

GHID DE EVALUARE DISCIPLINA INFORMATICĂ

Coordonatori:

prof.univ. dr. Dan Potolea
prof. univ. dr. Ioan Neacșu
prof. univ. dr. Manole Manolescu

Manager proiect:
Gabriela Guțu

Autori:

Cristina Sichim
Nușă Dumitriu-Lupan
Livia Demetra Țoca
Maria Niță
Rodica Pinte

București 2011

CUPRINS

Argument	5
I. Cadru de referință pentru sistemul de evaluare a rezultatelor școlare	7
1. Dinamica definițiilor evaluării școlare	7
2. Tendințe în modernizarea evaluării școlare	9
2.1. Evoluții reprezentative în aria evaluării școlare	9
2.2. Trecerea de la evaluarea tradițională la evaluarea modernă.....	9
3. Competențele de evaluare ale cadrelor didactice	13
4. Calitatea evaluării	14
II. Sistemul conceptual metodologic al evaluării școlare	15
1. Componentele evaluării	15
1.1. Obiectul evaluării	15
1.2. Evaluarea centrată pe competențe	16
1.3. Reconceptuarea evaluării din perspectiva competențelor	17
2. Criteriile în evaluarea educațională	18
2.1. Criteriu, criteriu de evaluare	18
2.2. Indicatorul în evaluare	18
3. Operațiile evaluării	19
3.1. Măsurarea – baza obiectivă a aprecierii	19
3.2. Aprecierea – exprimarea unei judecăți de valoare	19
3.3. Decizia - scopul demersului evaluativ	19
3.4. Complementaritatea operațiilor evaluării	20
4. Strategii de evaluare	20
4.1. Strategia evaluativă	20
4.2. Perspectiva criterială	20
4.3. Perspectiva axelor polare	21
5. Tipuri de evaluare	22
6. Metode, tehnici, instrumente de evaluare	24
6.1. Metoda de evaluare	24
6.2. Itemul de evaluare	25
6.3. Tehnicile de evaluare	26

6.4. Instrumentul de evaluare	26
Evaluarea competențelor la disciplina informatică	27
Curriculum și evaluare la disciplina informatică. Formarea și evaluarea competențelor	27
Instrumente de evaluare a competențelor la disciplina de studiu informatică	51
1. Itemi obiectivi	51
1.1. Itemi cu alegere duală	51
1.2. Itemi de tip pereche	54
1.3. Itemi cu alegere multiplă	55
2. Itemi semiobiectivi	56
1.1. Itemii cu răspuns scurt	57
1.2. Itemii de completare	57
1.3. Întrebările structurate	58
1.4. Itemi de tip eseu	60
2. Metode complementare de evaluare	64
2.1. Investigația	64
2.2. Referatul și proiectul	65
2.3. Portofoliul.....	67
2.4. Observarea sistematică a activității și comportamentului elevilor	69
2.5. Hărțile conceptuale	71
2.6. Metoda r. A. I.	73
Evaluarea competențelor specifice. Exemple de probe de evaluare	75
1. Exemple de probe de evaluare predictivă	78
1.1. Evaluare predictivă la început de ciclu liceal	79
1.2. Probă de ameliorare și progres	81
2. Exemple de probe de evaluare continuă	85
2.1. Evaluare orală.....	85
2.2. Evaluare scrisă.....	87
2.3. Evaluare practică	95
Evaluare digitală - concluzii	125
Bibliografie	126

ARGUMENT

Prezentul ghid metodologic are ca destinație diferite categorii de conceptori și utilizatori de probe și instrumente de evaluare a rezultatelor școlare ale elevilor – cadre didactice, manageri școlari și, într-o anumită măsură, specialiști-cercetători. Populația țintă dominantă o reprezintă totuși corpul profesoral, practicienii, în special din învățământul liceal.

Ghidul urmărește două *obiective* solidare:

A) sugerează repere și elemente de reflecție pentru fortificarea culturii evaluative a cadrelor didactice;

B) își propune să asiste practicienii, furnizându-le norme, reguli operaționale și ilustrări concludente, în vederea dezvoltării capacităților lor pe de o parte, de proiectare, validare și administrare a unor variate proceduri de evaluare, iar pe de altă parte, de interpretare și valorificare a rezultatelor evaluării. Finalitatea convergentă a celor două obiective rezidă în creșterea calității educației școlare.

Se speră, de asemenea, ca prin aplicarea sistematică și consecventă a ghidului să rezulte treptat o bancă de itemi pe discipline, arii curriculare și teme crosscurriculare, care să poată fi utilizată selectiv, în funcție de contextele și nevoile specifice de evaluare. Se poate observa că acest ghid nu abordează de tipuri de evaluări ca evaluare instituțională, evaluare de programe, testările standardizate etc.; centrul de greutate îl reprezintă evaluarea învățării, ca produs și ca proces, și a resurselor interne ale școlii (teachers made tests).

Legitimitatea și concepția ghidului are la bază câteva principii:

➤ Reforma învățământului presupune schimbări semnificative corelate în toate componentele sale majore: structuri instituționale, management, curriculum, instruire și, nu în ultimul rând, evaluare.

Componenta evaluării educaționale își are propria identitate, revendică nevoi interne de dezvoltare; funcționalitatea sa depinde însă și de natura și calitatea interacțiunilor cu celelalte componente conexe aparținând sistemului de învățământ: curriculum, instruire, formarea și dezvoltarea profesională a cadrelor didactice. Viziunea sistemică este indispensabilă atât teoreticienilor, cât și practicienilor din aria evaluării școlare.

➤ Creșterea calității sistemului de evaluare educațională este unul dintre obiectivele prioritare ale reformei școlii care pretinde investiții de concepție și practici bune. Dacă examinăm schimbările care s-au produs la noi în ultimii 15 ani, în sfera evaluării

educaționale, constatăm că atât consistența, cât și anvergura acestora nu s-a distribuit egal pe toate treptele învățământului. În mod surprinzător, permeabilitatea la transformările inovatoare s-a redus progresiv odată cu trecerea la treptele superioare de școlarizare. Se pare că veriga învățământului liceal a concentrat mai multe vulnerabilități – indecizii și inconsecvențe politice, practici tradiționale frecvent sau exclusiv utilizate etc. Probabil că în această zonă sunt necesare acțiuni compensatorii și ameliorative mai accentuate. Ghidul vine în întâmpinarea acestei nevoi.

➔ Dezvoltarea profesională a cadrelor didactice în domeniul evaluării educaționale solicită două componente: cultura evaluării și competențele metodologice ale evaluării. Prima integrează concepte nodale teoretice și metodologice, informații de profil aduse la zi, gândire critic-constructivă aplicabilă noilor tendințe și inovații, convingeri raționale privind importanța și limitele evaluării, capacitatea de reflecție a cadrelor didactice asupra propriilor prestații evaluative. Cea de a doua include competențe practice care se distribuie pe un continuum, de la proiectarea evaluării, până la utilizarea rezultatelor evaluării în scopul adoptării unor decizii. *Cultura* oferă o concepție și o atitudine, pe când competențele metodologice sunt *instrumentele* concepției.

Există astăzi suficiente evidențe (unele vor fi semnalate mai jos) care atestă prezența unor schimbări relevante și inovatoare la nivelul fiecărei componente. Acestea ar trebui să se regăsească în sistemul de formare a cadrelor didactice și, mai mult, în practicile curente de evaluare.

➔ Cercetarea științifică dedicată evaluării educaționale, achizițiile din domeniile conexe - învățare, curriculum, instruire - sunt surse importante pentru funcționarea și optimizarea procesului de evaluare școlară. De exemplu, cercetările inspirate de modelul neobehaviorist al învățării vor sugera o anumită strategie a evaluării – definirea riguroasă a criteriilor, preferabil în termeni cantitativi, controlul strâns al învățării prin evaluare și feedback corectiv, „întărirea” rezultatelor prin confirmarea succesului etc. În schimb, modelul constructivist al învățării va orienta demersurile evaluării pe o altă traiectorie: sarcini „autentice de rezolvat”, evaluarea autentică, construcția și nu selecția răspunsurilor, încurajarea opiniilor personale, implicarea elevilor în procesele de evaluare și autoevaluare etc. De altfel, putem constata că abordarea/ evaluarea constructivistă câștigă tot mai mult termen în cadrul evaluării școlare, împrejurare care nu poate fi ignorată de concepatorii și utilizatorii instrumentelor de evaluare.

➔ Proiectarea și exploatarea cu succes a strategiilor, metodelor și tehnicilor de evaluare presupune combinația – în doze diferite, potrivit naturii probei - de principii și reguli cu imaginație creativă. Evaluarea este știință și artă; ea nu se reduce la aplicarea unor structuri algoritmice predeterminate, după cum nu se poate realiza numai pe temeiul spontaneității și experienței. Este întotdeauna un aliaj subtil între știință și artă. Din această perspectivă ghidul de față nu poate fi un rețetar de bucate; el oferă o viziune, perspective, principii și norme operaționale care pot fi valorizate adecvat și inventiv.

I.CADRU DE REFERINȚĂ PENTRU SISTEMUL DE EVALUARE A REZULTATELOR ȘCOLARE

1. DINAMICA DEFINIȚIILOR EVALUĂRII ȘCOLARE

Este de reținut faptul că în evoluția conceptului de evaluare sunt identificate trei categorii de definiții (Hadji, Stufflebeam, 1980, C. Cucoș, 2008): definiții „vechi”, care pun semnul egalității între evaluare și măsurare; definiții care interpretează evaluarea prin raportare la obiectivele educaționale operaționalizate; definiții „moderne” evaluarea fiind concepută ca emitere de judecăți de valoare despre procesul și produsul învățării pe baza criteriilor calitative.

Fiecare din aceste categorii de definiții oferă avantaje și dezavantaje.

Definiții recente, deși diverse, au multe note comune, ca de exemplu:

➔ trecerea accentuată de la evaluarea estimativă bazată pe cantitate, predominant sumativă, la evaluarea apreciativă, bazată pe calitate, cu puternice accente formative;

➔ deplasarea accentului de la înțelegerea evaluării ca examinare și control la „evaluarea școlară ca parte integrantă a procesului de învățare și jalon al acesteia” (Y. Abernot, 1996).

Câteva definiții semnificative pot fi orientative și utile cadrelor didactice. Astfel, evaluarea:

➔ constă în măsurarea și aprecierea cu ajutorul criteriilor, a atingerii obiectivelor sau a gradului de apropiere sau de proximitate a unui produs al elevului în raport cu o normă;

➔ are sensul de atribuire a unei note sau a unui calificativ unei prestații a elevului (Y. Abernot);

➔ examinează gradul de corespondență între un ansamblu de informații privind învățarea de către elev și un ansamblu de criterii adecvate obiectivului fixat, în vederea luării unei decizii (de Ketele, 1982);

➔ este „actul prin care [...] referitor la un subiect sau un obiect, se emite o judecată având ca referință unul sau mai multe criterii”. (Noizet, 1978);

➔ înseamnă „a verifica, a judeca, a estima, a situa, a reprezenta, a determina, a da un verdict etc” (Hadji).

Sinteza interpretărilor privind evaluarea evidențiază o pluralitate de termeni care pot desemna activități integrate în procesul de evaluare. Astfel, a evalua semnifică:

- A verifica ceea ce a fost învățat, înțeles, reținut; a verifica achizițiile în cadrul unei progresii;
- A judeca activitatea elevului sau efortul acestuia în funcție de anumite recomandări; a judeca nivelul de pregătire al unui elev în raport cu anumite norme prestabilite;
- A estima nivelul competenței unui elev;
- A situa elevul în raport cu posibilitățile sale sau în raport cu ceilalți; a situa produsul unui elev în raport cu nivelul general;
- A reprezenta printr-un număr (notă) sau calificativ gradul reușitei unei producții școlare a elevului în funcție de diverse criterii;
- A pronunța un verdict asupra cunoștințelor sau abilităților pe care le are un elev;
- A fixa/stabili valoarea unei prestații a elevului etc.

Evaluarea se definește din mai multe perspective. Astfel:

➔ *Din punct de vedere structural*, presupune: obiectul evaluării, criteriile de evaluare și analiza comparativă a relațiilor dintre caracteristicile obiectului de evaluat și criteriile în funcție de care se face evaluarea;

➔ *Din punct de vedere funcțional* evaluarea presupune:

- a. un scop specific - determinarea valorii ce se atribuie calității rezultatelor școlare, proceselor, programelor etc.;
- b. funcții - maniera în care sunt valorificate rezultatele evaluării: pentru certificare, pentru selecție, pentru reglare/ameliorare etc.

