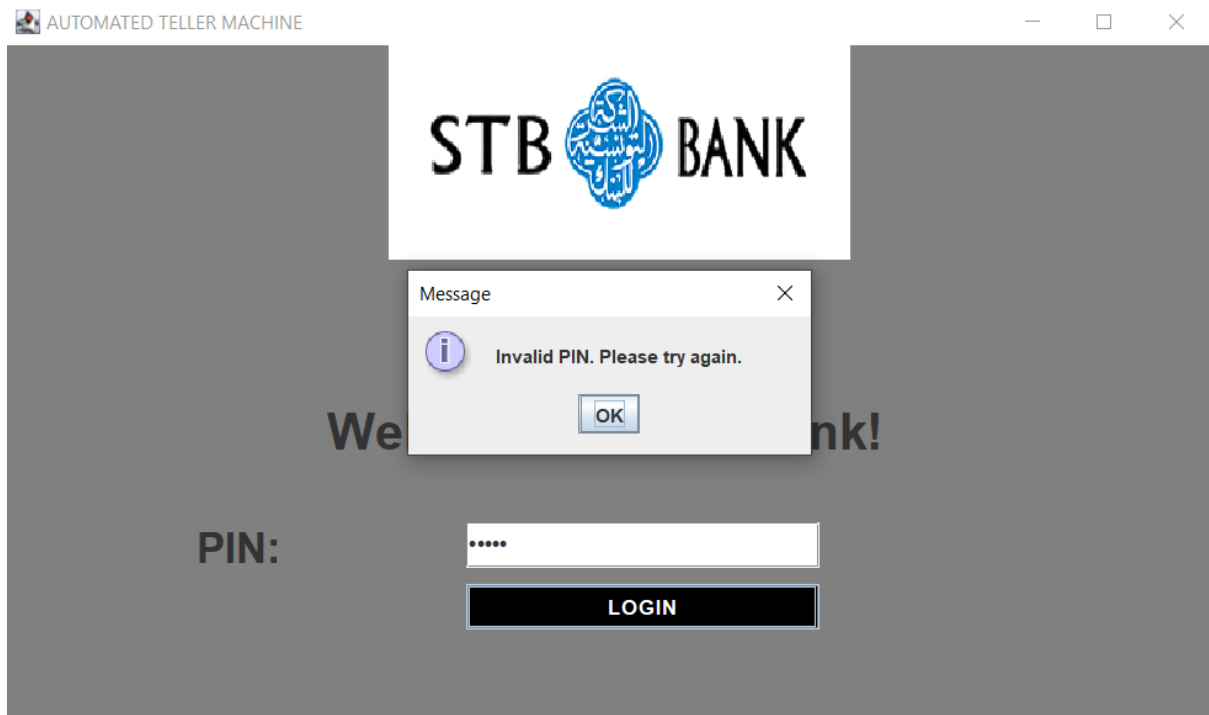


**Welcome to STB-Bank!**

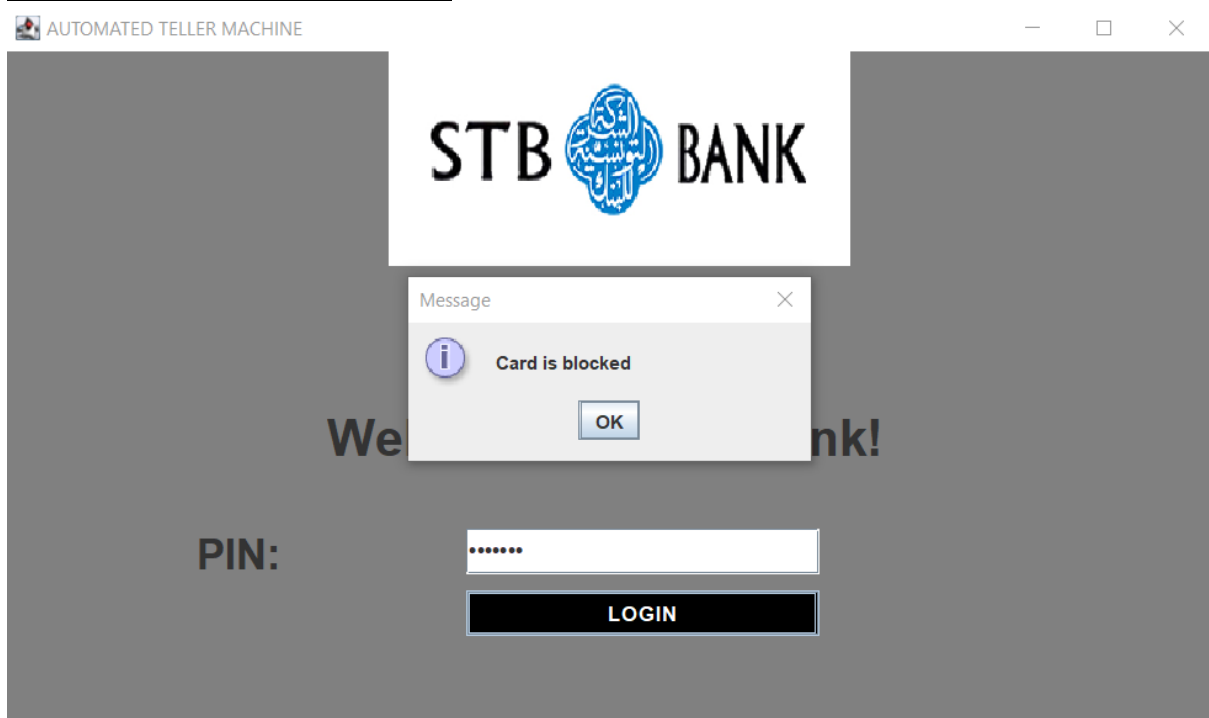
