**Aplicație pentru gestionarea unei biblioteci**

**Enunț**

Scrieți o aplicație pentru o bibliotecă. Aplicația stochează: cărți, pentru care se rețin ID-ul, titlul, descrierea și autorul, și clienți, pentru care se rețin ID-ul, numele și CNP-ul.

**Funcționalități**

|  |
| --- |
| **Bibliotecă- Lista de funcționalități** |
| F1. Adaugă o nouă carte |
| F2. Adaugă un nou client |
| F3. Actualizează o carte |
| F4. Actualizează un client |
| F5. Șterge o carte din lista de cărți |
| F6. Șterge un client din lista de clienți |
| F7. Caută o carte după id |
| F8. Caută o carte după titlu |
| F9. Caută o carte după autor |
| F10. Caută un client după id |
| F11. Caută un client după nume |
| F12. Caută un client după CNP |
| F13. Închiriere carte |
| F14. Returnare carte |
| F15. Tipărește cele mai închiriate cărți |
| F16. Tipărește clienții cu cărți închiriate ordonat dupa: nume, după numărul de cărți închiriate |
| F17. Tipărește primii 20% dintre cei mai activi clienți (nume client si numărul de cărți închiriate) |
| F18. Tipărește carte |
| F19. Tipărește client |

**Plan de iterații**

|  |  |
| --- | --- |
| **Iteration** | **Planned features** |
| I1 | F1. Adaugă o nouă carte  F2. Adaugă un nou client  F3. Actualizează o carte  F4. Actualizează un client  F18. Tipărește carte  F19.Tipărește client |
| I2 | F5. Șterge o carte din lista de cărți  F6. Șterge un client din lista de clienți  F7. Caută o carte după id  F8.Caută o carte după titlu  F9.Caută o carte după autor  F10. Caută un client după id  F11.Caută un client după nume  F12.Caută un client după CNP |
| I3 | F13. Închiriere carte  F14. Returnare carte  F15. Tipărește cele mai închiriate cărți  F16. Tipărește clienții cu cărți închiriate ordonat dupa: nume, după numărul de cărți închiriate  F17. Tipărește primii 20% dintre cei mai activi clienți (nume client si numărul de cărți închiriate) |

**Scenarii de rulare**

**Scenariu de rulare adaugă o nouă carte**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id carte: | Programul asteapta primirea id-ului cartii care va fi introdusa in lista |
| 23 |  |  |
|  | Introduceti titlul cartii: | Programul asteapta primirea titlului cartii care va fi introdusa in lista |
| Ion |  |  |
|  | Introduceti o scurta descriere a cartii: | Programul asteapta primirea unei scurte descrieri a cartii care va fi introdusa in lista |
| roman social |  |  |
|  | Introduceti autorul cartii: | Programul asteapta primirea autorului cartii care va fi introdusa in lista |
| Liviu Rebreanu |  |  |
|  | Comanda efectuata! | Cartea a fost adaugata in lista |

**Scenariu de rulare adaugă un nou client**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id client: | Programul asteapta primirea id-ului clientului care va fi introdus in lista |
| 23 |  |  |
|  | Introduceti numele clientului: | Programul asteapta primirea numelui clientului care va fi introdus in lista |
| Ion |  |  |
|  | Introduceti CNP-ul clientului: | Programul asteapta primirea cnp-ului clientului care va fi introdus in lista |
| 5041126784653 |  |  |
|  | Comanda efectuata! | Clientul a fost adaugat in lista |

**Scenariu de rulare actualizează o carte**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id pentru cartea careia ii actualizati datele: | Programul asteapta primirea id-ului cartii care va fi actualizata |
| 23 |  |  |
|  | Introduceti titlul cartii: | Programul asteapta primirea titlului cartii care va fi actualizata |
| Ion |  |  |
|  | Introduceti o scurta descriere a cartii: | Programul asteapta primirea unei scurte descrieri a cartii care va fi actualizata |
| roman social, obiectiv, publicat in 1920 |  |  |
|  | Introduceti autorul cartii: | Programul asteapta primirea autorului cartii care va fi actualizata |
| Liviu Rebreanu |  |  |
|  | Comanda efectuata! | Cartea a fost actualizata |