➔ *Din punct de vedere al operațiilor* presupune o desfășurare procesuală, ce implică anumite operații specifice: măsurare, apreciere, decizie. Cele trei operații se susțin una pe cealaltă și se justifică numai împreună. O caracteristică a ultimelor lucrări de referință în domeniu este aceea că abordează evaluarea prioritar în termeni de procese. Dorința de a asigura obiectivitate cât mai ridicată prin operația de măsurare este diminuată; dezbaterile pe această temă reflectă tendința de a depăși înțelegerea tradițională a evaluării ca instrument de măsură și control, abordarea acesteia din perspectiva unui demers centrat pe învățare, pe procesele cognitive ale elevului, pe reglarea și pe autoreglarea cunoașterii.

Analizele de mai sus conduc la ideea potrivit căreia la ora actuală, dar mai ales în perspectivă, schimbările dominante în domeniul școlar instituie evaluarea ca mijloc de formare a elevului și de observare a evoluției competențelor sale. Efectele oricărei acțiuni de evaluare se manifestă în moduri diferite, cu funcții și consecințe dintre cele mai diverse, în raport de intențiile dominante care stau la baza demersului respectiv: control sau remediere, certificare sau selecție, diagnosticare sau prognosticare etc.

Funcțiile evaluării

Funcțiile evaluării vizează semnificații, conotații, mecanisme și consecințe pe baza a ceea ce considerăm a fi, pe de o parte, planuri de analiză (individual, social, de grup) și, pe de alta, criterii psihopedagogice, sociologice, docimologice.

Evaluarea îndeplinește următoarele funcții:

- constatativă, diagnostică - de cunoaștere a stării, fenomenului, obiectului evaluat;
- diagnostică - de explicare a situației existente;
- predictivă, de prognosticare și orientare a activității didactice, atât de predare cât și de învățare, concretizată în deciziile de ameliorare sau de reproiectare

curriculară;

- selectivă - asigură ierarhizarea și clasificarea elevilor într-un mediu competitiv;
- feedback (de reglaj și autoreglaj); analiza rezultatelor obținute, cu scopul de reglare și autoreglare conduitei ambilor actori;
- social-economică: evidențiază eficiența învățământului, în funcție de calitatea și valoarea „produsului” școlii;
- educativă, menită să conștientizeze și să motiveze, să stimuleze interesul pentru studiu, pentru perfecționare și obținerea unor performanțe cât mai înalte;
- socială, prin care se informează comunitatea și familia asupra rezultatelor obținute de elevi.

Aceste funcții sunt complementare.

2. TENDINȚE ÎN MODERNIZAREA EVALUĂRII ȘCOLARE

2.1. EVOLUȚII REPREZENTATIVE ÎN ARIA EVALUĂRII ȘCOLARE

➔ Caracteristica esențială a activității evaluative o reprezintă astăzi abordarea acesteia atât în termeni de procese, formulare și de proceduri privind măsurarea rezultatelor învățării. Activitatea presupune desfășurare, procesualitate, reglare, autoreglare etc.

➔ Căutarea echilibrului între învățarea ca proces și învățarea ca produs; între aspectele sumative, clasificatoare, certificatoare și cele care permit identificarea cauzelor/dificultăților întâmpinate de elevi în învățare, precum și între mecanismele reglatorii și cele autoreglatorii.

➔ *Evaluarea formativă*, concept operant în teoria și practica evaluării reprezintă:

- a. nucleul priorităților în deciziile privind combinatorica între procesele de învățare și competențele văzute ca rezultat al învățării;
- b. co-responsabilizarea celui care învață, prin dezvoltarea capacității de auto-reflecție asupra propriei învățări, și funcționalitatea mecanismelor meta-cognitive/ cunoaștere despre autocunoaștere;
- c. centrarea învățământului pe competențe generale și specifice, pe parcursul și la finalul unui ciclu de instruire, al unui an de studiu etc.

➔ În prezent, teoria pedagogică dar și practica în domeniu au drept țință:

- a. diversificarea metodologiei, dispozitivelor, tehnicilor și instrumentelor de evaluare pentru a realiza ceea ce G. de Landsheere aprecia: evaluarea școlară să devină mai exactă din punct de vedere științific și mai echitabilă din punct de vedere moral;
- b. regândirea „culturii controlului și examinării” și promovarea a ceea ce numim „*cultură a evaluării*”, centrată pe procesele socio-cognitive, metacognitive în învățare; asigurarea feedbackului orientat spre finalitățile proiectelor

tului evaluativ.

➔ La nivelul clasei de elevi se insistă pe anumite inovații, rezultate din complementaritatea *metodelor tradiționale* (evaluări orale, scrise, probe practice etc.) cu altele noi, *moderne* (portofoliul, proiectul, investigația, autoevaluarea etc.), în fapt *alternative*.

➔ Elaborarea probelor prin integrarea de „itemi obiectivi, semiobiectivi și subiectivi”, prin realizarea și aplicarea matricelor de evaluare, de statistici privind evoluția rezultatelor elevilor, de diminuare a erorilor mai frecvente în procesul evaluativ.

O remarcă specială merită făcută cu privire la *trecerea de la evaluarea tradițională la evaluarea modernă*, în care conduitele cadrelor didactice și evaluatorilor externi se vor axa pe:

- măsurarea și aprecierea obiectivă și evolutivă a rezultatelor;
- adoptarea unor decizii și măsuri ameliorative;
- emiterea unor judecați de valoare;
- acoperirea domeniului cognitiv dar și a celui social, afectiv, spiritual și psihomotor;
- feedbackul oferit elevului;
- informații semnificative oferite cadrelor didactice privind eficiența activității lor;
- cunoașterea criteriilor/ normelor cu care se evaluează, creșterea gradului de adecvare la situații didactice concrete;
- evitarea sancționării cu orice preț a erorilor; respectarea principiilor contractului pedagogic.

2.2. TRECEREA DE LA EVALUAREA TRADIȚIONALĂ LA EVALUAREA MODERNĂ

Simptomatic pentru anvergura și diversitatea schimbărilor reale sau preconizate în cadrul sistemelor actuale, europene sau transeuropene de evaluare a progresului școlar este faptul că aceste schimbări acoperă întreaga problematică majoră a evaluării, sintetizată de întrebările: *Ce se evaluează?, De ce?, Cum?, Cu ce agenți?, Când?, Cu ce costuri?*. Dincolo de răspunsurile „clasice” pe care le găsim în manualele consacrate evaluării, se conturează noi răspunsuri sau cel puțin sunt formulate noi accente. Acestea, pe ansamblu, configurează o nouă paradigmă a evaluării școlare, care are implicații asupra politicilor și practicilor educației.

EVALUAREA TRADIȚIONALĂ	EVALUAREA MODERNĂ
<ul style="list-style-type: none"> • Cultura controlului/ examinării 	<ul style="list-style-type: none"> • Cultura evaluării - promovarea unei <i>noi mentalități</i> privind evaluarea școlară în context cotidian (și nu numai): dirijarea învățării, asigurarea feed-back-ului, comunicarea, creșterea calității evaluatorilor (Perretti, Hadji, de Ketele, Abernot etc.) - cadrele didactice evaluatori trebuie să respecte câteva reguli simple:

	<ul style="list-style-type: none"> ➔ interpretarea mesajelor evaluării; ➔ identifică intențiile dominante ale activității evaluative; ➔ buna gestionare a potențialului formativ al evaluării; ➔ evitarea capcanelor/ erorilor specifice; ➔ economia mijloacelor de evaluare; ➔ evitarea redundanțelor (Hadji).
<ul style="list-style-type: none"> • Evaluarea intrărilor - evaluarea cunoștințelor 	<ul style="list-style-type: none"> • Evaluarea ieșirilor din sistemul de formare - diversificarea spectrului de achiziții școlare supuse evaluării: cunoștințe, deprinderi, capacități, produse creative, valori și atitudini; - tranziția de la cunoștințe la capacități și de la capacități primare la capacități cognitive de ordin superior; - centrarea evaluării pe competențele educaționale/ profesionale. Reprezentările despre structura și tipologia competențelor trasează direcții de evaluare și solicită metode și tehnici diferite (conform „Cadrul european al calificărilor”).
<ul style="list-style-type: none"> • Scopul evaluării - măsurarea cantitativă a cunoștințelor - controlul rezultatelor cuantificabile ale învățării - sancționarea 	<ul style="list-style-type: none"> • Multiplicarea scopurilor/ funcțiilor evaluării - creșterea rolului evaluării de impact; - determinarea valorii unui program educațional prin rezultatele produse; - stabilirea răspunderilor pentru calitatea rezultatelor (funcția „accountability”); - dezvoltarea evaluării pentru învățare; - utilizarea pârgurilor evaluării formative pentru motivarea învățării - evaluarea este pusă în serviciul optimizării învățării; - comunică elevului informații utile despre calitatea progreselor, orientându-i eforturile, pornind de la statutul său de ființă care nu a încheiat procesul de dezvoltare (Y. Abernot).
<ul style="list-style-type: none"> • Metodologia evaluării - metode clasice - „cultura testării” (testing culture) care apelează la măsurători și itemi obiectivi și semiobiectivi 	<ul style="list-style-type: none"> • Perfecționarea și inovarea metodologiei evaluării - consolidarea și dezvoltarea regulilor și condițiilor de utilizare a metodelor „clasice”; - maturizare și rafinare tehnică; - „cultura aprecierii” (assessment culture), care exploatează potențialul metodelor alternative (proiectul, investigația, portofoliul), itemii deschiși, introduce alte criterii de evaluare, calitative, considerând că învățarea este o activitate complexă multidimensională, iar calitatea ei nu se reduce numai la un ansamblu de rezultate exclusiv cuantificabile. Se apreciază, de asemenea, că metodele calitative sunt apte să detecteze progresul în învățare și să evalueze capacitățile cognitive de ordin superior. - valorificarea resurselor oferite de TIC în evaluare: computerizarea evaluării; dezvoltarea de software-uri specifice;

	<ul style="list-style-type: none"> - apreciază drumul parcurs de elev, cât de semnificative și relevante sunt progresele într-un context dat; - este pusă în slujba procesului educativ și integrată acestuia.
<ul style="list-style-type: none"> • Evaluatorii - profesorul este unicul evaluator 	<ul style="list-style-type: none"> • Diversificarea agenților evaluatori - profesorul își conservă rolul de evaluator esențial al performanțelor școlare; - elevul participă la procesul de evaluare în două forme: <ul style="list-style-type: none"> ➔ evaluarea colegială (peer evaluation) ➔ autoevaluarea <p>Competențele de evaluare/ autoevaluare ale elevilor extind registrul competențelor promovat de școli și sunt expresia concludentă a învățării centrate pe elevi și în aria evaluării.</p> <ul style="list-style-type: none"> - echilibrarea evaluării interne cu evaluarea externă.
<ul style="list-style-type: none"> • Evaluarea în orizontul de timp - evaluarea tradițională acordă de regulă preponderență identificării și evaluării rezultatelor finale ale învățării – evaluare sumativă, utilizându-se mai ales probe specifice sumative. 	<ul style="list-style-type: none"> • Evaluarea în orizontul de timp - concepția actuală plasează evaluarea înaintea, în timpul și după învățare - evaluare inițială, formativă și sumativă. Fiecare din cele trei tipuri revendică construirea și utilizarea unor probe specifice: diagnostice, de progres și sumative.
<ul style="list-style-type: none"> • Costurile evaluării - costurile materiale și financiare – minimalizate sau tratate ad hoc - resursă umană – redusă la profesorul clasei 	<ul style="list-style-type: none"> • Costurile evaluării - achiziționarea de teste educaționale, proiectarea, validarea, administrarea și utilizarea rezultatelor evaluării antrenează costuri de resurse umane, materiale și financiare; - problematica acestor costuri ale evaluării nu poate lipsi din strategia edificării unui sistem eficace și eficient de evaluare școlară.

Putem concluziona că, din perspectivă modernă, evaluarea nu este sinonimă nici aprecierii clasice, nici acordării notei, nici controlului continuu al învățării școlare și nici clasamentului/ clasificării. Evaluarea se bazează pe judecata specializată a profesorului, pe competența profesională a experților implicați în evaluare.

Totodată, trebuie subliniat faptul că polaritățile menționate (de ex. Cunoștințe versus capacități; evaluarea de control versus evaluarea în serviciul învățării) nu se află în raporturi disjunctive, de excludere reciprocă. Ele reprezintă mai mult capetele unui continuum pe traseul căruia pot funcționa diferite variante, selecționate în raport cu obiectivele și situațiile particulare de evaluare. Este eronată ideea că orientările „clasice” ar trebui excomunicate totalmente, iar orientările „moderne” sunt universal valabile, în orice circumstanță. Tranzițiile, mișcările, deplasările de accent de la o poziție la alta, semnalate mai sus, ar trebui interpretate ca evoluții tendențiale, schimburi de ponderi și nu ca abandon categoric al „punctelor” de plecare.