**PIN:**

**LOGIN**

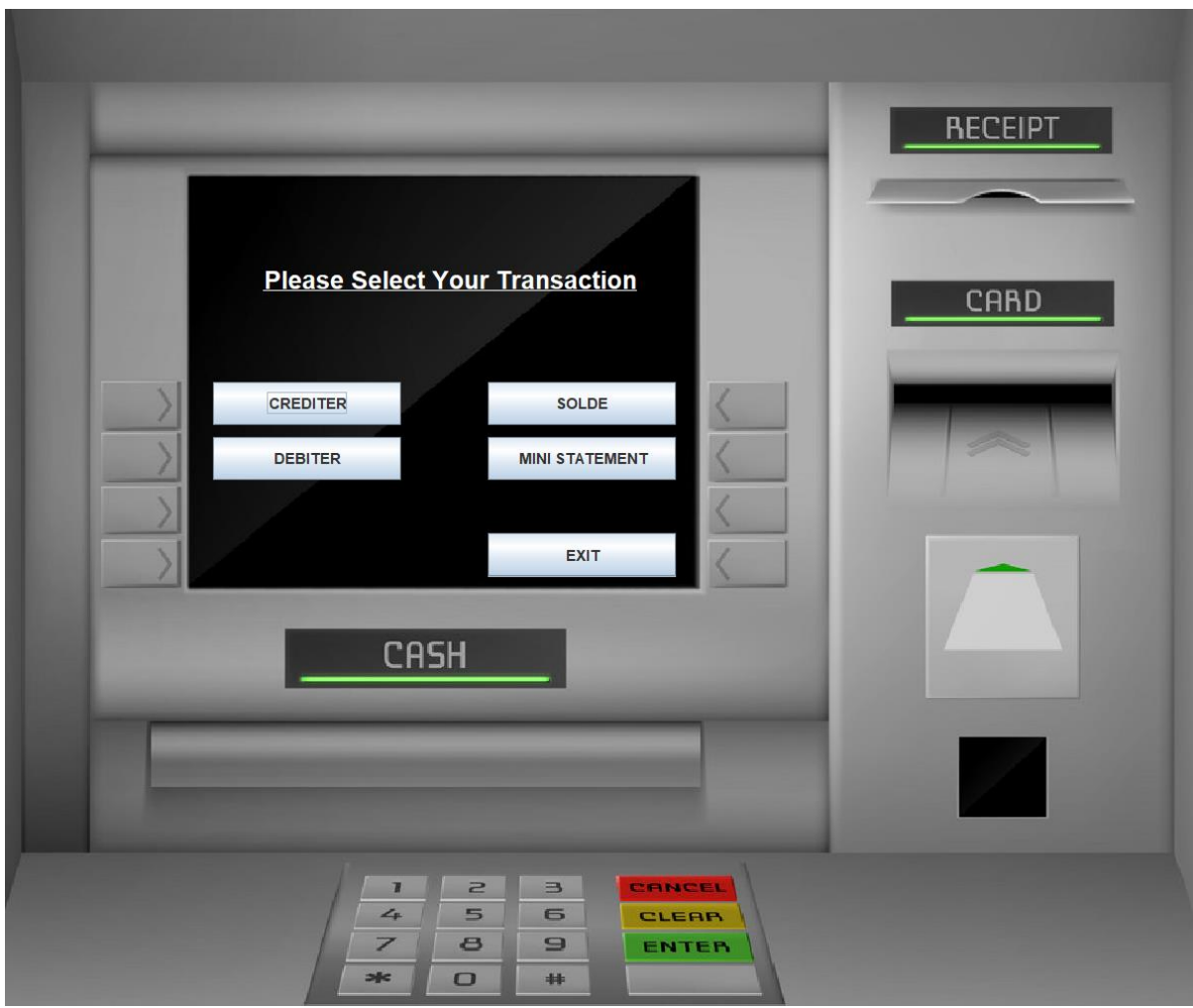
## 1.1 Cas d'échec N°1 : code dépassant 4 chiffres



