Exercițiul, ca metodă didactică, constă în execuția repetată, conștientă și sistematică a unor operații menite să transforme cunoștințele în deprinderi, facilitând astfel munca independentă și dezvoltarea gândirii critice.

#### 1. Trei caracteristici a metodei

* Repetitivitatea conștientă: Exercițiul presupune efectuarea constantă a unor acțiuni care consolidează cunoștințele, de la recunoașterea unor noțiuni până la aplicarea lor practică.
* Dezvoltarea gândirii critice: Prin identificarea, compararea și eliminarea erorilor, elevii își formează capacitatea de analiză și autoevaluare.
* Aplicabilitatea transversală: Metoda se aplică în orice lecție de informatică, de la verificarea proprietăților grafurilor la rezolvarea problemelor complexe, contribuind la dezvoltarea unui raționament flexibil și operant.

#### 2. Două tipuri de exerciții

1. Exerciții de verificare a proprietăților
   * Exemplu: Verificarea dacă un graf dat este hamiltonian (are un ciclu care trece prin fiecare nod exact o dată).
2. Probleme practice
   * Exemplu: Determinarea unui ciclu hamiltonian într-un graf reprezentând rețeaua de străzi ale unei localități.

Utilizarea metodei

* Mijloace de învățământ: Un mediu de programare pentru implementarea algoritmilor de rezolvare a problemelor; Calculatorul ; Fișă de lucru cu probleme
* Forma de organizare a clasei: Lucru în perechi pentru analiza colaborativă a grafurilor.

#### Scenariul didactic

| Etapă | Activitatea profesorului | Activitatea elevilor |
| --- | --- | --- |
| Introducere | - Definește noțiunea de graf hamiltonian și diferența față de graf eulerian. | - Notează definiția și exemple simple (ex: graf complet, ciclu simplu). |
|  | - Prezintă teorema lui Dirac/Ore (condiții suficiente pentru existența ciclului). | - Discută condițiile și identifică contraexemple. |
| Proiectare | - Desenează pe tablă un graf și cere verificarea proprietății hamiltoniene. | - Încercă să construiască un ciclu hamiltonian pe hârtie. |
|  | - Introduce algoritmul backtracking pentru generarea ciclurilor. | - Notează pașii algoritmului și întreabă despre optimizări. |
| Implementare | - Propune implementarea verificării într-un limbaj de programare. | - Scriu codul (ex: verificare existență ciclu folosind permutări de noduri). |
| Validare | - Furnizează teste: grafuri hamiltoniene/non-hamiltoniene (ex: "casa cu etaj" vs. graf disconex). | - Rulează programul pe teste și corectează erorile. |
| Generalizare | - Leagă de aplicații reale: traseu optim pentru vizitarea punctelor turistice. | - Propun analogii (ex: curierul care trebuie să livreze pachete în mai multe locații). |

#### Corectitudinea științifică

* Terminologie corectă: "ciclu hamiltonian" vs. "drum hamiltonian".
* Algoritmi: Se folosește backtracking sau condiții teoretice (Dirac/Ore), nu metode euristice nejustificate.