3. COMPETENȚELE DE EVALUARE ALE CADRELOR DIDACTICE

Evoluțiile și dezvoltările actuale din cercetare, teoria și practicile bune ale evaluării impun reexaminarea tipurilor și conținuturilor intrinseci ale competențelor de evaluare ale educatorilor. Criteriile de definire pot fi variate: strategiile sau tipurile de evaluare, metodele sau tipurile de itemi, „fizionomia” testelor educaționale.

În SUA, Comisia de Standarde pentru Competențele Evaluative ale Cadrelor Didactice a identificat un număr de șapte competențe/ standarde (Apud Hanna Dettner – 2004) ale cadrelor didactice:

- ➔ alegerea adecvată a metodelor de evaluare;
- ➔ elaborarea metodelor, probelor de evaluare;
- ➔ administrarea și interpretarea rezultatelor evaluării obținute prin instrumentele dezvoltării de profil sau utilizând teste elaborate extern;
- ➔ utilizarea rezultatelor evaluării valorificându-le în adaptarea de decizii privind elevii, dezvoltarea curriculumului, planificarea instruirii și dezvoltarea instituțională a școlii;
- ➔ dezvoltarea și aplicarea procedeele de notare a elevilor;
- ➔ comunicarea rezultatelor evaluării având în vedere diferite categorii de audiență: elevi, părinți, administrație, comunitate;
- ➔ recunoașterea și evitarea implicațiilor non etice, ilegale, efectelor distorsionate ale unor proceduri de evaluare.

Observăm că în afara ultimei competențe, care indică mai mult obligativitatea respectării unui cod deontologic, etic în evaluare, toate celelalte gravitează în jurul construcției, selecției, utilizării și evaluării rezultatelor instrumentelor de evaluare – obiectul de interes major al acestui ghid. În consecință, ghidul ar putea avea o contribuție specifică la dezvoltarea competențelor cheie ale corpului didactic în aria evaluării educaționale.

4. CALITATEA EVALUĂRII

Evaluarea educațională, ca și alte activități subsumate educației instituționalizate nu poate evita ralierea la standardele calității. Mai mult, se justifică dezvoltarea unui mecanism propriu de asigurare a calității care să jaloneze politicile și managementul evaluării școlare. Acesta propune:

- ➔ Definirea standardelor calitative ale evaluării
- ➔ Evaluarea calității procedurilor de evaluare
- ➔ Abilitarea cadrelor didactice cu sistemele conceptuale și metodologice specifice evaluării academice
- ➔ Asumarea responsabilității pentru calitatea evaluării
- ➔ Modernizarea sistemului de înregistrare și comunicare a rezultatelor obținute de studenți

În mod cert, este nevoie de protejarea, menținerea și creșterea calității evaluării. Un număr de indicatori calitativi pot aduce servicii în această direcție, și pot inspira cu suc-

ces concepția și practicile evaluative ale cadrelor didactice:

➔ Evaluarea este concepută și se aplică diferențiat potrivit funcțiilor pe care și le asumă: diagnostică, prognostică, de informare, de selecție, de certificare, de orientare-consiliere etc.

➔ Evaluarea este utilizată ca factor reglator al interacțiunii predare-învățare, și nu doar ca o componentă finală a procesului de formare.

➔ Tipurile fundamentale de evaluare sunt corelate funcțional și valorificate echilibrat.

➔ Există o corespondență clară între obiectivele învățării, ceea ce se predă și se învață, și cunoștințele, capacitățile și atitudinile evaluate (coerența curriculum-evaluare)

➔ Sarcinile de evaluare vizează competențe profesionale specifice, dar și competențe generice transversale.

➔ Focalizarea probelor pe sarcini „autentice” – situații, probleme reale, cu impact semnificativ.

➔ Proiectarea probelor de evaluare se realizează profesional, asigurându-se condițiile necesare de validare și fidelitate.

➔ O varietate de metode este folosită astfel încât limitele metodelor particulare să fie minimalizate, iar efectul lor cumulativ potențat.

➔ Notele sunt acordate transparent și obiectiv, pe baza rezultatelor învățării și a criteriilor privind nivelul de performanță.

➔ Elevii primesc un feedback evaluativ sistematic care le permite să-și organizeze procesul de învățare.

➔ Implicarea elevilor în procese de evaluare și autoevaluare.

➔ Probele de evaluare sunt controlate pentru a se asigura că nu există influențe subiective care pot defavoriza grupuri particulare.

➔ Transparența criteriilor și procedurilor de evaluare; accesarea lor fără dificultăți.

➔ Crearea unui climat cu impact emoțional pozitiv, motivant și securizat; reducerea stărilor de stres și anxietate prin ambianță și comportamentul cadrului didactic – deschis, cooperant, prietenos.

➔ Prevenirea și combaterea prin reguli clare și aplicate a fraudelor academice (copiat, plagiat, „importul” de lucrări etc).

II. SISTEMUL CONCEPTUAL METODOLOGIC AL EVALUĂRII ȘCOLARE

1. COMPONENTELE EVALUĂRII

Elementele apreciate drept componente sau variabile ale evaluării școlare sunt:

1. *obiectul evaluării (ce vom evalua: tipuri de procese/rezultate/produse de evaluat.)*
2. *criteriile evaluării (în raport de ce se evaluează)*
3. *operațiile evaluării (ce pași trebuie făcuți)*
4. *strategiile (proiectarea și coordonarea acțiunilor)*
5. *tipurile/formele de evaluare*
6. *metode, tehnici, instrumente (cum vom evalua)*
7. *timpul evaluării (momentele oportune pentru evaluare: înainte, în timpul, sau după acțiunea educativă)*
8. *agenții evaluării (factori de conducere, cadre didactice, elevi, experți externi)*

1.1. OBIECTUL EVALUĂRII

„Obiectul evaluării” reprezintă realitatea educațională concretizată în procesul și produsul învățării, supusă atenției evaluatorului, în vederea măsurării și aprecierii. Prin evaluare, se emit judecăți privind valoarea procesului și produsului învățării realizată de elev. Valoarea „obiectului” evaluat rezultă din conformitatea mai mică sau mai mare cu o normă ideală; ceea ce a învățat elevul și intră în atenția evaluatorului este comparat cu etalonul, cu dezirabilul.

În practica școlară identificăm mai multe modalități de determinare/specificare a „obiectului evaluării”:

1. Specificarea prin conținut - modalitate proprie învățământului tradițional centrat pe „materia” de învățat. Conținuturile sunt importante în sine.
2. Specificarea prin obiective operaționale/comportamentale - modalitate derivată din PPO (Pedagogia prin obiective). Acest tip de specificare vizează comportamente observabile și pe cât posibil măsurabile ale elevului în procesul învățării.
3. Specificarea pe bază de competențe. Competențele generale și competențele specifice, identificabile pentru fiecare disciplină din învățământul preuniversitar vizează

competențe, capacități, sub-capacități și performanțe ce urmează a fi dezvoltate și respectiv evaluate pe parcursul și la finalul diverselor perioade de timp pentru care acestea au fost definite.

1.2. EVALUAREA CENTRATĂ PE COMPETENȚE

Competența - un concept polisemantic

Din cauză că are o mare doză de polisemantism competența este înțeleasă diferit și este tratată diferit în diverse sisteme de învățământ.

A) *Competența, în general*, reprezintă capacitatea unui individ de a mobiliza un ansamblu integrat de resurse (cognitive, afective, relaționale, comportamentale etc.) pentru a rezolva cu eficiență diverse categorii de probleme sau familii de situații - problemă.

B) *În domeniul educațional*, competența este capacitatea de selecție și combinare a cunoștințelor și capacităților susținute valoric și atitudinal de a rezolva cu succes o sarcină de învățare în raport cu standardele determinate (Dan Potolea).

Competența școlară poate fi considerată ca o disponibilitate acțională a elevului, bazată pe resurse bine precizate, dar și pe experiența prealabilă, suficientă și semnificativ organizată. Se materializează în performanțe ale elevului, predictibile în mare măsură pe baza prestațiilor anterioare.

C) *Structura unei competențe:*

➔ resursele, constituite din: cunoștințe („a ști”), deprinderi/abilități („a face”) și atitudini, valori („a fi, a deveni”);

➔ situațiile concrete în care elevul învață și pune în practică acel potențial. Fără crearea situațiilor concrete create pentru a pune în aplicare ceea ce a învățat, acel potențial rămâne doar în planul lui „a ști”, nu trece în planul lui „a face”. Rămâne, în plan teoretic, în planul lui „a ști”. Trebuie completat cu „a face” și „a deveni”.

D) *Competența - un potențial*

Competența trebuie probată/demonstrată în situații concrete. Pentru a fi evaluată, competența trebuie să beneficieze de situații concrete în care cel ce studiază va demonstra că este capabil să pună în practică, să valorifice ceea ce a învățat. Situațiile în care acesta dovedește o competență sunt integrate în familii de situații. Fiecărei competențe i se asociază o „**familie de situații**”. Acestea sunt *situații echivalente*.

E) *Competența se exprimă în performanțe.*

Performanțele unui elev exprimă nivelul la care o competență/competențele a/au fost dobândite de către acesta. Deci, performanța este expresia competenței, forma și nivelul ei de manifestare în plan personal. Teoria și practica pedagogică intenționează să se deplaseze accentual de la paradigma tradițională a evaluării centrată pe cantitate, pe obiectivitate maximă, la evaluarea centrată pe calitate. În contextul evaluării centrate pe competențe, standardele la care se raportează rezultatele învățării elevului trebuie să fie de natură calitativă. În pedagogia modernă, aceste standarde sunt reprezentate de „descriptorii de performanță”. Un standard este o unitate de măsură/apreciere etalon, este un „stass”. Pentru a asigura o evaluare corectă și unitară, procesul și produsul învățării fiecărui elev trebuie să fie raportate la standardele de performanță stabilite la nivel național. Performanțele personale/individuale trebuie apreciate în funcție de gradul de apropiere sau depărtare de aceste unități cu valoare de „etalon”.

F) *Competența reprezintă un mega-rezultat educațional.*

Poate fi corelată cu un obiectiv educațional cu grad mare de generalitate: obiectiv de arie curriculară, obiectiv cadru general etc., putând fi integrat în soclurile de competență.

G) *Competența - finalitate a procesului educațional și „obiect” al evaluării*

În condițiile învățământului modern, competența se transformă în finalitate a procesului educațional și „obiect” al evaluării școlare. Competențele școlare disciplinare/transversale dobândite de elev în cadrul unei instruirii și evaluări „autentice” au luat locul obiectivelor operaționale/comportamente (observabile și măsurabile). În domeniul evaluativ ne aflăm în faza reconceperii evaluării, a trecerii de la evaluarea obiectivelor la evaluarea competențelor școlare.

1.3. RECONCEPEREA EVALUĂRII DIN PERSPECTIVA COMPETENȚELOR

Presupune:

▪ Extinderea evaluării de la verificare și apreciere a rezultatelor la evaluarea procesului, a strategiei de învățare a elevului, purtătoare de succes; evaluarea elevilor dar și a obiectivelor, a conținutului, metodelor, a situației de învățare, a evaluării însăși.

▪ Luarea în considerare, pe lângă achizițiile cognitive, și a altor indicatori, precum: personalitatea, conduita, atitudinile; aplicarea în practică a celor învățate; diversificarea tehnicilor de evaluare și adecvarea acestora la situațiile concrete (teste docimologice, lucrări de sinteză, tehnici de evaluare a achizițiilor practice, probe de aptitudini, conduita, valorizare etc.);

▪ Deschiderea evaluării spre viață: competențe relaționale, comunicare profesor - elev, disponibilități de integrare socială;

▪ Scurtarea drumului evaluare – decizie - acțiune ameliorativă, inclusiv prin integrarea eforturilor și disponibilităților participative ale elevilor; centrarea pe aspectele pozitive și nesancționarea în permanență a celor negative;

▪ Transformarea elevului într-un partener al profesorului în evaluare, prin: autoevaluare, inter-evaluare, evaluare controlată.

1.4. CENTRAREA PE COMPETENȚE – UN MODEL INTEGRATOR AL EVALUĂRII

Această nouă concepție prefigurează construcția unui *nou model integrator al evaluării*, care valorifică deopotrivă și încearcă să coreleze cunoștințe, deprinderi, capacități de aplicare a cunoștințelor, valori și atitudini ale elevului.