## 1.2 Cas d'échec N°2 : plus que 3 tentatives incorrectes



## 2-Interface des transactions :



### 3-Affichage du solde :



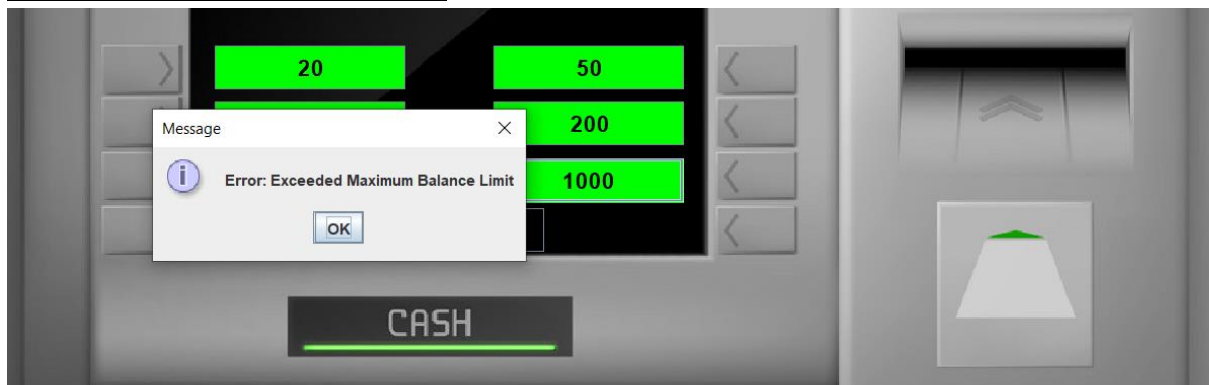
#### 4-Choisir le montant à créditer :



#### **4.1 Créditer cas N°1 : succès**



#### **4.2 Créditer cas N°2 :excéder le montant maximal**



## 6\_Choisir le montant à débiter :

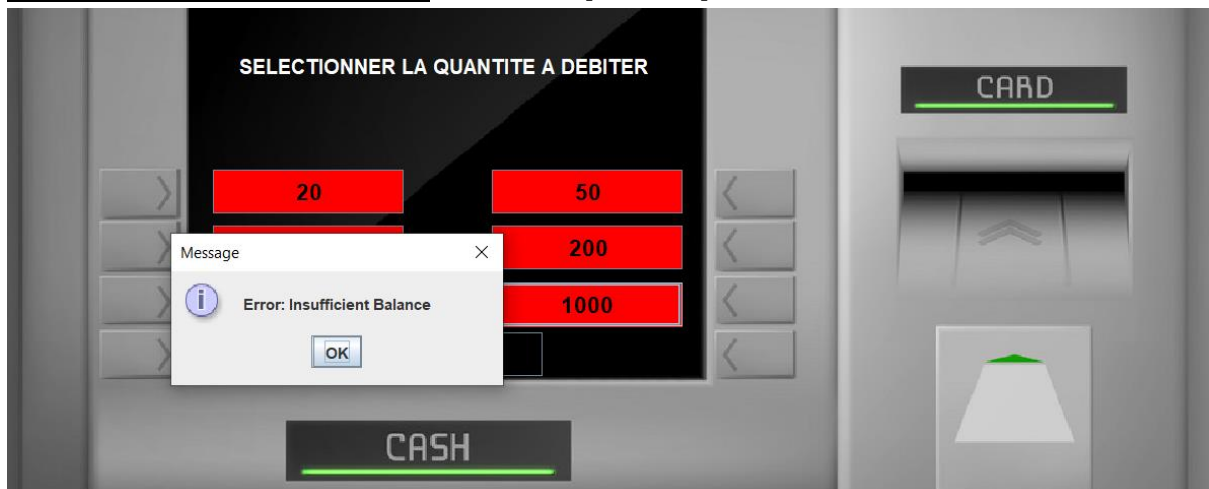




## 6.1 Débiter cas N°1 : succès



## 6.2 Débiter cas N°2 :débiter plus que le solde



## **7 Mini Statement : contient l'historique de toutes les transactions affectées**



Transaction History

— □ ×

Date	Hour	Transaction	Amount
02/12/2023	23:21	Credit	1000
02/12/2023	23:21	Credit	1000
02/12/2023	23:21	Credit	1000
02/12/2023	23:21	Credit	1000
02/12/2023	23:23	Debit	50
02/12/2023	23:23	Debit	1000
02/12/2023	23:23	Debit	1000
02/12/2023	23:23	Debit	1000
02/12/2023	23:23	Debit	1000

# Sommaire

<u>Environnement du développement Java Card</u>	<u>p1</u>
<u>Développement d'une application coté serveur</u>	<u>p20</u>
<u>Programmation d'une application coté client</u>	<u>p33</u>
<u>Interface Graphique</u>	<u>p43</u>