Tutorial JavaCard A

Tutorial JavaCard A – Notes

JavaCard applets are typically compiled for **Java 1.2 to 1.5 bytecode**, depending on the **JavaCard API version** and target card capabilities. However, **Java 1.5 is very old**, and most modern JDKs no longer support generating class files in that bytecode version.

✅ Last Java SDK Supporting Java 1.5

The **latest JDK that officially supports** `-target 1.5` is:

> **JDK 8 (Oracle or OpenJDK 8)**

- Even though Java 8 defaults to `-target 1.8`, you can still compile for **1.5** with:

bash Copiar Editar

```
javac -source 1.5 -target 1.5 MyApplet.java
```

- You may need to also set the `-bootclasspath` to point to the **JavaCard API** classes, to avoid linking with standard Java SE libraries.

Tutorial JavaCard A – Notes

✅ Recommended Setup

- **Use JDK 8 (e.g., OpenJDK 1.8)** for compiling JavaCard applets.
- Make sure to include JavaCard's `api_classic.jar` (from JCDK) in your `-bootclasspath` to restrict usage to classes available in the JavaCard runtime.

🧪 Example Compilation Command

bash Copiar Editar

```
javac -source 1.5 -target 1.5 \
  -bootclasspath /path/to/javacard/framework/api_classic.jar \
  -classpath /path/to/javacard/framework/api_classic.jar \
  MyApplet.java
```

Tutorial JavaCard A – Notes

✅ Opção 2: Instalar múltiplas versões com o SDKMAN (recomendado para devs)

Se quiser gerir várias versões do JDK (ex: 8, 11, 17, 22), o SDKMAN é a ferramenta ideal.

1. Instalar o SDKMAN

```bash
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
```

2. Instalar o JDK 8

```bash
sdk install java 8.0.392-tem
```

3. Tornar o JDK 8 a versão ativa

```bash
sdk use java 8.0.392-tem
```

Ou para definir como padrão:

```bash
sdk default java 8.0.392-tem
```

4. Verificar

```bash
java -version
```

Tutorial JavaCard A – Notes

O jCardSim , sendo uma **implementação simulada da JavaCard Runtime Environment (JCVM)**, tem um suporte mais flexível do que um microcontrolador JavaCard real — mas ainda **segue as restrições da plataforma JavaCard em termos de bytecode e APIs suportadas**

✅ Versão de bytecode Java suportada pelo jCardSim

O jCardSim suporta classes compiladas para bytecode Java 1.2 e 1.3, até um subconjunto do Java 1.5, dependendo da versão utilizada:

Recurso	Suporte no jCardSim
Bytecode Java	Java 1.2 / 1.3 / parcialmente 1.5
APIs JavaCard	JavaCard 2.2.1, 2.2.2, 3.0.1 (parcial)
Compilador recomendado	`javac -source 1.5 -target 1.2` (ou 1.5)
JDK compatível para compilação	JDK 6 a JDK 8

⚠️ Mesmo que o `javac` gere bytecode Java 1.5, a VM do jCardSim só interpreta instruções do subconjunto aceito pela especificação JavaCard, que é muito limitado (sem `float`, `double`, `String`, `Thread`, etc.).

Tutorial JavaCard A – Notes

Importante: *Compilar com o API da JavaCard*

- Se usar qualquer classe que esteja no Java SE e não no JavaCard (ex: java.lang.String, java.util.*), o applet não funcionará no jCardSim nem num smartcard.

✅ **Solução: compilar com o `-bootclasspath` da JavaCard**

bash

```
javac -source 1.5 -target 1.5 \
  -bootclasspath /path/to/javacard/framework/api_classic.jar \
  -classpath /path/to/javacard/framework/api_classic.jar \
  -d build/classes \
  src/com/example/MyApplet.java
```

Tutorial JavaCard A – Links

- Video - https://www.youtube.com/watch?v=oj2l0X8D4y0&hd=1
- Java Card Development Kit: https://www.oracle.com/java/technologies/javacard-downloads.html#archive
- JCardSim: http://jcardsim.org/docs/quick-start-guide-using-in-cli-mode
- Windows JavaCOS tool: https://javacardos.com/javacardforum/viewforum.php?f=26
- GP.exe (GlobalPlatformPro) for installing/deleting apps & general stuff:
  - https://github.com/martinpaljak/GlobalPlatformPro

Exercícios

1. Descarregar o Javacard SDK 2.2.2 do site da Oracle (http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javame-419430.html)
2. Na pasta "**lib"** dos entregáveis desta semana no Moodle encontra o .jar do simulador JcardSim http://jcardsim.org/
   - Estudar o exemplo de uso do simulador em CLI mode: http://jcardsim.org/docs/quick-start-guide-using-in-cli-mode
3. Na pasta "**src**" dos entregáveis desta semana no Moodle existe código fonte para alguns applets javacard que vai poder experimentar em modo simulação e modificar:
   - Estude o código do Applet "Echo". Compile o applet com "javac", e experimente correr o bytecode da classe no simulador "**JcardSim**".
     - Modifique o applet de forma a que ele mantenha o número de APDU processadas e devolva no R-APDU o complemento binário dos dados que recebe.
     - (Use o operador Java ´~´" ou então XOR de cada byte com =0XFF)
   1. Analise o código fonte do applet **Wallet**. Compile o applet, e experimente correr o código no simulador "**JcardSim**", mudando os APDU de exemplo de forma a compreender melhor como se define o PIN no momento da instalação e verificar se as operações que afetam o "saldo" se processam da forma correta. Verifique o que acontece se fornecer o PIN de forma errada várias vezes seguidas. De seguida modifique o código do applet para:
      - Que este aceite como argumentos de inicialização um **PUK (Personal Unblocking Key)** de desbloqueio e um valor inicial para o Saldo.
      - Mudar o construtor da classe Wallet para inicializar um PUK e Balanço a partir dos argumentos de inicialização.
      - Definir um APDU e implementar o método necessário para desbloquear um Wallet bloqueado por falhas de verificação do PIN através do fornecimento do **PUK**