2. CRITERIILE ÎN EVALUAREA EDUCAȚIONALĂ

2.1. CRITERIU, CRITERIU DE EVALUARE

„Criteriu” vine de la latinescul „criterium” și desemnează principiul care stă la baza unei judecăți, a unei estimări, a unei clasificări, permite distingerea adevărului de fals etc. Criteriile de evaluare sunt puncte de vedere, caracteristici, dimensiuni în funcție de care se evaluează rezultatele școlare ale elevilor. Utilizarea criteriilor în evaluare devine un element de obligativitate. Existența criteriilor este esențială atât pentru elev cât și pentru cadrul didactic, în orice tip de evaluare, fie ea inițială, formativă sau sumativă.

Tipuri de criterii în evaluare

Activitatea de învățare a elevilor a fost și este evaluată, în mod tradițional, prin raportare la cel puțin *patru tipuri de criterii* principale, dispuse pe două axe polare:

➤ Axa 1: Norma/media clasei (norma statistică a clasei respective) sau standardele procentuale locale, naționale sau internaționale versus „norma” individuală (raportarea la sine însuși).

➤ Axa 2: Raportarea la obiective (evaluarea criterială) versus raportarea la conținutul programei.

2.2. INDICATORUL ÎN EVALUARE

Indicatorul este un element care indică prezența altui element. Acesta are valoare de semnal. Indicatorul nu poate, în sine, prin statutul său, să furnizeze un sens rezultatului pe care îl subliniază; el trebuie să se refere la un criteriu.

Relația criteriu - indicator este foarte strânsă.

Criteriul desemnează o caracteristică, iar indicatorul „semnalează” niveluri de dezvoltare, de prezență a acestei caracteristici într-o anumită situație evaluativă. Într-un context școlar dat, dacă luăm drept criteriu „nivelul performanței în învățare a elevilor”, acestea (performanțele) se distribuie în minimale, medii, maxime. Indicatorii sunt elementele din descriptorii de performanță asociați calificativelor care semnalează prezența diverselor aspecte care trebuie să caracterizeze rezultatul elevului pentru a i se acorda o notă sau un calificativ. În învățământul gimnazial și liceal criteriul de repartizare a performanțelor elevilor este reprezentat de scala numerică de la 10 la 1. Indicatorii enumeră, precizează cum trebuie să arate răspunsul elevului pentru a i se acorda nota 10, sau 9, sau... sau 5 sau un anumit punctaj stabilit prin baremul de evaluare și notare.

3. OPERAȚIILE EVALUĂRII

Operațiile evaluării vizează pașii ce trebuie făcuți în procesul evaluativ până la momentul sau etapa emiterii unei judecăți de valoare asupra prestației elevului. Aceste operații sunt următoarele: măsurarea, aprecierea, decizia.

3.1. MĂSURAREA – BAZA OBIECTIVĂ A APRECIERII

Măsurarea este operația prin care se asigură baza obiectivă a aprecierii. Este prima operație a evaluării. Această operație constituie o primă etapă în evaluarea considerată ca un demers sau un proces. Măsurarea asigură rigurozitate evaluării. Prin ea se strâng informații de către evaluator „despre proprietățile sau caracteristicile rezultatelor înregistrate, despre însușirile procesului, acțiunii sau fenomenului educativ dat”. Informațiile se colectează prin intermediul tehnicilor și instrumentelor, care „produc” dovezi semnificative despre aspectele sau rezultatele luate în considerare. Cu cât instrumentele de măsurare: probe orale, scrise, practice, extemporale, lucrări de sinteză, teste etc. sunt mai bine puse la punct, cu atât informațiile sunt mai concludente.

3.2. APRECIEREA – EXPRIMAREA UNEI JUDECĂȚI DE VALOARE

Aprecierea corespunde emiterii unei judecăți de valoare. Prin această operație, pe baza informațiilor culese prin măsurare dar și prin alte surse mai mult sau mai puțin formale (observare, analize etc.) se stabilește *valoarea* rezultatelor școlare precum și a procesului de învățare. Aprecierea este, deci, ulterioară măsurării. În cazul aprecierii, alocarea de valori numerice, literale sau calificative se realizează pe baza unor criterii precis identificabile, relativ independente de instrumentul prin care s-a făcut măsurarea.

3.3. DECIZIA - SCOPUL DEMERSULUI EVALUATIV

Cea de-a treia operație a evaluării este *decizia*. Luarea deciziilor reprezintă finalul înlănțuirii de operații ce definesc actul evaluării în ansamblul lui și scopul acestui demers. *În decizie își găsesc justificare și măsurarea și aprecierea*. De abia în această etapă își găsesc răspuns întrebări de tipul: „Pentru ce evaluăm? Pentru ce aplicăm proba sau testul? Pentru ce examinăm?” etc.

3.4. COMPLEMENTARITATEA OPERAȚIILOR EVALUĂRII

Cele trei operații se află într-o strânsă interdependență. Evaluare înseamnă: *măsurare + apreciere + decizie*. Una fără alta, aceste trei operații nu se justifică. Modernizarea sistemului de evaluare implică modernizarea acestor trei operații.

4. STRATEGII DE EVALUARE

4.1. STRATEGIA EVALUATIVĂ

În domeniul educațional, strategia evaluativă este un demers care prefigurează *perspectiva* din care va fi concepută evaluarea. Rol esențial îl au: proiectarea dispozitivului de evaluare, construcția acestuia, aplicarea și emiterea judecăților de valoare privind procesul învățării și rezultatele obținute de către elevi. În diferențierea strategiilor evaluative identificăm două perspective de analiză: perspectiva criterială și perspectiva axelor polare.

4.2. PERSPECTIVA CRITERIALĂ

În principal, strategiile educaționale evaluative se proiectează în temeiul următoarelor criterii:

Actorii evaluării (elevi sau profesori), rezultând o evaluare *centrată pe elev* și pe personalitatea sa și o evaluare *centrată pe profesor*, pe corectitudinea sa.

Instrumentele evaluării pe baza cărora distingem între:

- strategii obiective* (evaluare obiectivă) bazate pe teste, probe standardizate și alte instrumente care pot măsura cât mai fidel prestația/permanența elevului și
- strategii calitative* centrate mai ales pe calitatea rezultatelor, fundamentate pe criterii calitative.

Obiectul evaluării conform căruia identificăm:

- strategii sumative* (evaluare sumativă), axată pe produsul final/rezultatele învățării elevilor
- strategii formative* (evaluare formativă), axate pe procesul de învățare ce conduce spre produs.

Forma de organizare (numărul subiecților) potrivit căreia avem:

- strategii de evaluare frontală (eșantionul integral)
- strategii de evaluare de grup
- strategii de evaluare individuală

Referențialul de bază în funcție de care distingem:

- criteriul „conținut” sau norma programei
- norma statistică a grupului școlar (*media clasei*)
- standarde locale, naționale sau internaționale

d. *norma individuală* (raportarea la sine însuși)

e. *evaluarea criterială* (raportarea la obiective).

Parametrul „timp” în evaluare. După momentul plasării evaluării: (Parisat, J. C., 1987):

- evaluarea *inițială*
- evaluarea *curentă* sau formativă sau continuă
- evaluarea *finală* sau recapitulativă sau de bilanț.

Natura deciziilor consecutive. După *natura deciziilor* luate (Meyer, G. 1995):

- evaluare de selecție și ierarhizare
- evaluare de reechilibrare, recuperare și dezvoltare.

Criterii combinate. După un *criteriu compozit* (*combinat*) alcătuit pe baza următoarelor parametri:

- Gradul de cuprindere a elevilor în evaluare;
- Gradul de cuprindere a conținuturilor de evaluat;

Rezultă următoarea clasificare:

- evaluare *exclusiv parțială*; este incidentală, prin sondaj (se evaluează doar *unii* elevi, din *unele* conținuturi și doar *uneori*);
- evaluare *parțială – aditivă*; este evaluarea sumativă/cumulativă; se evaluează, de regulă, *toți elevii*, din *toată materia* parcursă într-un interval dat, dar *numai uneori*.
- evaluarea *cvasitotală* este evaluarea *formativă*, axată pe evaluarea *tuturor* elevilor, din *toate* conținuturile predate/învățate și *tot* timpul. Se apropie de o evaluare ideală.

4.3. PERSPECTIVA AXELOR POLARE

Este posibilă și aplicarea criteriului continuumului *polarității axelor tipologice/conceptuale*. Rezultă, de aici, următoarea configurație a tipurilor de evaluare:

- *formativă – recapitulativă*;
- *criterială – normativă*;
- *produs – proces*;
- *descriere/apreciere – măsurare*;
- *proactivă – retroactivă*;
- „*globală*”, *holistică* – „*analitică*”;
- *internă – externă*;
- *personale – oficiale*;
- *categorială/frontală – personalizată*;
- *integrativă – contextualizată*;
- *reflexivă – participativă*;
- *imperativă – negociabilă*;
- *motivantă – sancționantă*;
- *formală – informală*.

Strategii evaluative normative/comparative

Punctul de plecare apreciem a fi faptul că realizarea performanțelor în învățare ale elevilor este profund diferențiată și selectivă. În consecință, nevoia de a oferi elevilor un avantaj larg al standardelor, de la nivelul inferior și accesibil tuturor până la nivelul celui superior, accesibil unei mici categorii de elevi. Se va realiza astfel, o selecție a elevilor în

funcție de accesul lor la anumite standarde de conținut. Astfel elevii vor fi clasificați utilizându-se curba distribuției acestora. Strategiile care se construiesc în baza acestei concepții sunt strategii normative, comparative; elevii sunt comparați, clasați și ierarhizați. Aceasta tinde să fie înlocuită cu evaluarea criterială sau prin obiective.

Strategii evaluative criteriale

Strategiile criteriale de evaluare au la bază evaluarea prin obiective educaționale. Esența acestor strategii criteriale constă în stabilirea cu mai multă rigoare și finețe numită și evaluare bazată pe „standardul minim acceptat” sau „performanța minimă acceptată”, care exprimă pragul de reușită a unui elev într-o anumită situație educațională.

După modul diferit în care obiectivele pot fi derivate, ierarhizate, definite, formulate și operaționalizate, se face distincție (D. Ungureanu) între următoarele tipuri de strategii evaluative criteriale: cu *obiective prestabilite*; cu *obiective prestabilite dar contextualizate*; derulate în raport cu *obiective conjuncturale sau* configurate ad-hoc; *obiective operaționalizate* prin proceduri riguroase; cu *obiective slab structurate*, orientative, direcționale (fără a se preciza în ce ritm, în ce timp, în ce succesiune).

5. TIPURI DE EVALUARE

Evaluarea rezultatelor școlare ale elevilor se realizează într-o diversitate de *forme/ tipuri*, condiționate de variabile și criterii multiple.

A. După natura funcției școlare și sociale îndeplinite, se pot face următoarele diferențieri (I. T. Radu, 1999, pag. 97):

➔ *evaluări curente*, pe secvențe mici de activitate. Din perspectiva teoriei moderne a evaluării, aceste forme sunt integrate organic în procesul didactic, având preponderent funcții reglatorii, ameliorative.

➔ *evaluări la intrarea într-un ciclu de învățământ*, prin selecție. Admiterea se poate realiza pe baza unui examen concurs sau pe baza performanțelor obținute în ciclul de învățământ absolvit. În practica școlară românească au funcționat ambele forme de evaluare, luând forma „examenului de capacitate”, respectiv a „tezelor cu subiect unic”. Cele două modalități au avantaje și dezavantaje (C. Cucuș, 2008).

➔ *verificări finale/examene*, la sfârșit de an școlar, ciclu de învățământ. „Se prezintă ca formă de control asupra rezultatelor școlare, ca acțiuni relativ separate de programul de instruire propriu zis... Examenul constituie una din formele principale ale evaluărilor de bilanț, utilizată cu deosebire la încheierea unei perioade mai îndelungate de activitate: an școlar, ciclu de învățământ etc” (I.T. Radu, op. Cât). Evaluările la finalul unui ciclu de învățământ marchează, de fapt, și intrarea în viață activă (bacalaureatul, examenul de licență).

B. O altă taxonomie a formelor de evaluare rezultă din raportarea la următoarele criterii (Adrian Stoica, *Evaluarea progresului școlar*, Editura Humanitas Educațional, București, 2003, pag. 136-137):

- ➔ Reprezentativitatea populației școlare vizate;
- ➔ Domenii curriculare considerate importante
- ➔ Variația în timp a performanțelor școlare la o anumită vârstă de la o generație

la alta etc.

