PROIECTAREA UNOR LECȚII DIN UNITATEA DE ÎNVĂȚARE „METODE DE SORTARE”

Lecţia este înţeleasă ca o componentă operaţională (Cum?) pe termen scurt a unităţii de învăţare. Dacă unitatea de învăţare oferă înţelegerea procesului din perspectivă strategică, lecţia oferă înţelegerea procesului din perspectivă operativă, tactică.

Nu există un model unic de proiect de lecţie; construirea acestuia depinde de o serie de variabile precum natura conţinutului, obiectivele urmărite, nivelul de pregătire al elevilor, tipul strategiilor didactice utilizate şi cui se adresează.

Proiectarea lecţiei presupune:

- încadrarea lecției, activităţii didactice în sistemul de lecţii/activităţi;
- stabilirea obiectivelor operaționale;
- selectarea, prelucrarea şi adecvarea conţinutului;
- elaborarea strategiei didactice;
- stabilirea metodologiei de evaluare/autoevaluare.

Ca urmare, trecerea de la unitatea de învăţare la o lecţie componentă trebuie să permită o replicare în acelaşi timp funcţională (De ce?), structurală (Cu ce?) şi operaţională (Cum?) a unităţii de învăţare, la o scară temporală mai mică şi într-un mod subordonat.

În tipologia lecţiilor se ia ca principal criteriu de clasificare scopul didactic; pentru disciplina Informatică, pornind de la acest criteriu, se stabilesc următoarele tipuri de lecţii:

- **Lecţia de comunicare şi însuşire de noi cunoştinţe (sau lecție de predare)**, în care concentrarea activităţii didactice se realizează în direcţia dobândirii de către elevi a unor cunoştinţe şi dezvoltării, pe această bază a proceselor şi însuşirilor psihice, a capacităţii instrumentale şi operaţionale. Poate avea următoarele variante: lecţie introductivă (la început de capitol), lecție prelegere, lecţie seminar, lecţie de descoperire pe cale inductivă sau deductivă.
- **Lecţie de formare a priceperilor şi deprinderilor** sau tipul lecţiei de muncă independentă, caracterizată de activitatea independentă a elevilor consacrată rezolvării sarcinilor de învățare în vederea elaborării unor componente acţionale (

priceperi, deprinderi, algoritmi etc). Poate avea următoarele forme: lecţii pe bază de exerciţii aplicative, lecţii în laborator, lecţii de muncă independentă cu ajutorul fişelor.

– **Lecţia de consolidare şi sistematizare**, în care se urmăreşte fixarea şi consolidarea cunoştinţelor prin stabilirea de noi corelaţii între cunoştinţe prin elaborarea unor generalizări mai largi, prin relevarea unor structuri logice între cunoștințe, toate acestea asigurând totodată aprofundarea şi reorganizarea cunoștințelor în jurul unei idei centrale. Poate avea următoarele forme: lecţie de sinteză (încheiere de capitol, sfârşit de semestru sau de an şcolar), lecţie de sinteză prin exerciţii aplicative, lecţie de sinteză combinate cu activitatea de grup.

– **Lecţie de verificare şi apreciere** (de control şi evaluare), prin care se urmărește pe de o parte verificarea bagajului de cunoștințe asimilate, concomitent cu capacitatea de aprofundare, înţelegere şi operare, iar pe de altă parte, măsurarea şi evaluarea celor constatate. Poate avea următoarele forme: lecţii de verificare prin chestionare orală (individual, frontal), lecţii de verificare prin teme scrise (lucrări de control sau semestriale), lecţii destinate analizei lucrărilor scrise (prin relevarea lucrărilor tipice, reprezentative şi elucidarea cauzelor greşelilor sau ale obţinerii succesului), lecţii de verificare şi evaluare prin lucrări practice.

– **Lecţia combinată (mixtă)**, care urmăreşte atât predarea, cât şi fixarea, verificarea şi aplicarea cunoștințelor.

Proiect didactic
Metoda QuickSort
Lecţie mixtă

Unitatea şcolară: ....”
Profil: Real, **Specializarea**: Matematică-Informatică, intensiv Informatică
Disciplina: Informatică,
Profesor: .....
Clasa/Nr. ore săpt.: a IX-a A / 4 ore / săptămână
Anul şcolar: .....

