

# Structuri de date (Seria CB)

## Tema 1 — ATM

Responsabili temă	Daniel Hinoveanu, Florin Iancu, Andreea Mitran
Data publicării	<b>07.03.2019</b>
Termenul de predare	<b>27.03.2019 (ora 23:55)</b>  Se acceptă teme trimise cu penalizare de 10 puncte / zi (din maxim 100 puncte) până la data de <b>30.03.2019 (ora 23:55)</b>
Versiune document	<b>2</b>

### 1. Introducere

În desfășurarea activităților unei bănci este necesară buna gestionare a cerințelor clienților, administrarea conturilor acestora, precum și menținerea unei evidențe a istoricului tranzacțiilor desfășurate. Toate acestea sunt în responsabilitatea unui sistem ce are rolul de a superviza baza de date și de a procesa cererile utilizatorilor. Aceste cereri sunt primite adesea prin intermediul unui ATM, ce oferă clientului posibilitatea de a efectua tranzacții.

### 2. Cerință

Tema presupune să realizați un program care să simuleze funcționalitatea unui ATM. Pentru acest lucru, este necesară cunoașterea și păstrarea datelor cardurilor deținute de clienți. Funcționalitățile pe care le va avea ATM-ul și detaliile de implementare sunt descrise în secțiunile următoare.

### 3. Implementare

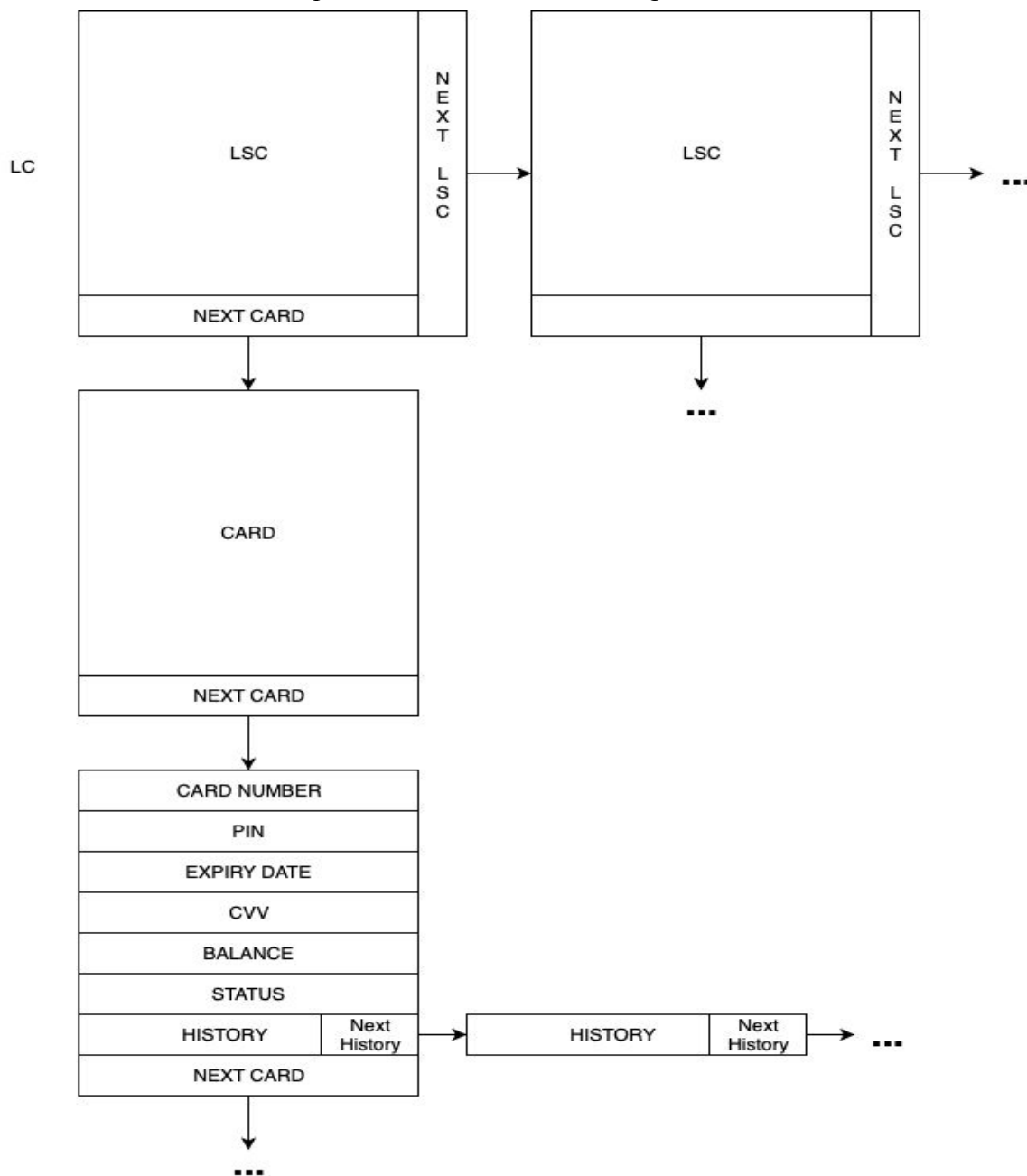
Gestionarea datelor despre carduri se va realiza în baza de date cu structura următoare:

- va exista o **listă de carduri (LC)**, care conține toate cardurile existente. Lista cardurilor este împărțită în submulțimi de carduri, submulțime creată pe baza numărului cardului. Fiecare submulțime de carduri este stocată într-un element al listei de carduri, sub forma

**unei noi liste (LSC).** Fiecare element din lista LSC reține datele despre carduri. Asocierea cardului corespunzător unei celule din lista LC este făcută pe baza numărului cardului;

- fiecare element din **lista LSC** (adică fiecare card) va avea de asemenea asociată o **listă** în care se va reține fiecare tranzacție desfășurată cu acel card (istoricul tranzațiilor unui card). Inserarea unui nou card într-o listă se va realiza mereu **la începutul listei** și de asemenea, tranzacțiile unui card vor fi adăugate în lista istoricului acestuia **la începutul listei**.

Această schemă de implementare este descrisă în figura următoare:



## 4. Funcționalitate

Se consideră două tipuri de utilizatori: **client** (persoana care interacționează cu bancomatul) și **administrator** (bancomatul în sine).

- **Operații specifice administratorilor:**

- a) **Adăugare card în baza de date**

```
add_card <CARD_NUMBER> <PIN> <EXPIRY_DATE> <CVV>
```

Creează cardul cu datele specificate și îl adaugă în baza de date. În cazul în care cardul cu numărul `<CARD_NUMBER>` există deja în baza de date, este returnat mesajul `The card already exists`.

Adăugarea unui card în lista LC se va realiza în celula aflată pe poziția **poz**, în care poziția se calculează conform următoarei formule după formula următoare:

```
poz = suma % (nr_max_carduri)
```

unde `suma` reprezintă suma cifrelor numărului unui card, `nr_max_carduri` este un număr ce se va găsi pe prima linie a fișierului de input.

- b) **Ștergere card din baza de date**

```
delete_card <CARD_NUMBER>
```

Cardul este eliminat din baza de date.

- c) **Anularea unei tranzacții**

```
reverse_transaction <CARD_NUMBER_SOURCE> <CARD_NUMBER_DEST> <SUM>
```

Este posibil ca proprietarul cardului sursă să introducă greșit cardul către care dorește să transfere bani. În acest caz, el poate face o cerere către administrator ca acesta să anuleze tranzacția; astfel, operația retrage `SUM` lei de pe cardul `CARD_NUMBER_DEST` și îi pune înapoi pe cardul `CARD_NUMBER_SOURCE`. Dacă proprietarul cardului destinație nu are fonduri suficiente pe card pentru a se putea realiza operația de anulare a tranzacției, atunci va fi returnat mesajul `The transaction cannot be reversed`. Altfel, tranzacția este anulată și este eliminată din istoricul cardului destinație (astfel încât acesta nu va mai vedea niciodată ca aceasta a existat).