Rezultă:

➔ Studii internaționale de evaluare (TIMSS, PISA, PIRLS și altele), în cadrul cărora diferite țări desfășoară același tip de evaluare; se finalizează prin rapoarte internaționale și naționale;

➔ Evaluări naționale desfășurate pe eșantioane ale unei populații - țintă (de exemplu, NAEP, în SUA; evaluarea la clasa a IV-a, în România);

➔ Evaluări naționale organizate pentru întreaga populație școlară de o anumită vârstă.

C. După modul de integrare în procesul de învățământ, distingem următoarele moduri/tipuri de evaluare (I.T. Radu, C. Cucuș, D. Potolea- M. Manolescu):

➔ Evaluarea inițială, realizată la debutul unui program de instruire;

➔ Evaluarea formativă, realizată pe parcursul programului și integrată acestuia;

➔ Evaluarea sumativă, cumulativă, realizată la finalul programului.

Evaluarea inițială

Evaluarea inițială este realizată la începutul unui program de instruire și vizează, în principal: identificarea condițiilor în care elevii pot să se pregătească și să integreze optimal în activitatea de învățare, în programul de instruire care urmează. Are funcții *diagnostice și prognostice*, de pregătire a noului program de instruire.

Evaluarea formativă

Este acel tip de evaluare care se realizează pe tot parcursul unui demers pedagogic, „este frecventă sub aspect temporal și are ca finalitate remedierea lacunelor sau erorilor săvârșite de elevi; nu-l judecă și nu-l clasează pe elev; Compară performanța acestuia cu un prag de reușită stabilit dinainte” (Bloom; G. Meyer). Caracteristici: este o evaluare criterială, bazată pe obiectivele învățării; face parte din procesul educativ normal; acceptă „nereușitele” elevului, considerându-le momente în rezolvarea unei probleme; intervine în timpul fiecărei sarcini de învățare; informează elevul și profesorul asupra gradului de stăpânire a obiectivelor, ajutându-i pe aceștia să determine mai bine achizițiile necesare pentru a aborda sarcina următoare, într-un ansamblu secvențial; asigură o reglare a proceselor de formare a elevului; îndrumă elevul în surmontarea dificultăților de învățare; este continuă, analitică, centrată mai mult pe cel ce învață decât pe produsul finit.

Evaluarea formatoare

Evaluarea formatoare este din ce în ce mai mult invocată în ultima perioadă, în acord cu achizițiile științei și cu evoluțiile din planul teoriei și practicii educaționale.

Evaluarea formatoare este considerată forma desăvârșită a evaluării formative. Reprezintă o nouă etapă, superioară de dezvoltare a evaluării formative, care „va fi funcțională odată cu instaurarea obiectivului de asumare de către elevul însuși a propriei învățări: la început conștientizarea, eventual negocierea obiectivelor de atins și apoi integrarea de către subiect a datelor furnizate prin demersul evaluativ în administrarea propriului parcurs. *Evaluarea formatoare*, are drept scop: *promovarea activității de învățare ca motor motivațional pentru elev*, sprijin în conștientizarea metacognitivă, autoreglare” (J. Vogler,); valorizarea relației predare - învățare, articularea fazelor evaluării în funcție de eficacitatea pedagogică (G. Nunziati, 1980).

Evaluarea sumativă sau „certificativă”

Evaluarea sumativă se prezintă în cel puțin două variante/forme mai importante pentru demersul nostru: realizată la finalul unui capitol, unități de învățare, sistem de lecții, teză semestrială, sau finală sau de bilanț, realizată la încheierea unui ciclu școlar, al unui

nivel de studii etc.

Caracteristicile esențiale ale evaluării sumative:

- ➔ este determinată de contexte specifice;
- ➔ este construită de profesori și elevi, în funcție de criteriile convenite;
- ➔ acceptă negocierea în temeiul convingerii că evaluarea este în beneficiul învățării realizate de elev (Belair);
- ➔ evidențiază rezultatele învățării și nu procesele;
- ➔ este internă, dar de cele mai multe ori este externă (ex: capacitate, bacalaureat, diplomă etc.).

Evaluarea inițială, cea continuă și sumativă reunesc conceptual și practic funcțiile esențiale ale actului evaluativ.

6. METODE, TEHNICI, INSTRUMENTE DE EVALUARE

6.1. METODA DE EVALUARE

Reprezintă calea de acțiune pe care o urmează profesorul și elevii și care conduce la punerea în aplicare a oricărui demers evaluativ, în vederea colectării informațiilor privind procesul și produsul învățării, prelucrării și valorificării lor în diverse scopuri. Metodele de evaluare sunt importante în raport cu situațiile educaționale în care sunt folosite. Importanța lor se stabilește îndeosebi după modul de aplicare în situațiile cele mai potrivite.

Fiecare metodă, tehnică sau instrument de evaluare prezintă avantaje și dezavantaje. Ele vizează capacități cognitive diferite și, în consecință, nu oferă toate aceleași informații despre procesul didactic. Datorită acestui fapt dar și diversității obiectivelor activității didactice, nicio metodă și niciun un instrument nu pot fi considerate universal valabile pentru toate tipurile de competențe și toate conținuturile.

Urmărirea și verificarea complexă a realizării obiectivelor vizate în procesul de instruire și educație se obțin prin îmbinarea diferitelor metode, tehnici și instrumente de evaluare, prin folosirea, de fiecare dată, a celei mai potrivite.

Caracteristici generale:

- ➔ din perspectiva învățământului modern, predominant formativ, metodele de evaluare însoțesc și facilitează desfășurarea procesului instructiv - educativ. Într-un context de evaluare formativă, însoțesc și permit reglarea desfășurării procesului de învățământ;
- ➔ se elaborează și se aplică în strânsă legătură cu diferitele componente ale procesului de învățământ, aflate în ipostaza de „obiecte ale evaluării”;
- ➔ se concep, se îmbină și se folosesc în legătură cu particularitățile de vârstă și individuale, cu modul de acționare al factorilor educativi;
- ➔ au caracter dinamic, fiind deschise înnoirilor și perfecționărilor;
- ➔ au caracter sistemic: fără a-și pierde entitatea specifică, se îmbină, se completează și se influențează reciproc, alcătuind un ansamblu metodologic coerent;
- ➔ raporturile dintre ele se schimbă în funcție de context. Trebuie remarcate raporturile dinamice dintre aceste concepte. În diverse contexte educaționale unele dintre acestea pot fi metode prin intermediul cărora este condus procesul evaluativ, în timp ce în alte împrejurări pot deveni mijloace de culegere, prelucrare a informațiilor sau de comunicare socială profesor - elev.

Tipologia metodelor de evaluare:

Criteriul cel mai frecvent folosit în clasificarea metodelor de evaluare este cel cronologic/istoric. În funcție de acest criteriu, distingem:

1. *Metode tradiționale de evaluare:* evaluarea orală

- ➔ evaluarea scrisă
- ➔ evaluarea prin probe practice
- ➔ testul docimologic

2. *Metode alternative și complementare de evaluare:*

- ➔ observarea sistematică a comportamentului elevului față de activitatea școlară
- ➔ portofoliul
- ➔ investigația
- ➔ proiectul
- ➔ autoevaluarea etc.

Dintr-o perspectivă a evoluției evaluării spre procesele de învățare - „obiecte” specifice ale educației cognitive - se justifică pe deplin **complementaritatea** metodelor tradiționale și a celor alternative de evaluare, fiecare categorie dovedind virtuți și limite.

6.2. ITEMUL DE EVALUARE

Repere conceptuale

Reprezintă cea mai mică componentă identificabilă a unui test sau a unei probe de evaluare. Din punct de vedere științific, itemul este un element component al unui chestionar standardizat care vizează evaluarea elevului în condiții de maximă rigurozitate. În practica școlară cotidiană, unde nu poate fi vorba întotdeauna de evaluări realizate „în condiții de maximă rigurozitate”, itemii reprezintă elementele chestionarului sau probei unui examen sau ale unei evaluări normale, la clasă.

Tipologia itemilor de evaluare

A. După *criteriul asigurării obiectivității* în notarea sau aprecierea elevilor identificăm:

- *itemi obiectivi* care sunt de trei tipuri:
 - ➔ itemi cu alegere multiplă
 - ➔ itemi cu alegere duală
 - ➔ itemi tip pereche.
- *itemi semiobiectivi sau itemi cu răspuns construit scurt* cu următoarea tipologie:
 - ➔ itemi cu răspuns scurt, cu următoarele variante: întrebarea clasică, exercițiul, chestionarul cu răspunsuri deschise scurte, textul indus
 - ➔ itemi de completare, cu următoarele variante/forme: textul lacunar, textul „perforat”
 - ➔ întrebarea structurată.
- *itemi subiectivi*

Itemii subiectivi solicită răspunsuri dezvoltate, elaborate. Redactarea răspunsului solicită mobilizarea cunoștințelor și abilităților care iau forma unor structuri integrate și integrative. Solicitățile formulate de cadrul didactic și răspunsurile elevilor se caracterizează prin aspectul lor integrativ. Formularea răspunsului la un item subiectiv acoperă toate tipurile de obiective. Itemii subiectivi au următoarea tipologie:

- ➔ itemul cu răspuns scurt, puțin elaborat
- ➔ itemul tip rezolvare de probleme
- ➔ itemul tip eseu
- ➔ itemul cu răspuns elaborat/ dezvoltat

B. După operațiile implicate în elaborarea itemilor, diferențiem: *itemi de identificare, de selecționare, de elaborare, de construcție* etc. Itemii se integrează în instrumente de evaluare. Cadrul didactic are la dispoziție o mare varietate de tehnici și instrumente de evaluare, mergând de la cele care solicită tehnicile cele mai „închise” până la cele care permit exprimarea liberă a elevului. Itemii de evaluare trebuie folosiți în funcție de complexitatea obiectivelor vizate. Realizarea/construcția itemilor și a probelor de evaluare solicită o atitudine flexibilă din partea cadrului didactic. Fiecare instrument de evaluare, fiecare tip de item are avantaje și dezavantaje.

6.3. TEHNICILE DE EVALUARE

Constituie modalitățile prin care evaluatorul declanșează și orientează obținerea unor răspunsuri din partea subiecților, în conformitate cu obiectivele sau specificațiile probei. Fiecare tip de item declanșează o anumită tehnică la care elevul apelează pentru a da răspunsul său. Un item cu alegere multiplă (IAM) face apel la „tehnica răspunsului cu alegere multiplă”. Elevul va încercui, va bifa sau va marca printr-o cruciuliță varianta de răspuns pe care o consideră corectă. Un item tip „completare de frază” va face apel la „tehnica textului lacunar”. Elevul va completa spațiile libere din textul respectiv etc.

6.4. INSTRUMENTUL DE EVALUARE

Este o probă, o grilă, un chestionar, un test de evaluare care „colectează” informații, „produce” dovezi semnificative despre aspectele sau rezultatele luate în considerare. Instrumentul de evaluare se compune, de regulă, din mai mulți itemi. O probă de evaluare (un instrument) se poate compune dintr-un singur item (o singură întrebare, cerință, problemă etc, îndeosebi atunci când răspunsul pe care trebuie să-l formuleze elevul este complex) sau din mai mulți itemi.

Un instrument de evaluare integrează fie un singur tip de itemi (spre exemplu numai itemi cu alegere multiplă - IAM) și, în acest caz, constituie un „Chestionar cu alegere multiplă” (CAM), fie itemi de diverse tipuri, care solicită, în consecință, tehnici diverse de redactare, formulare sau prezentare a răspunsurilor.

Construcția probelor/instrumentelor de evaluare este o activitate laborioasă. Între complexitatea obiectivelor educaționale ce trebuie evaluate și „deschiderea „tehnicilor și instrumentelor de evaluare trebuie să funcționeze corespondențe progresive. Obiectivele se dezvoltă de la simplu la complex, iar instrumentele de evaluare se dezvoltă de la „închise” spre „deschise”. Există o puternică corelație între instrumentele de evaluare și operațiile evaluării (măsurarea, aprecierea, decizia). De asemenea sunt corelații importante între instrumentele de evaluare și strategiile/tipurile de evaluare, precum și între instrumente și metode. Fiecare operație, metodă, strategie etc. solicită instrumentul evaluativ cel mai potrivit.

EVALUAREA COMPETENȚELOR LA DISCIPLINA INFORMATICA

CURRICULUM ȘI EVALUARE LA DISCIPLINA INFORMATICA. FORMAREA ȘI EVALUAREA COMPETENȚELOR

Instrumentele informatice sunt implicate în majoritatea proceselor socio-economice și vin în ajutorul multor persoane, facilitându-le munca, accesul la informație și ajutându-le în organizarea activităților. Tot ele sunt adesea modalități de deconectare inteligentă și instructivă.