**Scenariu de rulare actualizează un client**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id client: | Programul asteapta primirea id-ului clientului care va fi actualizat |
| 23 |  |  |
|  | Introduceti numele clientului: | Programul asteapta primirea numelui clientului care va fi actualizat |
| Ion Daniel |  |  |
|  | Introduceti CNP-ul clientului: | Programul asteapta primirea cnp-ului clientului care va fi actualizat |
| 5041126784653 |  |  |
|  | Comanda efectuata! | Clientul a fost actualizat |

**Scenariu de rulare tipărește o carte**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Nu exista carti! | Programul afișează un mesaj dacă nu există cărți în listă |
| **Adaugam o carte: 23 Ion roman social Liviu Rebreanu** | | |
|  | 23 Ion roman social Liviu Rebreanu | Programul afișează cărțile existente în listă |

**Scenariu de rulare tipărește un client**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Nu exista clienti! | Programul afișează un mesaj dacă nu există clienți în listă |
| **Adaugam un client: 23 Ion Daniel 5041126784653** | | |
|  | 23 Ion Daniel 5041126784653 | Programul afișează clienții existenți în listă |

**Scenariu de rulare șterge o carte**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id pentru cartea pe care doriti sa o stergeti: | Programul asteapta primirea id-ului cărții care va fi ștearsă |
| 23 |  |  |
|  | element inexistent! | Programul afișează mesaj de eroare dacă elementul nu se află în listă |
| **Adaugam o carte: 23 Ion roman social Liviu Rebreanu** | | |
|  | Introduceti id pentru cartea pe care doriti sa o stergeti: | Programul asteapta primirea id-ului cartii care va fi ștearsă |
| 23 |  |  |
|  | Comanda efectuata! | Cartea a fost ștearsă din listă |

**Scenariu de rulare șterge un client**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id pentru clientul pe care doriti sa il stergeti: | Programul asteapta primirea id-ului clientului care va fi șters |
| 23 |  |  |
|  | element inexistent! | Programul afișează mesaj de eroare dacă elementul nu se află în listă |
| **Adaugam un client: 23 Ion Daniel 5041126784653** | | |
|  | Introduceti id pentru clientul pe care doriti sa il stergeti: | Programul asteapta primirea id-ului clientului care va fi șters |
| 23 |  |  |
|  | Comanda efectuata! | Clientul a fost șters din listă |

**Scenariu de rulare caută carte după id**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id pentru cartea pe care o cautati: | Programul asteapta primirea id-ului cărții căutate |
| 23 |  |  |
|  | element inexistent! | Programul afișează mesaj de eroare dacă elementul nu se află în listă |
| **Adaugam o carte: 23 Ion roman social Liviu Rebreanu** | | |
|  | Introduceti id pentru cartea pe care o cautati: | Programul asteapta primirea id-ului cărții căutate |
| 23 |  |  |
|  | 23 Ion roman social Liviu Rebreanu  Comanda efectuata! | Programul afișează cartea cu id-ul menționat |

**Scenariu de rulare caută carte după titlu**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti titlul sau fragmente din titlul cartii pe care o cautati: | Programul asteapta primirea titlului cărții căutate |
| Ion |  |  |
|  | []  Comanda efectuata! | Programul afișează listă vidă dacă nu există o carte cu titlul menționat |
| **Adaugam o carte: 23 Ion roman social Liviu Rebreanu** | | |
|  | Introduceti titlul sau fragmente din titlul cartii pe care o cautati: | Programul asteapta primirea titlului cărții căutate |
| Ion |  |  |
|  | [[23, 'Ion', 'roman social', 'Liviu Rebreanu']]  Comanda efectuata! | Programul afișează cartea cu titlul menționat |

