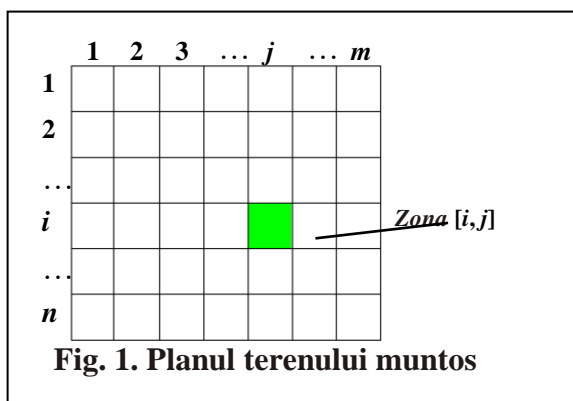


Sarcina 1: Implementați metoda de programare utilizând limbajul C++ care rezolvă următoarea sarcină:

3. Teren muntos. Planul unui teren muntos are forma unei matrice cu $n \times m$ ($n, m \leq 50$) zone (vezi figura 1). Fiecare zonă reală a terenului are o anumită altitudine (înălțime în raport cu nivelul mării).

Informațiile necesare despre zonele terenului considerat sunt înregistrate în următoarele două fișiere:



- fișierul text **Munte.in** conține pe prima linie numerele naturale n și m , separate prin spațiu. Următoarele n linii ale acestui fișier conțin câte m numere reale, separate prin spațiu, fiecare din aceste valori reprezentând altitudinea zonei respective $[i, j]$;
- fișierul text **Roci.in**, fiecare din cele n linii ale căruia conține câte m tipuri ale stratului superior al scoarței terestre muntoase (nisip, podzol, pietris, granit, marmura, bazalt etc.), separate prin câte un spațiu, rocile fiind atribute ale zonelor respective $[i, j]$, $1 \leq i \leq n$, $1 \leq j \leq m$.

Să se creeze o aplicație care să realizeze, la solicitarea utilizatorului, următoarele prescripții:
(Design-ul aplicației vă aparține).

- 1) Înscrie în planul terenului un nou rând (marginal) / o nouă coloană (marginală); poziția rândului (nord/sud) / coloanei (vest/est) de înscris se va preciza de la tastatură;
- 2) Exclde de pe tablă un rând / o coloană; alternativa aleasă și numărul de ordine al rândului / coloanei de exclus se va preciza de la tastatură;
- 3) Determină coordonatele (numărul rândului și numărul coloanei) zonelor de minim local propriu al altitudinilor, adică coordonatele zonelor nemarginale cu altitudine strict mai mică decât fiecare din altitudinile celor maxim 8 zone vecine);
- 4) Determină altitudinea medie a acelor zone din teren, care sunt asociate tipului de rocă introdus de la tastatură;
- 5) Afișează pe ecran lista rocilor distincte din zonele terenului în ordinea ascendentă a numerelor totale de zone ce reprezintă respectivele roci; sortarea datelor se va realiza prin metoda inserției.
- 6) Creează fișierul textual **RociTip.txt**, în care se vor copia acele linii ale fișierului de intrare **Munte.in**, care reprezintă doar zone cu același tip de rocă;
- 7) Determină în rețeaua planului terenului un dreptunghi de arie maximală ce conține doar zone de altitudine nu mai mică ca numărul real **H**; laturile dreptunghiului vor coincide cu laturile zonelor din plan. Numărul **H** se va citi de la tastatură; la ecran se va afișa aria **S**, coordonatele colțurilor stânga –jos și dreapta –sus ale dreptunghiului găsit;
- 8) **Rezolvă problema:** Un alpinist pleacă dintr-o anumită zonă **A**, dată prin coordonatele ei $[x_a, y_a]$ și trebuie să ajungă într-o zonă **B** de maxim absolut **Hmax** în altitudine. Dintr-o zonă, alpinistul se poate deplasa diagonal sau ortogonal într-una din zonele învecinate, doar urcând sau mergând la același nivel. Poate el ajunge într-unul din vârfurile absolute? Dacă da, arătați una din soluțiile problemei.