Punând la baza folosirii instrumentelor informatice o pregătire prealabilă conceptuală, metodologică și tehnologică se creează o arhitectură mentală capabilă să utilizeze cu precizie și siguranță oricare dintre instrumentele care funcționează pe aceleași principii, indiferent de marca, generația sau versiunea lor. Punând accentul în principal pe termeni și pe principii și doar secundar pe instrumentarul tehnologic existent, cadrul educațional al învățării informaticii asigură o formare reală, de „bătăie foarte lungă” pentru elevi. În acest sens se are în vedere faptul că informatica este o platformă științifică pentru ceea ce vor fi, vor face și vor utiliza elevii pe parcursul întregii vieți, în diferite domenii de activitate, și nu numai pentru cei ce doresc continuarea studiului informaticii la facultăți sau cursuri de specializare.

Obiectivele majore ale disciplinei Informatică sunt:

- dezvoltarea gândirii prin formarea gândirii algoritmice,
- organizarea datelor prin înțelegerea și utilizarea sistematică a structurilor de date și
- dobândirea abilităților de construire a aplicațiilor pornind de la cerințe concrete de prelucrare.

Instrumentarul informatic pus în slujba consolidării gândirii și sistematizării datelor este în același timp o manieră dinamică de învățare, o cale de aprofundare a studiului altor discipline.

Maniera științifică, riguroasă și sistematică a abordării unei situații problemă, indiferent de domeniul din care face parte aceasta, este un rezultat vizibil și măsurabil pe care informatica îl vizează. La informatică se folosesc echipamentele IT și limbajele de programare ca instrumente de verificare a corectitudinii rezolvărilor, pentru experimentele de implemetare și testare. Problemele alese pentru rezolvare pot avea fundamente conceptuale din matematică, fizică, chimie, biologie sau pot avea tangență cu teme de lingvistică sau socio-umane. Problemele „de viață” a căror rezolvare este descrisă algoritmic reprezintă modele comportamentale de zi cu zi pentru elev. Astfel, el este încurajat să analizeze, să etapeze și să eficientizeze un proces pe care îl poate organiza sau influența în scopul îmbunătățirii vieții.

Profesorului de informatică îi revine sarcina de a alege conținuturi, activități de învățare și strategii didactice aflate în acord cu aceste obiective, să urmărească și să regleze întregul său demers didactic pe baza rezultatelor evaluării. Utilizarea competențelor a instrumentelor și a formelor de evaluare este o premiză atât a obținerii unor informații

relevante privind calitatea actului didactic, cât și o pârgie motivațională a învățării.

Încurajând fundamentarea științifică și didactică a activităților de învățare la informatică, se formează o atitudine responsabilă și se creează un cadru emoțional propice învățării.

Tabelele următoare exprimă raportul dintre curriculum (competențele și conținuturile învățării precizate în programele școlare), formare (activitățile de învățare ce se pot organiza în vederea formării competențelor) și evaluare (competențele de evaluat derivate din competențele specifice și formele de evaluare propuse prin care se pot evalua competențele).

Pentru a evita repetările supărătoare (date de paralelismul programelor de informatică pentru specializările matematică intensiv și matematică-informatică, intensiv informatică și de abordările concentrice prezente în programe), nu s-a făcut o departajare pe ani de studiu și specializări, ci se prezintă o tratare verticală, pentru fiecare competență generală în parte.

Unele dintre competențele specifice din programă au fost reformulate, fără a altera esența lor și fără a afecta conținuturile de învățare aferente.

COMPETENȚE GENERALE

1. Identificarea conexiunilor dintre informatică și societate.

Sesizarea și aprecierea contribuției informaticii la viața socială și promovarea cu responsabilitate dezvoltării instrumentelor informatice.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
1.1. Identificarea aplicațiilor informatice în viața socială	Descrierea unei aplicații informatice pe care o cunoaște ca utilizator sau din vedere Enumerarea domeniilor în care au utilitate evidentă aplicațiile informatice Explicarea rolului aplicațiilor informatice în viața socială, în economie, în finanțe, în mass media etc. Descrierea modului de utilizarea a aplicațiilor informatice într-un domeniu, la alegere	Definirea informaticii ca știință Rolul informaticii în societate Studii de caz ale unor situații sociale, în abordare informatizată Reguli elementare pentru crearea și susținerea unei prezentări publice	Analizarea utilizării unei aplicații pu-blice Analizarea unor instrumente infor-mactice folosite în medicină Realizarea unei scheme cu utili-zarea aplicațiilor grafice în diverse domenii Descrierea sumară a unui algoritm de arhivare Analizarea impli-cațiilor algoritmilor de criptare etc.	Evaluare orală Eseu Referat
1.2. Recunoașterea situațiilor în care este necesară prelucrarea algoritmică a informațiilor	Enumerarea unor domenii de activitate în care informațiile sunt prelucrate algoritmic Identificarea datelor care se prelucrează într-un domeniu Identificarea formelor în care se obțin rezultatele Descrierea utilității rezultatelor obținute Exemplificarea și explicarea sumară a prelucrărilor algoritmice dintr-un domeniu economic (o bancă, un institut de cercetări etc.)			
1.3. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	Evidențierea posibilităților utilizării practice ale rezultatelor unei probleme algoritmice, ale unui program, ale unei aplicații (unui proiect) Enumerarea posibilităților beneficiari ai unei aplicații Descrierea proprietăților unui program/aplicație care dau acestuia o mai mare aplicabilitate practică			

2. Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea

Elevul trebuie să fie capabil să identifice, să clasifice, să structureze, să coreleze și să utilizeze datele.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
2.1. Identificarea și clasificarea datelor necesare pentru rezolvarea unei probleme	Identificarea într-un enunț a tuturor elementelor ce reprezintă date Reformularea sau completarea unui enunț dat pentru a specifica datele implicate Identificarea constantelor și variabilelor ce intervin în rezolvare	Date cu care lucrează algoritmi: → constante → variabile → expresii	Analizarea enunțului unei probleme în scopul identificării și clasificării datelor necesare	Probă scrisă Evaluare orală Proiect (prezentare)
2.2. Identificarea tipurilor de date necesare pentru rezolvarea unei probleme (de intrare, de ieșire, de manevră)	Identificarea numărului, numelui și valorilor posibile pentru datele de intrare și de ieșire Explicarea dependențelor existente între datele de intrare și cele de ieșire Identificarea datelor de manevră necesare în procesul de obținere a rezultatelor Completarea datelor de intrare într-un enunț lacunar Explicarea unui algoritmi dat prin redenumirea variabilelor Explicarea unui algoritmi dat introducând noi date de manevră Reducerea numărului datelor de manevră într-un algoritmi dat	Tipuri structurate de date : → tablouri: → șiruri de caractere → înregistrări → liste → stive → cozi	Completarea unui enunț lacunar cu cerința/ datele/ restricțiile impuse asupra datelor/ sintagmele de legătură etc. Exerciții de reprezentare a unor structuri de date precizate (șir, matrice, graf, arbore etc.)	Harta conceptuală Tehnica 3-2-1 R.A.I.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
2.3. Descrierea unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire	Identificarea pașilor de citire și afișare a datelor Identificarea pașilor de prelucrare a datelor Identificarea tuturor pașilor de prelucrare posibili la un moment dat Alegerea din mai multe posibilități de prelucrare a celei care corespunde cerinței Alegerea unei ordini de prelucrare a datelor care să corespundă cerinței Stabilirea valorii unor date specifice (rezultate, date de intrare pentru un rezultat specificat etc.) pentru un algoritmi dat Descrierea în limbaj natural a unor secvențe de prelucrare date Legarea coerentă a descrierilor pentru obținerea unui enunț	Structuri de date alocate dinamic → liste simple → înălțuite → liste dublu înălțuite → liste circulare Grafuri orientate și neorientate: → terminologie → proprietăți → reprezentarea în memorie → tipuri speciale de grafuri	Folosirea termenilor specifici structurilor de date în diferite contexte (studiu, demonstrare, exersare). Exemplificarea principiilor proprietăților ale unor structuri de date Realizarea unei scheme de clasificare a datelor și structurilor de date.	Probă scrisă Evaluare orală Proiect (prezentare) Harta conceptuală Tehnica 3-2-1 R.A.I.
2.4. Evidențierea necesității structurării datelor	Identificarea situațiilor problemă în care nu sunt suficiente datele simple Alegerea din mai multe enunțuri a celor în care sunt suficiente datele simple Alegerea din mai multe enunțuri a celor care necesită date structurate	Structuri de date arborescente: → terminologie → proprietăți → reprezentarea în memorie, tipuri speciale de arbori		

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
2.5. Identificarea relațiilor dintre date	Enumerarea rezultatelor ce se pot obține dintr-un set de date Stabilirea dependențelor existente între datele de intrare și cele de ieșire Reprezentarea grafică a dependențelor existente între date Identificarea compatibilității de tipuri necesară între date într-o prelucrare Descrierea în limbaj natural a modului de organizare a datelor într-o structură Clasificarea structurilor de date după tipul datelor, durata de viață, acces etc. Modelarea datelor și relațiilor existente între acestea cu ajutorul diagramelor entități-relații	Modelul conceptual al problemei Tipuri de date Relații între entități Modele de organizare a datelor		
2.6. Utilizarea structurilor de date în modelarea unor situații problemă	Identificarea structurilor de date necesare pentru memorarea unor date Identificarea unui element al structurii Identificarea operațiilor care se pot efectua asupra structurii Utilizarea formelor specifice de reprezentare a grafurilor Utilizarea formelor specifice de reprezentare a listelor Utilizarea formelor specifice de reprezentare a arborilor Realizarea diagramelor entități-relații pentru o problemă de gestiune dată	Tipuri structurate de date tablouri, șiruri de caractere, înregistrări, liste, stive, cozi Structuri de date alocate dinamic (liste simplu înlănțuite, liste dublu înlănțuite, liste circulare)	Analizarea enunțului unei probleme în scopul identificării și clasificării datelor necesare	

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
2.7. Alegerea structurii de date adecvate rezolvării unei probleme	Enumerarea diferitelor modalități de structurare a datelor pentru o problemă Alegerea din mai multe structurări posibile a celei adecvate	Grafuri orientate și neorientate: → terminologie → proprietăți → reprezentarea în memorie → tipuri speciale de grafuri	Completerea unui enunț lacunar cu cerința/ datele/ restricțiile impuse asupra datelor/ sintagmele de legătură etc.	Probă scrisă Evaluare orală Proiect (prezentare) Harta conceptuală Tehnica 3-2-1 R.A.I.
2.8. Transpunerea unei probleme din limbaj natural în limbaj ce folosește corect terminologia specifică grafurilor	Identificarea elementelor unui enunț care sugerează utilizarea teoriei grafurilor Identificarea elementelor compatibile cu teoria grafurilor dintr-o implementare dată Reformularea un enunț folosind termeni din teoria grafurilor Formularea un enunț natural corespunzător unei sintagme date din teoria grafurilor	Structuri de date arborescente: → terminologie → proprietăți → reprezentarea în memorie, tipuri speciale de arbori	Folosirea termenilor specifici structurilor de date în diferite contexte (studiu, demonstrare, exersare). Exemplificarea principalelor proprietăți ale unor structuri de date	
2.9. Identificarea avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme	Rezolvarea unei probleme folosind două sau mai multe modalități de structurare a datelor Enumerarea avantajelor alegerii unei anumite structuri de date Descrierea fiecăruia dintre avantajele alegerii unei anumite structuri de date (eficiența de lucru, expresivitatea sursei, posibilități de depanare etc.)	Modelul conceptual al problemei Tipuri de date Modele de organizare a datelor	Realizarea unei scheme de clasificare a datelor și structurilor de date.	