**Scenariu de rulare caută carte după autor**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti autorul sau o parte din numele autorului cartii pe care o cautati: | Programul asteapta primirea autorului cărții căutate |
| Liviu |  |  |
|  | []  Comanda efectuata! | Programul afișează listă vidă dacă nu există o carte cu autorul menționat |
| **Adaugam o carte: 23 Ion roman social Liviu Rebreanu** | | |
|  | Introduceti autorul sau o parte din numele autorului cartii pe care o cautati: | Programul asteapta primirea autorului cărții căutate |
| Liviu |  |  |
|  | [[23, 'Ion', 'roman social', 'Liviu Rebreanu']]  Comanda efectuata! | Programul afișează cartea cu autorul menționat |

**Scenariu de rulare caută client după id**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti id pentru clientul pe care il cautati: | Programul asteapta primirea id-ului clientului căutat |
| 23 |  |  |
|  | element inexistent! | Programul afișează mesaj de eroare dacă elementul nu se află în listă |
| **Adaugam un client: 23 Ion Daniel 5041126784653** | | |
|  | Introduceti id pentru clientul pe care il cautati: | Programul asteapta primirea id-ului clientului căutat |
| 23 |  |  |
|  | 23 Ion Daniel 5041126784653  Comanda efectuata! | Programul afișează clientul cu id-ul menționat |

**Scenariu de rulare caută client după nume**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti numele sau o parte din numele clientului pe care il cautati: | Programul asteapta primirea numelui clientului căutat |
| Ion |  |  |
|  | []  Comanda efectuata! | Programul afișează listă vidă dacă nu există un client cu numele menționat |
| **Adaugam un client: 23 Ion Daniel 5041126784653** | | |
|  | Introduceti numele sau o parte din numele clientului pe care il cautati: | Programul asteapta primirea numelui clientului căutat |
| Ion |  |  |
|  | [[23, 'Ion Daniel', '5041126784653']]  Comanda efectuata! | Programul afișează clientul cu numele menționat |

**Scenariu de rulare caută client după CNP**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Introduceti CNP-ul sau o parte a CNP-ului clientului pe care il cautati: | Programul asteapta primirea CNP-ului clientului căutat |
| 5041126784653 |  |  |
|  | []  Comanda efectuata! | Programul afișează listă vidă dacă nu există un client cu CNP-ul menționat |
| **Adaugam un client: 23 Ion Daniel 5041126784653** | | |
|  | Introduceti CNP-ul sau o parte a CNP-ului clientului pe care il cautati: | Programul asteapta primirea CNP-ului clientului căutat |
| 5041126784653 |  |  |
|  | [[23, 'Ion Daniel', '5041126784653']]  Comanda efectuata! | Programul afișează clientul cu CNP-ul menționat |

**Scenariu de rulare închiriază o carte**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Cine vrea sa inchirieze o carte?  23 Ion Daniel 5041126784653 | Programul asteapta primirea id-ului clientului care inchiriaza o carte |
| 23 |  |  |
|  | Ce carte doriti sa inchiriati?  23 Ion roman social Liviu Rebreanu | Programul asteapta primirea id-ului cartii care va fi inchiriata |
| 23 |  |  |
|  | Comanda efectuata! | Inchirierea a fost adaugata in lista |
|  | Cine vrea sa inchirieze o carte?  23 Ion Daniel 5041126784653 | Programul asteapta primirea id-ului clientului care inchiriaza o carte |
| 23 |  |  |
|  | Ce carte doriti sa inchiriati?  23 Ion roman social Liviu Rebreanu | Programul asteapta primirea id-ului cartii care va fi inchiriata |
| 23 |  |  |
|  | Cartea nu este disponibila momentan. Va rugam reveniti! | Cartea este deja inchiriata |