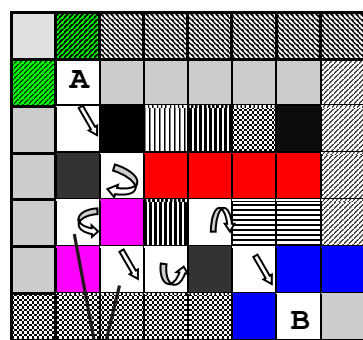


Fig. 2. Zone vizitate. Zonă de maxim absolut

Date de intrare. Numerele naturale x_a și y_a se introduc de la tastatură, iar informațiile despre altitudinile zonelor terenului –din fișierul text **Munte.in**, descris anterior.

Date de ieșire. Se va afișa pe ecran răspunsul (DA/NU) la întrebarea „Poate alpinistul ajunge într-unul din vârfurile absolute?”, și în caz afirmativ – numărul **Hmax** și drumul găsit, reprezentat prin coordonatele zonelor respective.

De exemplu, pentru ilustrarea din figura 2, drumul parcurs se va afișa astfel: $[2, 2] - [3, 2] - [4, 3] - [5, 2] - [6, 3] - [6, 4] - [5, 5] - [6, 6] - [7, 7]$

Sarcina 2: Scrieți un program Java care modelează următoarea activitate:

Creați un produs program pentru evidența activității unui lombard. Lombardul are mai multe filiale în țară. Fiecare filială este caracterizată prin nume, adresă, telefon, etc.

La lombard se adresează clienți cu scopul de a primi o sumă de bani pentru obiecte, lăsate la păstrare. Clienții sunt persoane fizice. Pentru fiecare client programul va conține următoarea informație: nume, prenume, codul personal (IDNP), adresa domiciliu, nr. telefon, alte informații.

Pentru fiecare obiect lăsat în lombard este încheiat un contract, care prevede începutul perioadei de păstrare, costul obiectului (suma plătită clientului), comisionul lombardului, sfârșitul perioadei de păstrare, clientul. Dacă clientul nu returnează suma primită în termenul stabilit, obiectul trece în proprietatea lombardului și poate fi vândut la costul inițial sau la alt preț. Datele despre cazurile de vânzare sunt stocate și afișate la solicitare.

Contractul este perfectat de către agenții de vânzări. Agentul de vânzări este angajat într-o filială. Pentru fiecare agent de vânzări în program se va conține următoarea informație: nume, prenume, codul personal (IDNP), adresa domiciliu, nr. telefon. Produsul program trebuie să permită calculul remunerării agentului de vânzări. Pentru fiecare contract perfectat agentul de vânzări primește un anumit procent din suma totală a contractului.

Cerințe față de sarcină: Implementați concepte ale POO

- Creați clase având ca metode (funcții) citirea și afișarea fiecărui câmp (set-teri și get-teri), precum și o metodă care calculează o valoare nouă a unui sau a mai multor câmpuri (ca exemplu, de convertit prețul din lei în EURO sau de calculat durata deplasării). Adăugați cel puțin trei tipuri de constructori (unul trebuie să fie fără parametri, altul- cu toți parametrii indicați, iar al treilea- cu doar câțiva parametri indicați, în dependență de temă, în ultimul caz unele câmpuri ar putea avea valori egale sau careva valori standard).
- Clase care moștenesc câmpurile și metodele clasei părinte vor avea proprietăți proprii, creând și constructorii respectivi (care să corespundă constructorilor clasei de bază). Folosiți relațiile de asociere și agregare între obiecte (acolo unde este cazul).
- Creați o clasă abstractă și o interfață, care are cel puțin o metodă abstractă, apoi le utilizează în procesul de moștenire și o realizează pe deplin. Fiecare clasă trebuie să conțină constructori supraîncărcați, metoda *toString()* supradefinită
- Prelucrați diferite tipuri de excepție (atunci când avem împărțire la zero, radical din număr negativ, când încercăm să accesăm un element inexistent al unui tablou, atunci când în loc de număr utilizatorul introduce litere, atunci când încercăm să apelăm o metodă a unui obiect care are referință nulă etc.) O metodă care generează excepție proprie.
- Înscrieți obiectele într-o listă, minim 20 de înregistrări. Se va utiliza lista și la afișarea pe ecran.
- Programul să conțină un meniu minimalizat.
- Adăugați clase și / sau membri sau metode la clasele dacă se consideră necesar.