3. Elaborarea algoritmilor de rezolvare a problemelor

Elevul trebuie să urmărească algoritmul de rezolvare a unei probleme, să descrie o metodă algoritmică de rezolvare, să reprezinte o strategie de rezolvare într-o manieră științifică și expresivă (pseudocod) și să coreleze o situație practică de prelucrare cu unul dintre algoritmi cunoscuți.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
3.1. Identificarea elementelor unui enunț	Identificarea datelor și rezultatelor unei probleme Identificarea restricțiilor impuse asupra datelor Sublinierea unor cuvinte esențiale necesare pentru rezolvare Sublinierea unor sintagme caracteristice unui mare număr de probleme (distincte, cel mai ... , eficient , toate etc.) Reformularea enunțului unei probleme (pentru corectitudine, completitudine, claritate, expresivitate, eficiență etc.)	Etapile rezolvării problemelor. Exemple. Noțiunea de algoritm. Caracteristici. Exemple. Reprezentarea algoritmilor. Pseudocod. Principiile programării structurate. Structuri de bază: → structura liniară → structura alternativă → structura repetitivă	Simularea executării unui algoritm dat Analizarea unui flux de prelucrare în vederea structurării algoritmului corespunzător Exerciții de completare, modificare sau restructurare a unui algoritm dat Scrierea în pseudocod a algoritmilor fundamentali.	Teste scrise (temi obiective și semioiective) Rezolvare de probleme Portofoliu de probleme Proiect de modelare Hărți conceptuale 3-2-1 R.A.I.
3.2. Descompunerea rezolvării unei probleme în pași	Identificarea primului pas al unui algoritm Identificarea următorului pas posibil al unui algoritm dat parțial Identificarea pasului lipsă al unui algoritm dat Ordonarea pașilor unui algoritm dat (neordonat) Formularea în limbaj natural a rolului unui pas dintr-o prelucrare algoritmică Recunoașterea succesiunii pașilor unui algoritm într-o reprezentare printr-o schemă logică dată	Algoritmi elementari: → cifrele unui număr → probleme de divizibilitate expresii simple → secvențe de valori → șiruri recurente	Adaptarea dirijată a unor algoritmi pentru a rezolva cerințe noi Exerciții de estimare a numărului de operații corespunzătoare unui algoritm	

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
3.3. Reprezentarea algoritmilor în pseudocod	Transcrierea în pseudocod a rezolvării descrișe în limbaj natural sau dată prin schemă logică structurată Alinierea sugestivă a instrucțiunilor unui program pseudocod	Aplicații din viața cotidiană : → situația școlare → balanța de cheltuieli → cheltuieli la întreținere → salariul unei persoane etc.	Reprezentarea prin diagramă top-down a rezolvării unei probleme care se poate descompune în subprobleme Rezolvarea dirijată a unei probleme cu subprobleme secvențiale (independente)	Teste scrise (temi obiective și semioiective) Rezolvare de probleme
3.4. Respectarea principiilor programării structurate în elaborarea algoritmilor	Utilizarea tehnicilor de structurare a unei rezolvări (folosirea unor semnalizatori, schimbarea ordinii unor pași, refolosirea unor pași, schimbarea unor condiții etc.) Alegerea tehnicii adecvate de structurare a unei rezolvări Scrierea indentată a instrucțiunilor pentru a evidenția structurile folosite	Aplicații interdisciplinare, specifice profilului: → ecuația de gradul I → ecuația de gradul al II-lea → fracții → aplicații de geometrie → progresii → mobile în mișcare → masa moleculară	Rezolvarea dirijată a unei probleme cu subprobleme dependente Exerciții de calcul al valorii unei funcții recursive și de urmărire a executării unei proceduri recursive Reprezentarea prin diagramă a apelurilor recursive	Portofoliu de probleme Proiect de modelare Hărți conceptuale 3-2-1 R.A.I.
3.5. Aplicarea creativă a algoritmilor fundamentali în rezolvarea unor probleme concrete	Reproducerea enunțului unei probleme concrete-practice Evidențierea elementelor din enunț care sugerează un algoritm fundamental Evidențierea elementelor din enunț care-l diferențiază de aplicarea directă a algoritmului sugerat Adaptarea algoritmului pentru a răspunde cerințelor unei probleme noi Implementarea algoritmului adaptat Combinarea a doi sau mai mulți algoritmi pentru a rezolva o problemă concretă			

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
	Formularea unei cerințe cu caracter practic ce utilizează în rezolvare un algoritm dat Formularea unei cerințe cu caracter practic în rezolvarea căreia se utilizează succesiv doi algoritmi dați Formularea unei cerințe cu caracter practic ce combină doi algoritmi dați	<ul style="list-style-type: none"> ➤ prelucrări statistice ale unei serii de valori ➤ valoarea unei expresii algebrice ➤ combinatorică ➤ mărimi fizice într-un circuit electric generică etc.) 	Realizarea tabelului de valori pentru parametrii și variabilele locale ale subprogramelor Exersarea aplicării metodelor de programare pe seturi de date concrete	
3.6. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării	Rezolvarea problemei pentru exemple concrete de date alese judicios Identificarea datelor specifice problemei (stabilirea numărului de date necesare, a numărului de rezultate solicitate, a mulțimii de valori pentru date, a denumirilor expresive pentru variabile etc.) Descrierea rezolvării în limbaj natural Descrierea rezolvării în pseudocod Aplicarea pașilor unui algoritm pentru seturi de date rezonabile	Metode de programare Analiza eficienței unui algoritm Subprograme. Aplicații folosind subprograme Modularizarea aplicației Etapă în dezvoltarea aplicațiilor	Urmărirea executării unui algoritm backtracking dat Scrierea domeniilor de valori și a condițiilor de validare pentru un algoritm backtracking	Teste scrise (temi obiective și semioiective) Rezolvare de probleme Portofoliu de probleme
3.7. Alegerea unui algoritm eficient de rezolvare a unei probleme	Alegerea unui algoritm de rezolvare dintr-o mulțime de algoritmi dați, conform unui criteriu Estimarea eficienței, ca timp de executare, a unei rezolvări Estimarea eficienței, ca spațiu de memorie utilizat, a unei rezolvări Modificarea unui algoritm pentru a îndeplini o anumită condiție de eficiență (număr variabile, număr structuri repetitive, număr de executări ale unui pas etc.)	Subprograme. Aplicații folosind subprograme Modularizarea aplicației Etapă în dezvoltarea aplicațiilor	Discutarea și proiectarea a două sau mai multe rezolvări (bazate pe metode diferite) pentru aceeași problemă și analizarea eficienței fiecăreia dintre ele.	Proiect de modelare Hărți conceptuale 3-2-1 R.A.I.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
3.8. Identificarea subproblemelor unei probleme	Identificarea unei cerințe simple deductibile din cerința compusă a unei probleme Asamblarea unei cerințe compuse, pe baza a două sau mai multe cerințe simple date Descompunerea unei cerințe compuse în cerințe simple din care se compune Crearea de enunțuri care să solicite doi sau mai mulți algoritmi cunoscuți	Etapă în dezvoltarea aplicațiilor Criterii de eficiență a aplicațiilor	Realizarea în echipă a modelului unei aplicații simple cu baze de date sau cu clase și obiecte	Teste scrise (temi obiective și semioiective) Rezolvare de probleme Portofoliu de probleme
3.9. Utilizarea subprogramelor	Recunoașterea situațiilor în care este necesară utilizarea unor subprograme Descompunerea unei cerințe compuse în cerințe simple din care se compune Compararea programului ce implementează un algoritm cu programul ce implementează același algoritm cu ajutorul subprogramelor Recunoașterea avantajelor de editare și depănare a programelor ce utilizează subprograme Recunoașterea metodelor de refolosire a codului unei secvențe de prelucrare Identificarea datelor de intrare, a datelor de ieșire și a celor de intrare-ieșire într-o prelucrare modulară Diferențierea definirii unui subprogram de apelul celui subprogram Diferențierea unui subprogram procedural de unul funcțional Diferențierea transferului prin valoarea de transfer prin referință	Subprograme. Aplicații folosind subprograme Recursivitate Structura unei aplicații Modularizarea aplicației Etapă în dezvoltarea aplicațiilor	Realizarea tabelului de valori pentru parametrii și variabilele locale ale subprogramelor Exersarea aplicării metodelor de programare pe seturi de date concrete Discutarea și proiectarea a două sau mai multe rezolvări (bazate pe metode diferite)	Teste scrise (temi obiective și semioiective) Rezolvare de probleme Portofoliu de probleme Proiect de modelare Hărți conceptuale 3-2-1 R.A.I.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
	Identificarea variabilelor locale, variabilelor globale și a parametrilor Enumerarea avantajelor în lucrul cu subprograme	Criterii de eficiență a aplicațiilor	pentru aceeași problemă și analizarea eficienței fiecăreia dintre ele.	
3.10. Utilizarea recursivității	Diferențierea definiției directe în raport cu definiția recurentă a unui șir Descrierea metodei de rezolvare a unei probleme în termeni recursivi Evaluarea unui apel recursiv Reprezentarea stivei de apeluri în diferite puncte ale executării unui apel recursiv Completarea condiției/condițiilor de oprire pentru o definiție de subprogram recursiv Completarea apelului recursiv al unui subprogram cu o cerință dată Reprezentarea liniară/arborescentă a apelurilor recursive pentru un apel dat		Exersarea aplicării metodelor de programare pe seturi de date concrete Discutarea și proiectarea a două sau mai multe rezolvări (bazate pe metode diferite) pentru aceeași problemă și analizarea eficienței fiecăreia dintre ele	Teste scrise (itemi obiectivi și semioiectivi) Rezolvare de probleme Portofoliu de probleme

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
3.11. Identificarea metodei de programare adecvate pentru rezolvarea unei probleme	Caracterizarea metodelor studiate (Backtracking, Divide et impera, Greedy, Programare dinamică) Identificarea elementelor din cerință (criteriu de optim, subprobleme, independență, succesiune de decizii etc.) ce încadrează problema într-o clasă dintre cele studiate Descrierea etapelor de rezolvare a unei probleme din fiecare clasă Identificarea unor probleme cunoscute dintr-o clasă dată Clasificarea unui set de probleme date în mulțimi corespunzătoare metodelor învățate			Proiect de modelare Hărți conceptuale 3-2-1 R.A.I.
3.12. Construirea unor soluții pentru probleme simple care se rezolvă cu ajutorul metodelor de programare	Descrierea rezolvării de principiu pentru fiecare metodă. Descrierea rezolvării backtracking pentru o problemă cunoscută Descrierea rezolvării divide et impera pentru o problemă cunoscută Descrierea rezolvării greedy pentru o problemă cunoscută	Aplicații practice folosind tablouri, șiruri de caractere, înregistrări, liste, grafuri, arbori. Subprograme. Aplicații folosind subprograme recursive	Realizarea în echipă a modelului unei aplicații simple cu clase și obiecte.	
3.13. Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme practice	Reproducerea enunțului unei probleme cu caracter practic/interdisciplinar Stabilirea analogiei unei probleme cu o problemă cunoscută Adaptarea rezolvării unei probleme cunoscute ce utilizează una dintre metode pentru a răspunde cerințelor unei probleme noi Proiectarea unei aplicații practice/interdisciplinare simple ce implementează una dintre metodele studiate	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) Structura unei aplicații Modularizarea aplicației Etapă în dezvoltarea aplicațiilor Criterii de eficiență a aplicațiilor		Teste scrise (itemi obiectivi și semioiectivi) Rezolvare de probleme Portofoliu de probleme

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
3.14. Analiza comparativă a eficienței diferitelor metode de rezolvare a aceleiași probleme	Enumerarea metodelor corecte de rezolvare pentru o aceeași problemă și ordonarea acestora în funcție de eficiență Alegerea unui algoritm eficient de rezolvare a unei probleme		Exersarea aplicării metodelor de programare pe seturi de date concrete	Proiect de modelare Hărți conceptuale 3-2-1, R.A.I.
3.15. Elaborarea strategiei de realizare a unei aplicații	Enumerarea etapelor realizării unui proiect Enumerarea activităților specifice fiecărei etape și a condițiilor specifice de desfășurare Planificarea și executarea etapelor realizării unei aplicații. Documentarea etapei de proiectare Gestionarea instrumentelor, produselor și versiunilor în etapa de implementare Respectarea termenelor de realizare		Discutarea și proiectarea a două sau mai multe rezolvări (baze pe metode diferite) pentru aceeași problemă și analizarea eficienței fiecăreia dintre ele.	