Operația `reverse_transaction <CARD_NUMBER_SOURCE> <CARD_NUMBER_DEST> <SUM>` este apoi adăugată în istoricul cardului sursă; astfel poate fi observată anularea tranzacției.

Astfel, se vor modifica și sumele de pe ambele carduri implicate în anularea unei tranzacții.

#### d) Deblocare card

`unlock_card <CARD_NUMBER>`

Operația deblochează cardul `CARD_NUMBER`. Dacă se încearcă operația de deblocare a cardului, deși acesta nu a fost blocat, se va reseta numărul de încercări de introducere a PIN-ului. Ca și exemplu, dacă s-a încercat introducerea unui card și a PIN-ul a fost greșit, dacă înainte de încercarea de introducere din nou a acelui card în bancomat, administratorul decide să realizeze operația de deblocare pe acel card, numărul de încercări eșuate nu va mai fi 1, ci se resetează la 0. În urma resetării, câmpul **status** se va actualiza și va deveni `ACTIVE`.

În urma acestei operații, dacă era blocat cardul, câmpul **status** se va actualiza și va deveni `ACTIVE`.

#### e) Afișează card (sau carduri)

`show [<CARD_NUMBER>]`

Dacă parametrul `CARD_NUMBER` nu este prezent, comanda va afișa informațiile despre fiecare card din baza de date. Altfel, vor fi afișate doar informațiile cardului respectiv.

#### f) Redimensionarea bazei de date

Redimensionarea bazei de date se referă la modificarea numărului de carduri asociate unei intrări în LC. Astfel, în cazul în care în baza de date numărul existent total de carduri depășește `nr_max_carduri` (în momentul inserării unui nou card), cardurile existente în baza de date vor fi repositionate în listele (LSC) asociate celulelor din LC. Astfel intrarea celulei din lista LC în care va fi mutat un card va fi calculată considerând formula actualizată conform:

`poz = suma % (2 * nr_max_carduri)`

unde `suma` reprezintă suma cifrelor numărului unui card. Astfel constanta corespunzătoare numărului maxim de carduri admis în baza de date devine `2 * nr_max_carduri`.

### • Operații specifice clienților:

#### a) Inserare card în bancomat

`insert_card <CARD_NUMBER> <PIN>`

În cazul în care cardul încă are pinul inițial, atunci utilizatorului i se va afișa mesajul `You must change your PIN`. Pot fi realizate alte operații asupra cardului și până la modificarea PIN-ului inițial. Totuși, la fiecare realizare a operației de inserare a unui card în bancomat al cărui PIN nu a fost schimbat față de cel inițial, se va afișa mesajul `You must change your PIN`.

Dacă pin-ul este introdus greșit (sau invalid: lungime incorectă, alte caractere), va fi returnat mesajul **Invalid PIN**.

În cazul în care utilizatorul introduce PIN-ul greșit de trei ori consecutiv, atunci cardul va fi blocat și se va afișa mesajul **Invalid PIN** urmat pe linia următoare de mesajul **The card is blocked. Please contact the administrator**. Cardul poate fi deblocat doar de către un administrator. Dacă un card este blocat, operațiile ulterioare pentru acesta vor eșua și vor avea ca rezultat mesajul **The card is blocked. Please contact the administrator**.

Se garantează faptul că după inserarea unui card, **nu** va fi realizată o altă operație de inserare până în momentul în care acesta este retras din bancomat.

**b) Depunere numerar**

**recharge <CARD\_NUMBER> <SUM>**

Operația adaugă **SUM** lei pe cardul utilizatorului. **SUM** trebuie să fie obligatoriu multiplu de 10. Altfel, se va returna mesajul **The added amount must be multiple of 10**.

Dacă operația se execută cu succes, atunci va fi afișat noul sold de pe card.

În ambele cazuri, cardul este deconectat.