**Scenariu de rulare returnează o carte**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Cartea cu id-ul 23 a fost inchiriata de clientul cu id-ul 23  Ce carte doriti sa returnati?  ID carte: | Programul asteapta primirea id-ului cartii care va fi returnata |
| 23 |  |  |
|  | Comanda efectuata! | Cartea a fost returnata, iar inchirierea a fost stearsa din lista |
|  | Nu exista carti inchiriate! | Programul va afisa un mesaj daca nu exista carti inchiriate in lista la momentul actual |

**Scenariu de rulare afișează o închiriere**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Cartea cu id-ul 23 a fost inchiriata de clientul cu id-ul 23  Comanda efectuata! | Programul afiseaza lista cu inchirieri din momentul actual |
| **Dacă returnăm cartea** | | |
|  | Comanda efectuata! | Programul nu contine inchirieri care sa fie afisate |

**Scenariu de rulare afișează cele mai închiriate cărți**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | 23  Comanda efectuata! | Programul afiseaza lista cu cele mai inchiriate carti din momentul actual |
| **Dacă adăugăm o nouă carte: 12 Mara roman Ioan slavici**  **și o nouă închiriere: 23 12** | | |
|  | 23  12 | Programul afiseaza lista cu cele mai inchiriate carti din momentul actual. Dacă numărul de inchirieri este egal între cărți, acestea se vor afișa în ordinea închirierii lor |

**Scenariu de rulare afișează clienții care au închiriat cărți după nume**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | 23 Ion Daniel 5041126784653  Comanda efectuata! | Programul afiseaza lista cu clientii care au carti inchiriate, in ordine alfabetica dupa nume |
| **Dacă adăugăm o nouă carte: 12 Mara roman Ioan slavici,**  **un nou client:** **24 Daniel 5044526784653**  **și o nouă închiriere: 24 12** | | |
|  | 24 Daniel 5044526784653  23 Ion Daniel 5041126784653 | Programul afiseaza lista cu clientii care au carti inchiriate, in ordine alfabetica dupa nume |

**Scenariu de rulare afișează clienții care au închiriat cărți după numărul de cărți închiriate**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | 23 Ion Daniel 5041126784653  Comanda efectuata! | Programul afiseaza lista cu clientii care au inchiriat carti in ordinea descrescatoare a numarului de carti inchiriate |
| **Dacă adăugăm o nouă carte: 12 Mara roman Ioan slavici,**  **un nou client:** **24 Daniel 5044526784653**  **și o nouă închiriere: 24 12** | | |
|  | [‘Ion Daniel’,1]  [‘ Daniel’,1] | Programul afiseaza lista cu clientii care au inchiriat carti in ordinea descrescatoare a numarului de carti inchiriate. Daca au acelasi numar de carti inchiriate se afiseaza in ordinea inchirierii |

**Scenariu de rulare afișează primii 20% dintre cei mai activi clienți**

|  |  |  |
| --- | --- | --- |
| **Utilizator** | **Program** | **Descriere** |
|  | Trebuie sa existe cel putin 5 clienti cu carti inchiriate! | Programul afiseaza mesaj de eroare daca nu exista cel putin 5 clienti care au inchiriat carti |
| **Dacă adăugăm cărțile: 12 Mara roman Ioan Slavici**  **13 Moara cu noroc nuvela Ioan Slavici**  **14 Enigma Otiliei roman Camil Petrescu**  **15 Baltagul Mihail Sadoveanu,**  **Clienții: 11 Dan 5044526984653**  **12 Max 5044735984653**  **13 Gabi 5074535984653**  **14 Gigi 5079254984653**  **și închirieri: 23 12, 11 13, 12 23, 13 14, 14 15** | | |
|  | [‘Ion Daniel’,1]  Comanda efectuata! | Programul afiseaza lista cu primii 20% de client cu cele mai multe carti inchiriate. Daca toti au acelasi numar de carti, se va afisa cel care a inchiriat prima carte |