Unitatea de învățare: „Metode sortare”
Tipul lecției: mixtă
Tema: Sortarea prin metoda *QuickSort*

Nivelul iniţial al clasei:

✓ Elevii şi-au însuşit toate noţiunile teoretice despre mecanismul metodei Divide et Impera;

✓ Elevii şi-au însuşit noţiunile despre metodele de sortare de dificultate redusă;

Competențe generale:

✓ Identificarea datelor care intervin într-o problemă şi a relațiilor dintre acestea

✓ Elaborarea algoritmilor de rezolvare a problemelor

✓ Aplicarea algoritmilor fundamentali în prelucrarea datelor

✓ Implementarea algoritmilor într-un limbaj de programare

Competenţe specifice vizate:

4.1. Transcrierea algoritmilor din pseudocod într-un limbaj de programare.

4.2. Identificarea necesităţii structurării datelor în tablouri.

4.3. Prelucrarea datelor structurate.

4.4. Utilizarea unui mediu de programare pentru limbajul C/C++

Obiective educaţionale:

- **Obiective cognitive:**

  Elevii vor putea:

  - ✓ Să definească corect noțiunea de sortare;
  - ✓ Să dovedească trăinicia noţiunilor dobândite la disciplina respectivă, la lecţia curentă;
  - ✓ Să identifice tipurile de probleme ce se pot rezolva prin metode de sortare.

- **Obiective afective:**

  Elevii vor putea:

  - ✓ Să aleagă corect programele care se pot rezolva prin utilizarea tipului tablou unidimensional;
  - ✓ Să aprecieze corect soluţiile oferite de colegi;
  - ✓ Să se implice cu plăcere şi interes la toate etapele lecţiei;
  - ✓ Să se bucure de rezultatele muncii depuse.

- **Obiective psihomotorii:**

  - ✓ Să utilizeze corect noţiunile teoretice însuşite anterior;
  - ✓ Să-şi formeze deprinderi de lucru specifice temei de studiu;
  - ✓ Să-şi dezvolte gândirea logică, capacitatea de generalizare şi problematizare.

Obiective operaţionale:

La sfârşitul lecţiei, elevii vor fi capabili:

O1: Să descrie pașii algoritmului de sortare rapidă;

O2: Să justifice necesitatea utilizării acestui algoritm de sortare;

O3: Să poată aplica metoda pentru un exemplu concret;

O4: Să-şi formeze capacitatea de a modela o problemă practică şi să adapteze algoritmul cunoscut la situaţii noi;

Strategii didactice:

- **Principii didactice:**
  - ✓ principiul participării şi învăţării active;
  - ✓ principiul asigurării progresului gradat al performanţei;
  - ✓ principiul conexiunii inverse.
- **Metode de învăţământ:**
  - ✓ metode de comunicare orală: expunerea, conversaţia, problematizarea.
  - ✓ metode de acţiune: exerciţiul, învăţarea prin descoperire, algoritmizarea.
- **Procedee de instruire:**
  - ✓ explicaţia în etapa de comunicare;
  - ✓ învăţarea prin descoperire, prin rezolvarea de aplicaţii;
  - ✓ conversaţia de consolidare în etapa de fixare a cunoştinţelor.
- **Forme de organizare:** frontală şi individuală**.**
- **Forme de dirijare a învăţării:** dirijată de profesor sau independentă.

Resurse materiale:
Mariana MILOȘESCU: *Informatică, manual pentru clasa a IX-a*, Editura Didactică și pedagogică R.A., Bucureşti, 2007;

Metode de evaluare:
- ✓ evaluare iniţială: întrebări orale;
- ✓ set de aplicaţii.

Timpul acordat: 50 minute

Forme de organizare a activităţii: frontală şi individuală, dirijată de profesor

Metode de evaluare:
- ✓ evaluare iniţială: întrebări orale;
- ✓ set de aplicaţii.

Timpul disponibul: 50 minute.