**c) Retrager numerar**

**cash\_withdrawal <CARD\_NUMBER> <SUM>**

Dacă **SUM** nu este multiplu de 10, atunci se va returna mesajul **The requested amount must be multiple of 10**. Dacă utilizatorul nu are suficienți bani pe card, atunci va fi returnat mesajul **Insufficient funds**. Dacă operația are succes, atunci va fi returnat soldul rămas.

În ambele cazuri, cardul este deconectat.

**d) Vizualizare sold curent**

**balance\_inquiry <CARD\_NUMBER>**

Va fi afișat soldul curent de pe card. La final, cardul este deconectat.

**e) Tranzacție către alt cont**

**transfer\_funds <CARD\_NUMBER\_SOURCE> <CARD\_NUMBER\_DEST> <SUM>**

Operația îi permite utilizatorului ce deține cardul **CARD\_NUMBER\_SOURCE** să vireze **SUM** lei pe cardul **CARD\_NUMBER\_DEST**. Dacă **SUM** nu este multiplu de 10, se va afișa mesajul **The transferred amount must be multiple of 10**, iar cardul va fi deconectat. Dacă nu

există suficienți bani pe card, transferul nu poate fi efectuat și va fi returnat mesajul **Insufficient funds**, după care cardul este deconectat.

Dacă transferul are succes, va fi returnat soldul rămas pe cardul **CARD\_NUMBER\_SOURCE** și respectivul card este deconectat. De asemenea, această operație va fi adăugată atât în istoricul cardului sursă, cât și în cel al cardului destinație. Operația va fi adăugată în istoricul cardului destinație **doar** dacă se efectuează cu succes.

**f) Schimbare pin**

**pin\_change** <CARD\_NUMBER> <NEW\_PIN>

Dacă noul PIN este nu este alcătuit din exact patru cifre, atunci va fi returnat mesajul **Invalid PIN**; altfel, PIN-ul este modificat.

La final, cardul este deconectat.

**g) Retrager card din bancomat**

**cancel** <CARD\_NUMBER>

Cardul este deconectat și această operație va fi introdusă în istoricul cardului respectiv.

**Alte mențiuni:**

Orice operație efectuată pe un card este adăugată în istoricul cardului. La execuția comenzii **show** de către admin, pentru fiecare card va exista o secțiune în care apar comenzile deja efectuate (exact așa cum se află ele în fișierul de intrare) și, pentru fiecare astfel de comandă, rezultatul acesteia.

Rezultatul unei operații poate fi **SUCCESS** sau **FAIL**.

Starea unui card poate fi **NEW** (nu a fost niciodată folosit, are PIN-ul inițial), **ACTIVE** (cardul a fost utilizat cel puțin odată și poate fi folosit în continuare) sau **LOCKED** (cardul este blocat în urma introducerii eronate a PIN-ului de trei ori consecutiv).

Prima linie din fișierul de intrare conține valoarea **nr\_max\_carduri**.

## 5. Exemplu

Un element în lista în care sunt reținute operațiile efectuate cu un card (istoricul) este de forma:

(**rezultat\_operatie**, **operație**), unde:

- **rezultat\_operatie** reprezintă rezultatul operației efectuate (**SUCCESS** sau **FAIL**)
- **operație** este operația așa cum apare în fișier

Exemplu:

Se consideră următorul exemplu de **fișier de intrare**:

```
7
add_card 1234567891234567 3124 09/19 213
add_card 9876543219876543 1234 08/20 922
show
insert_card 1234567891234567 3124
recharge 1234567891234567 300
show 1234567891234567
insert_card 1234567891234567 3124
pin_change 1234567891234567 2222
insert_card 1234567891234567 2222
balance_inquiry 1234567891234567
show
```