**Analiza complexitatii functiei de cautare a clientului dupa id**

|  |  |
| --- | --- |
| **Cod functie** | **Analiza complexitatii** |
| def **search\_id**(*self*, key\_client):  *"""*  *Searches a client in list*  *key\_client - int*  *raise Repoexception if we do not have a client with the id mentioned*  *"""*  if key\_client not in *self*.\_\_elems:  raise RepoException(*"element inexistent!\n"*)  for elem in *self*.\_\_elems:  if elem==key\_client:  return elem | Caz favorabil: primul element este cel cautat🡺*Θ(1)*  Caz mediu: exista elementul cautat pe o pozitie oarecare🡺  *Θ(n)*+Σni=1 1=*Θ(n)*+*Θ(n)*=2*Θ(n)*=*Θ(n)*  complexitate liniara  Overall complexity *O(n)* |

Gnome sort (stupid sort)

Gnome Sort este un algoritm de sortare propus inițial de către Dr. Hamid Sarbazi-Azad (Profesor de ingineria calculatoarelor la Sharif University of Technology) în anul 2000. Mai târziu acesta a fost descris de către Dick Grune, dându-i-se numele de Gnome Sort. Este un algoritm de sortare asemănător cu Insertion Sort, doar că etapa de mutare a unui element pe poziția corectă este însoțită de o serie de interschimbări, ca la bubble sort. În principiu, este un algoritm simplu, având complexitatea (O(n^2)), dar tinde spre O(n) dacă lista inițială este aproape sortată.

Gnome Sort este bazat pe metoda de aranjare a unor ghivece de flori a unui gnom. Acesta se uita la ghiveciul din stânga lui și la cel din dreapta lui. Dacă sunt în ordinea corecta, merge mai departe, dacă nu le schimbă și merge mai departe.

**Variante de implementare:**

1) def gnomeSort(list):

pos = 0

while pos < len(list):

if (pos == 0 or list[pos] >= list[pos-1]):

pos += 1

else:

list[pos],list[pos-1]=list[pos-1],list[pos]

pos = pos - 1

2) varianta optimizata

def optimizedGnomeSort(list):

for pos in range(1,len(list)):

gnomeSort(list, pos)

def gnomeSort(list, limitasuperioara):

pos = limitasuperioara

while pos > 0 and list[pos-1] > list[pos]:

list[pos],list[pos-1]=list[pos-1],list[pos]

pos = pos – 1

Exemplu:

|  |  |  |  |
| --- | --- | --- | --- |
| **Current array** | **pos** | **Condition in effect** | **Action to take** |
| [**5**, 3, 2, 4] | 0 | pos == 0 | increment pos |
| [5, **3**, 2, 4] | 1 | a[pos] < a[pos-1] | swap, decrement pos |
| [**3**, 5, 2, 4] | 0 | pos == 0 | increment pos |
| [3, **5**, 2, 4] | 1 | a[pos] ≥ a[pos-1] | increment pos |
| [3, 5, **2**, 4] | 2 | a[pos] < a[pos-1] | swap, decrement pos |
| [3, **2**, 5, 4] | 1 | a[pos] < a[pos-1] | swap, decrement pos |
| [**2**, 3, 5, 4] | 0 | pos == 0 | increment pos |
| [2, **3**, 5, 4] | 1 | a[pos] ≥ a[pos-1] | increment pos |
| [2, 3, **5**, 4] | 2 | a[pos] ≥ a[pos-1] | increment pos: |
| [2, 3, 5, **4**] | 3 | a[pos] < a[pos-1] | swap, decrement pos |
| [2, 3, **4**, 5] | 2 | a[pos] ≥ a[pos-1] | increment pos |
| [2, 3, 4, **5**] | 3 | a[pos] ≥ a[pos-1] | increment pos |
| [2, 3, 4, 5] | 4 | pos == length(a) | finished |