

Student Contest Card

Student: Boghiu Georgiana-Viorica

Grupa: A3

An: III

Structura

Aplicatia este constituita din doua module: modulul Java Card care implementeaza cardul studentului si modulul Java care implementeaza Terminalul din care se poate accesa cardul.

Cardul studentului poate fi accesat din Terminal si suporta urmatoarele operatii:

1. Inregistrarea unei discipline pe card
2. Inregistrarea punctajului de la un concurs pe card
3. Interogarea in vederea obtinerii datelor de pe card ale unei discipline
4. Echivalarea unei note la o disciplina pe baza punctajului obtinut la concursul disciplinei
5. Interogarea in vederea obtinerii punctajelor la toate disciplinele de pe card

Terminalul este intermediarul intre student/secretariat/profesori si cardul studentului. Din el se pot efectua urmatoarele operatii:

1. Autentificarea studentului
2. Verificarea pin-ului
3. Inregistrarea punctajului la concurs a studentului de catre comisie
4. Echivalarea notei la o disciplina de catre profesor
5. Actualizarea prioritatii pentru alegerea profesorului de licenta
6. Afisarea situatiei scolare a studentului

Baza de date este alcatuita din 3 fisiere in format CSV:

1. Baza_date.csv
2. Studenti.csv
3. discipline.csv

Logica aplicatiei

Studentul se va identifica prin ID. El va introduce codul operatiunii pe care doreste sa o efectueze:

1. Inregistrare punctaj
2. Echivalare nota
3. Actualizare
4. Afisare situatie scolara
5. Iesire

Cod 1: Inregistrare punctaj

Studentul se autentifica prin PIN.

Comisia va alege codul concursului din lista de concursuri afisate. El va introduce punctajul obtinut de student, care va fi apoi memorat pe card.

Cod 2: Echivalare nota

Studentul se autentifica prin PIN.

Profesorul alege codul disciplinei din lista de discipline afisate. Daca punctajul obtinut de student la concurs este ≥ 80 atunci nota va fi actualizata la 10.

Cod 3: Actualizare

Studentul se autentifica prin PIN.

Daca cel putin 3 dintre punctajele la concursurile pentru discipline sunt ≥ 80 , atunci se va actualiza prioritatea studentului la 1.

Cod 4: Afisare situatie scolara

Studentul se autentifica prin PIN. Daca PIN-ul este corect, se va afisa numele, id-ul, prioritatea si detaliile la fiecare disciplina ale studentului.

Detalii de implementare

Cardul studentului:

CLA: 0x80

Formatul comenzilor suportate de cardul studentului este urmatoarul:

1. Inregistrarea unei discipline pe card:

INS: 0x64 P1: 0x00 P2: 0x00 Lc: 0x06

Buffer: Id_disciplina Nota Zi Luna An Cod_concurs

Le: 0x7F

Exemplu pentru id=1, nota=10, zi=13, luna=7, an=22, cod_concurs=74 :

0x80 0x64 0x00 0x00 0x06 0x01 0x0A 0x0D 0x07 0x16 0x4A 0x7F

2. Inregistrarea punctajului de la un concurs pe card

INS: 0x65 P1: 0x00 P2: 0x00 Lc: 0x02

Buffer: Cod_concurs Punctaj

Le: 0x7F

Exemplu pentru cod_concurs=74 punctaj=90:

0x80 0x65 0x00 0x00 0x02 0x4A 0x5A 0x7F

3. Interogarea in vederea obtinerii datelor de pe card ale unei discipline

INS: 0x63 P1: 0x00 P2: 0x00 Lc: 0x01

Buffer: Id_disciplina

Le: 0x7F

Exemplu pentru id_disciplina=2:

0x80 0x63 0x00 0x00 0x01 0x02 0x7F

Raspuns Le: 0x06 Nota Zi Luna An Cod_concurs Punctaj

Daca interogam cardul pentru disciplina cu id=1, iar datele stocate pe el pentru disciplina respectiva sunt nota=10, zi=13, luna=7, an=22, cod_concurs=74, punctaj =90 atunci raspunsul primit va fi:

0x80 0x63 0x00 0x00 0x01 0x01 0x06 0x0A 0x0D 0x07 0x16 0x4A 0x5A 0x90 0x00

4. Echivalarea unei note la o disciplina pe baza punctajului obtinut la concursul disciplinei

INS: 0x66 P1: 0x00 P2: 0x00 Lc: 0x01

Buffer: 0xId_disciplina

Le: 0x7F

Exemplu pentru id_disciplina=2:

0x80 0x66 0x00 0x00 0x01 0x02 0x7F

5. Interogarea in vederea obtinerii punctajelor la toate disciplinele de pe card

INS: 0x62 P1: 0x00 P2: 0x00 Lc: 0x00

Buffer: null

Le: 0x7F

Exemplu interogare:

0x80 0x62 0x00 0x00 0x00 0x7F

Raspunsul Le: 0x05 punctaj1 punctaj2 punctaj3 punctaj4 punctaj5

Daca interogam cardul pentru obtinerea punctajelor la toate disciplinele, iar datele stocate pe el sunt: punctaj1=30, punctaj2=50, punctaj3=80, punctaj4=98, punctaj5=82, atunci raspunsul primit va fi:

0x80 0x62 0x00 0x00 0x00 0x05 0x1E 0x32 0x50 0x62 0x52 0x90 0x00

Terminalul:

Conexiune la simulator se face din Terminal folosind com.sun.javacard.apduio..