În **fișierul de ieșire** vom avea:

```
pos0: []
pos1: []
pos2: []
pos3: [
(card number: 9876543219876543, PIN: 1234, expiry date: 08/20, CVV: 922, balance: 0, status:
NEW, history: [])
(card number: 1234567891234567, PIN: 3124, expiry date: 09/19, CVV: 213, balance: 0, status:
NEW, history: [])
]
You must change your PIN
300
(card number: 1234567891234567, PIN: 3124, expiry date: 09/19, CVV: 213, balance: 300,
status: NEW, history: [(SUCCESS, recharge 1234567891234567 300), (SUCCESS, insert_card
1234567891234567 3124)])
You must change your PIN
300
pos0: []
pos1: []
pos2: []
pos3: [
```

```
(card number: 9876543219876543, PIN: 1234, expiry date: 08/20, CVV: 922, balance: 0, status: NEW, history: [])
(card number: 1234567891234567, PIN: 3124, expiry date: 09/19, CVV: 213, balance: 300, status: NEW, history: [(SUCCESS, balance_inquiry 1234567891234567), (SUCCESS, insert_card 12345678912345467 2222), (SUCCESS, pin_change 1234567891234567 2222), (SUCCESS, insert_card 1234567891234567 3124), (SUCCESS, recharge 1234567891234567 300), (SUCCESS, insert_card 1234567891234567 3124)])
]
```

Ieșirile comenzilor sunt scrise pe linii separate. Nu există linii libere între output-urile diferitelor comenzi și nici între output-ul pentru diferitele carduri în urma comenzii **show** fără parametri.

## 6. Notare

- **85 puncte** obținute pe testele de pe **vmchecker**;
- **10 puncte: coding style**, codul trebuie să fie comentat, consistent și ușor de citit (a se vedea [1]). De exemplu, tema nu trebuie să conțină:
  - warninguri la compilare;
  - linii mai lungi de 80 de caractere;
  - tab-uri amestecate cu spații;
  - denumire neadecvată a funcțiilor sau a variabilelor;
  - folosirea incorectă de pointeri, neverificarea codurilor de eroare;
  - utilizarea unor metode ce consumă resurse în mod inutil (alocare de memorie);
  - neeliberarea resurselor folosite (eliberare memoriei alocate, ștergerea fișierelor temporare, închiderea fișierelor);
  - alte situații nespecificate aici, dar considerate inadecvate;
- **5 puncte: README** – va conține detalii despre implementarea temei, precum și punctajul obținut la teste (la rularea pe calculatorul propriu)
- **Bonus: 20 puncte** pentru soluțiile ce nu au memory leak-uri (bonusul se va considera numai în cazul în care a fost obținut punctajul aferent testului)
- **Temele care nu compilează, nu rulează sau obțin punctaj 0 la teste, indiferent de motive, vor primi punctaj 0**

## 7. Reguli de trimitere a temelor



- Temele vor fi încărcate pe **vmchecker** (în secțiunea Structuri de Date seria CB: SD-CB), dar și pe **cs.curs.pub.ro**, în secțiunea destinată assignment-ului “Tema1”
- Arhiva finala a temei rezolvate trebuie sa conțină:
  - fișierele sursa
    - Fiecare fișier sursa creat sau modificat trebuie sa înceapă cu un comentariu de forma:  
 /\* **NUME Prenume - grupa** \*/
  - fișierul README în care va fi detaliat modul de implementare al rezolvării
  - fișierul Makefile cu trei reguli (build, run si clean)
    - Fișierul trebuie obligatoriu denumit Makefile și trebuie sa conțină cele 2 reguli menționate
    - Regula **build** va compila sursele și va crea executabilul numit **atm**
    - Regula **run** care rulează executabilul creat
    - Regula **clean** care va șterge executabilele create
  - Arhiva va conține numai fișierele menționate mai sus (nu se accepte fișiere executabile sau obiect)
  - Dacă arhiva nu respecta aceste specificații, aceasta nu va fi acceptata la upload și implicit tema nu va fi luată în considerare
- A se consulta și Regulile generale de trimitere și punctare a temelor [2]

## 8. Referințe

[1] <https://ocw.cs.pub.ro/courses/programare/coding-style>

[2] <http://acs.curs.pub.ro/> - curs SD - seria CB - secțiunea Regulament SD - Reguli de realizare, verificare si trimitere a temelor