Forme de organizare a activităţii: frontală şi individuală, dirijată de profesor

Scenariul didactic

Momentul lecţiei	Activitatea profesorului	Activitatea elevului	Durata
1. Moment organizatoric	- Crearea unei atmosfere specifice bunei desfăşurări a activităţii didactice; - Verificarea existenţei resurselor materiale necesare desfăşurării lecţiei - Organizarea şi pregătirea clasei: verificarea frecvenței elevilor	Se pregătesc pentru lecţie	2 minute
2. Verificarea cunoştinţelor şi deprinderilor anterioare dobândite, verificarea temei pentru acasă	Profesorul verifică, prin sondaj, modul în care a fost rezolvată tema pentru acasă şi cum s-a fixat lecţia anterioară, punând următoarele întrebări: 1. Care sunt algoritmii de sortare cu care am lucrat până acum? 2. Ce concluzii am tras în legătură cu aceşti algoritmi (comparare a algoritmilor ca ordin de complexitate din punct de vedere a timpului şi a memoriei ocupate)? Se verifică răspunsurile date şi se aduc completările sau corecţiile necesare.	Ascultă întrebările şi elaborează răspunsurile în concordanţă cu conţinuturile asimilate anterior şi cu experienţa dobândită în aplicarea acestora în exerciţii şi probleme;	10 minute
3. Anunţarea subiectului şi a obiectivelor operaţionale	Anunţă titlul, obiectivele operaţionale şi modul de desfăşurare al lecţiei. Profesorul pune întrebări care să realizeze trecerea spre lecţia nouă realizând comparaţia între algoritmii simpli de sortare învăţaţi şi alţi algoritmi performanţi (*QuickSort*)	Participă activ, Răspund la întrebări Notează în caiete titlul lecţiei	1 minut
4. Transmiterea cunoștințelor	Algoritmul de sortare rapidă utilizează metoda *Divide Et Impera*. Pașii algoritmului sunt: 1. **Se alege o valoare pivot**. Se ia valoarea elementului din mijloc ca valoare pivot, dar poate fi oricare altă valoare, care este în intervalul valorilor sortate. 2. **Partiționare,** Se rearanjează elementele în aşa fel încât, toate elementele care sunt mai mari decât pivotul merg în partea dreaptă a tabloului. Valorile egale cu pivotul pot sta în orice parte a tabloului. În plus, tabloul poate fi împărţit în părţi care nu au aceeaşi dimensiune (nu sunt egale). 3. **Se sortează cele două părți:** se aplică recursiv algoritmul de sortare rapidă în partea stângă şi în partea dreaptă.	Urmăresc prezentarea şi explicaţiile Răspund la întrebări	15 minute

Momentul lecţiei	Activitatea profesorului	Activitatea elevului	Durata
	Algoritmul de partiție în detaliu. Există 2 indici i şi j, şi la începutul algoritmului de partiționare i indică primul element al tabloului iar j indică ultimul element din tablou. La pasul următor algoritmul mută i înainte, până când un element cu o valoare mai mare sau egală cu pivotul este găsită. Indicele j este mutat înapoi, până când un element cu valoare mai mică sau egală cu pivotul este găsită. Dacă i<=j atunci i merge pe poziţia i+1 iar j merge pe poziţia j-1. Algoritmul se opreşte, când i devine mai mare decât j. Algoritm descris în pseudocod: **QuickSort(V,st,dr)** pivot←v[(st+dr) div 2)]; cât timp i<=j execută cât timp v[i] <pivot execută i←i+1; sfârșit cât timp cât timp v[j] >pivot execută j←j-1; sfârșit cât timp dacă i<=j atunci aux←v[i]; v[i] ←v[j]; v[j] ←aux; i←i+1; j←j-1; sfârșit dacă sfârșit cât timp dacă st<j atunci quikSort(v,st,j); sfârșit dacă dacă i<dr atunci quikSort(v,i,dr); sfârșit dacă sfârșit QuickSort	Rezolvă exerciţiile propuse la tablă sau folosind mediul de programare	

Momentul lecţiei	Activitatea profesorului	Activitatea elevului	Durata
5.Fixarea noilor cunoştinţe	Propune elevilor să descrie pas cu pas algoritmul, iar apoi să-l implementeze în limbaj de programare. Un elev va ieși la tablă și va propune soluția. Îndrumă elevii în rezolvarea problemei propuse.	Sunt atenţi la precizările profesorului. Se rezolvă la calculator.	20 min
6. Încheierea lecţiei si propunerea temei pentru acasă	Face aprecieri globale şi individuale cu privire la participarea elevilor la lecţie, notând răspunsurile date în timpul lecţiei. Propune tema pentru acasă. Dă indicaţii pentru rezolvarea temei. Prezintă tema lecţiei viitoare : „Sortarea prin interclasare”.	Primesc fişa cu tema pentru acasă şi notează indicaţiile pofesorului	2 minute