Aplicatia de Terminal este alcatuita din clasele:

1. Command
2. Wallet
3. Student
4. Disciplina
5. Convert

Command

Poate fi initializat cu un vector `byte[]` sau un `String`, continand comanda ce dorim sa o trimitem la simulator.

Se ocupa de setarea campurilor pentru o instanta `Apdu` care urmeaza a fi trimisa la simulator.

Wallet

Se ocupa de parsarea si trimiterea fisierului `cap-Wallet.script` din aplicatie la simulator.

Pregateste un wallet pentru userul identificat prin ID .

Pentru studentul autentificat cu ID si PIN, se initializeaza o instanta a clasei `Student` cu datele preluate din fisierele CSV.

Verifica PIN-ul introdus de student si citeste informatiile necesare apelarii comenzilor `apdu`. Valideaza datele introduse de utilizator, inainte de a trimite comenzile `apdu` la simulator.

Pasi efectuati in instantierea clasei `Wallet`:

-> Parsarea si trimiterea la simulator a fisierului `cap-Wallet.script`

`setApdu(); sendCAPWallet()`

-> Crearea , selectarea wallet-ului si autentificarea cu PIN. PIN-ul va avea forma `ID%10,(ID+1)%10,(ID+2)%10,(ID+3)%10`. Daca un student are id-ul 5 , atunci PIN-ul lui va fi 5678.

Se creaza o instanta a clasei `Student` care se initializeaza cu date din baza de date ale studentului cu ID-ul dat. Disciplinele si datele lor vor fi inregistrate cate una pe card cu ajutorul comenzii `SET_DATA`. Astfel, cardul este pregatit de operatiuni.

`createWallet(), selectWallet(), setWallet(), setData()`

Interactiunea cu utilizatorul:

-> citirea pinului de la tastatura: *`getPin()`*

-> citirea de la tastatura a datelor pentru inregistrarea punctajului pe card:

`registerKeyboard()`

Interactiunea cu simulatorul;

-> constructia comenzilor si trimiterea lor la simulator: *`verifyPin(), register(), update(), get_data(), getAllPoints(), setData()`*

Functii auxiliare:

-> modelarea comenzilor:

`check()` verifica daca raspunsul de la simulator este `SW1:90 SW2: 00`,

`concat()` concateneaza doi vectori de tip `byte[]` , este folosita in construirea comenzilor `Apdu` parametrizate,

getHexString() folosita pentru debug, afiseaza un vector de tip byte[] in reprezentarea hexadecimalea.

Student

Retine campurile id, nume, prenume, prioritate, discipline (lista cu disciplinele studentului). Seteaza prioritatea in baza de date cu functia *setPrioritate()*.

Disciplina

Retine datele pentru o disciplina: student_id, id_disciplina, denumire, nota, zi, luna, an, cod_concurs. Citeste din fisierul CSV codurile concursurilor si actualizeaz nota in baza de date.

Convert

Clasa auxiliara folosita pentru a facilita conversiile intre String , Hex si Int.

Testarea

Testarea comenzilor de pe SmartCard s-a efectuat cu scriptul urmator:

```
// create wallet applet
0x80 0xB8 0x00 0x00 0x14 0x0a 0xa0 0x0 0x0 0x0 0x62 0x3 0x1 0xc 0x6 0x1 0x08 0x0
0x0 0x05 0x01 0x02 0x03 0x04 0x05 0x7F;
```

```
////////////////////////////////////
// Initialize Wallet
////////////////////////////////////
```

```
//Select Wallet
0x00 0xA4 0x04 0x00 0x0a 0xa0 0x0 0x0 0x0 0x62 0x3 0x1 0xc 0x6 0x1 0x7F;
// 90 00 = SW_NO_ERROR
```

```
//Verify user pin
0x80 0x20 0x00 0x00 0x05 0x01 0x02 0x03 0x04 0x05 0x7F;
//90 00 = SW_NO_ERROR
```

```
//set data 1
0x80 0x64 0x00 0x00 0x06 0x01 0x08 0x09 0x02 0x22 0x10 0x7F;
```

```
//set data 2
0x80 0x64 0x00 0x00 0x06 0x02 0x07 0x0D 0x02 0x22 0x20 0x7F;
```

```
//set data 3  
0x80 0x64 0x00 0x00 0x06 0x03 0x05 0x0E 0x02 0x22 0x30 0x7F;
```

```
//set data 4  
0x80 0x64 0x00 0x00 0x06 0x04 0x0A 0x0A 0x02 0x22 0x40 0x7F;
```

```
//set data 5  
0x80 0x64 0x00 0x00 0x06 0x05 0x07 0x08 0x02 0x22 0x50 0x7F;
```

```
//get data id_disciplina = 01  
0x80 0x63 0x00 0x00 0x01 0x03 0x7F;
```

```
//register id=1 punctaj=81 (51 HEX)  
0x80 0x65 0x00 0x00 0x02 0x10 0x51 0x7F;
```

```
//update id=2 nota=10  
0x80 0x66 0x00 0x00 0x01 0x02 0x7F;
```

```
//get all  
0x80 0x62 0x00 0x00 0x00 0x7F;
```

Pentru testarea Terminalului s-au urmat mai multe scenarii si s-au observat modificarile din baza de date si a situatiei scolare a studentului.

Exemple de scenarii:

1. Pentru un student cu 2 punctaje ≥ 80 , actualizarea prioritatii nu se va efectua.
2. Pentru un student care nu introduce corect PIN-ul cardului, nu se va efectua operatia selectata.
3. Daca studentul are 3 punctaje = 80, actualizarea prioritatii se va efectua.
4. Pentru un student cu un punctaj < 80 , nu se va echivala nota la materia respectiva cu 10.