4. Aplicarea algoritmilor fundamentali în prelucrarea datelor

Elevul trebuie să cunoască particularitățile de prelucrare a diferitelor tipuri de structuri de date pentru a fi capabil să modeleze complet rezolvarea unei probleme simple, prin alegerea structurilor de date și a instrumentelor de prelucrare adecvate.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
4.1. Aplicarea algoritmilor fundamentali pentru date simple	Recunoașterea unor algoritmi ce prelucrează cifrele unui număr, de divizibilitate, de calcul etc. Adaptarea unor algoritmi dați pentru a răspunde unor cerințe suplimentare de prelucrare a datelor Reprezentarea pseudocod a unor algoritmi cu date simple	Operații asupra datelor: ➡ aritmetice ➡ logice ➡ relaționale	Aplicarea unor algoritmi de prelucrare pe structuri de date concrete cu respectarea pașilor de scriși în limbaj natural	Teste scrise (itemi obiectivi și semio- biectivi) Rezolvare de probleme

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
4.2. Aplicarea funcțiilor și procedurilor specifice de prelucrare a datelor structurate	Enumerarea prelucrărilor uzuale în lucrul cu șiruri de caractere Descrierea formală a apelului unui subprogram de prelucrare a șirurilor de caractere Enunțarea rolului parametrilor de intrare ai fiecăruia dintre subprograme Identificarea modului de obținere a rezultatului/rezultatelor prelucrărilor efectuate de subprogramele predefinite Utilizarea corectă a parametrilor subprogramelor predefinite	Prelucrare a datelor structurate în tablouri: ➡ căutare ➡ sortare ➡ prelucrări de matrice etc. Prelucrarea șirurilor de caractere și a înregistrărilor	Modificarea unui algoritm cunoscut pentru realizarea unei prelucrări înrudite (ex. Sortarea cu dublu criteriu) Construirea prin des-coperire a algoritmilor simpli de prelucrare	Portofoliu de probleme Proiect de prezentare animată/interactivă a prelucrărilor pe structuri speciale (stiva, heap etc.)
4.3. Elaborarea unor algoritmi de prelucrare a datelor structurate	Descrierea în limbaj natural a algoritmilor tipici pentru date structurate (parcursere, căutare, sortare, inserare etc.) Aplicarea unor metode de prelucrare pe seturi de date concrete Modificarea unui algoritm pentru un număr bine determinat de date simple astfel încât să opereze asupra unui șir de date, a unui număr bine determinat de șiruri astfel încât să opereze asupra unei matrice sau care utilizează diferite de valori pentru o aceeași entitate astfel încât să opereze asupra unui articol/unei înregistrări Construirea algoritmilor care să efectueze operații tipice asupra structurilor de date	Operații elementare pe liste în-lănțuite: ➡ inserare element ➡ ștergere element ➡ parcursere Parcurserea grafurilor. Aplicații: ➡ componente conexe ➡ tare conexe Determinarea drumurilor de	Declararea formală a unor subprograme predefinite pentru prelucra-rea structurilor de date Exerciții de reprezentare pseudocod a algoritmilor tipici de prelucrare a structurilor de date.	3-2-1 R.A.I.

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
4.4. Descrierea operațiilor specifice listelor și listelor în lanțuite	<p>Enumerarea operațiilor specifice listelor</p> <p>Recunoașterea operațiilor specifice listelor</p> <p>Identificarea și caracterizarea listelor speciale (coada și stiva)</p> <p>Utilizarea terminologiei specifice listelor și listelor speciale</p> <p>Descrierea unei prelucrări de liste în termeni de operații specifice</p> <p>Aplicarea unor metode de prelucrare a unor liste date</p> <p>Recunoașterea unui algoritm de inserare/eliminare în/ din listă folosind structuri statice</p> <p>Enunțarea principalelor mecanisme de alocare/deallocare a componentelor unei liste în lanțuite</p> <p>Descrierea în termeni de alocare/deallocare a operațiilor specifice cu liste</p> <p>Recunoașterea contextelor de prelucrare în care se recomandă utilizarea listelor în lanțuite</p>	<p>cost minim</p> <p>Arbori parțiali de cost minim</p> <p>Operații asupra datelor:</p> <p>➔ aritmetice</p> <p>➔ logice</p> <p>➔ relaționale</p> <p>Prelucrare a datelor structurate în tablouri:</p> <p>➔ căutare</p> <p>➔ sortare</p> <p>➔ prelucrări de matrice etc.</p>	<p>Urmărirea executării unui program pseudocod pentru o problemă ce combină mai multe prelucrări asupra aceleiași structuri de date.</p> <p>Rezolvarea dirijată a unei probleme ce combină structuri diferite de date și algoritmi fundamentali de prelucrare</p> <p>Realizarea unui scenariu de exploatare a unei baze de date simple (modelate anterior) prin definirea unor interogări cu utilitate practică</p>	<p>Teste scrise (itemi obiectivi și semioiectivi)</p> <p>Rezolvare de probleme</p> <p>Portofoliu</p> <p>de probleme</p> <p>Proiect de prezentare animată/interactivă a prelucrărilor pe structuri speciale (stiva, heap etc.)</p>
4.5. Construirea unor algoritmi simpli de verificare a unor proprietăți specifice grafurilor	<p>Descrierea în limbaj natural a algoritmilor de verificare a proprietății de graf neorientat</p> <p>Aplicarea unor metode de verificare pentru grafuri date</p> <p>Reprezentarea unor algoritmi de verificare prin calculul gradelor vârfurilor (graf complet, graf regulat), prin parcurgere (graf conex, tare conex) etc.</p> <p>Descrierea unei metode pentru verificare proprietății de lanț/ciclu/graf hamiltonian, de ciclu/graf eulerian</p> <p>Reprezentarea în pseudocod a unor secvențe specifice de verificare</p>	<p>Operații elementare pe liste în lanțuite:</p> <p>➔ inserare element</p> <p>➔ ștergere element</p> <p>➔ parcurgere</p>	<p>Construirea prin descriptoare a algoritmilor simpli de prelucrare</p>	<p>3-2-1 R.A.I.</p>

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
4.6. Aplicarea algoritmilor fundamentali de prelucrare a grafurilor	<p>Utilizarea corectă a termenilor specifici</p> <p>Descrierea unei metode de obținere a unui arbore parțial</p> <p>Descrierea uneia sau mai multor metode pentru drumuri optime</p> <p>Aplicarea unor metode de prelucrare pentru grafuri date</p> <p>Reprezentarea în pseudocod a unor secvențe specifice de prelucrare</p>	<p>Parcurea grafurilor. Aplicații:</p> <p>➔ componente conexe</p> <p>➔ tare conexe</p> <p>Determinarea drumurilor de cost minim</p> <p>Arbori parțiali de cost minim</p>	<p>Rezolvarea dirijată a unei probleme ce combină structuri diferite de date și algoritmi fundamentali de prelucrare</p>	
4.7. Descrierea operațiilor specifice structurilor arborescente	<p>Utilizarea corectă a termenilor specifici</p> <p>Enunțarea principalelor proprietăți ale unei structuri arborescente</p> <p>Aplicarea unor metode de determinare a caracteristicilor unor arbori dați</p> <p>Aplicarea unor metode de prelucrare pentru arbori dați</p> <p>Construirea algoritmilor de determinare a unor caracteristici (rădăcină, număr frunze, înălțime, număr maxim de fii etc.)</p> <p>Utilizarea parcurgerii grafurilor pentru prelucrări specifice structurilor arborescente</p> <p>Descrierea metodelor de parcurgere a arborilor binari</p> <p>Reprezentarea în pseudocod sau limbaj de programare a metodelor de parcurgere și a altor secvențe de prelucrare arborescentă</p>	<p>Operații asupra datelor:</p> <p>➔ aritmetice</p> <p>➔ logice</p> <p>➔ relaționale</p> <p>Prelucrare a datelor structurate în tablouri:</p> <p>➔ căutare</p> <p>➔ sortare</p> <p>➔ prelucrări de matrice etc.</p> <p>Prelucrarea șirurilor de caractere și a înregistrărilor</p> <p>Operații elementare</p>	<p>Realizarea unui scenariu de exploatare a unei baze de date simple (modelate anterior) prin definirea unor interogări cu utilitate practică</p>	<p>Teste scrise (itemi obiectivi și semioiectivi)</p> <p>Rezolvare de probleme</p> <p>Portofoliu</p> <p>de probleme</p> <p>Proiect de prezentare animată/interactivă a prelucrărilor pe structuri speciale (stiva, heap etc.)</p>

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
4.8. Descrierea operațiilor specifice bazelor de date	<p>Descrierea etapelor construirii unei baze de date</p> <p>Aplicarea metodelor de creare, actualizare și ștergere a unei tabele</p> <p>Aplicarea interogărilor simple, cu condiții de selecție și/sau cu opțiuni de ordonare</p> <p>Descrierea modalităților de obținere a datelor din mai multe tabele legate</p> <p>Aplicarea interogărilor cu legături</p> <p>Descrierea modalităților de obținere a datelor agregate</p> <p>Aplicarea funcțiilor agregate, a metodelor de grupare și selecție a grupurilor</p>	<p>tare pe liste înlănțuite:</p> <p>➔ inserare element</p> <p>➔ ștergere element</p> <p>➔ parcurgere</p> <p>Parcurgerea grafurilor. Aplicații:</p> <p>➔ componente conexe</p> <p>➔ tare conexe etc.</p> <p>Determinarea drumurilor de cost minim</p> <p>Arbori parțiali de cost minim</p>		3-2-1 R.A.I.

5. Implementarea algoritmilor într-un limbaj de programare

Elevul trebuie să se familiarizeze cu instrumentele de programare și cele de dezvoltare pentru a fi capabil să realizeze un produs program care să aibă toate atributele unei aplicații

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
5.1. Identificarea într-un program a structurilor de control învățate	<p>Identificarea instrucțiunilor simple (citire, scriere, atribuire)</p> <p>Identificarea structurilor de control (instrucțiunilor structurate)</p> <p>Enunțarea rolului instrucțiunii compuse în conținutul sintactic al structurilor de control</p>	<p>Elementele de bază ale limbajului de programare</p> <p>Mediul limbajului de programare studiat</p> <p>Fișiere text.</p>	<p>Editarea în mediul de dezvoltare a unui program dat</p> <p>Rularea pas cu pas a unui program (fișă de instruire asistată)</p> <p>Exersarea instrumentelor de urmărire și depanare pe un program simplu ce conține erori</p>	<p>Teste scrise (itemi obiectivi și semiobiectivi)</p> <p>Fișe de activitate practică</p> <p>Portofoliu de programe</p> <p>Investigația</p>
5.2. Utilizarea unui mediu de programare studiat	<p>Descrierea interfeței mediului de dezvoltare utilizat</p> <p>Enumerarea principalelor facilități oferite de mediul de dezvoltare utilizat</p> <p>Aplicarea pașilor de gestionare a unui program (simplu sau proiect de tip consolă)</p> <p>Utilizarea facilităților de editare ale mediului de dezvoltare</p> <p>Descrierea rolului etapei de compilare a programelor; Identificarea instrumentelor de depanare oferite de mediul de dezvoltare; Aplicarea principalelor operații de configurare a mediului de dezvoltare utilizat; Identificarea instrumentelor de autodocumentare oferite</p>	<p>Implementarea unor algoritmi elementari cu aplicabilitate practică</p> <p>Declararea și utilizarea structurilor de date.</p> <p>Aplicații practice folosind tablouri, șiruri de caractere, înregistrări, liste, grafuri, arbori.</p>	<p>Rularea mai multor implementări pentru o aceeași problemă cu completarea unor fișe de observații</p> <p>Elaborarea și realizarea unei aplicații simple (modelate anterior), folosind un mediu de programare specific</p>	<p>Referatul de prezentare a unui mediu sau a unei aplicații noi</p> <p>Proiectul</p>

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
5.3. Transcrierea algoritmilor din pseudocod într-un limbaj de programare	Recunoașterea elementelor unei descrieri formale (diagramă Wirth și/sau sintagmă cu terminale și neterminale) Enumerarea părților obligatorii și opționale ale unui program Identificarea rolului unei biblioteci Recunoașterea mecanismelor de utilizare a unei biblioteci Enunțarea rolului unei instrucțiuni de declarare Reproducerea sintagmei fiecărei instrucțiuni executabile din program Aplicarea principalelor reguli de scriere a expresiilor (operatori, compatibilități, priorități, evaluare) Enunțarea mecanismului de buffering Aplicarea mecanismelor specifice de citire și de scriere Aplicarea pașilor de gestionare a fișierelor-program specifice mediului de dezvoltare Aplicarea pașilor de obținere a unui program	Subprograme. Aplicații folosind subprograme Recursivitate Alocarea dinamică a memoriei (operații și mecanisme specifice) Elemente de programare orientată pe obiecte Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice)	Implementarea unor algoritmi fundamentali (unor metode) Implementarea algoritmilor specifici de prelucrare a structurilor de date Învățarea prin descoperire dirijată a tehnicilor moderne de editare oferite de un mediu nou, la prima lui utilizare Prezentarea unei aplicații în fața clasei (pe echipe)	
5.4. Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor	Identificarea rolului mecanismului de buffering Recunoașterea variabilelor asociate unui fișier text Explicarea rolului principalelor operații cu fișiere text (deschidere, citire/scriere, închidere) Identificare altor operații cu fișiere text (ștergere, redenumire, testare etc.) Aplicarea principalelor operații cu fișiere text	Structura unei aplicații Modularizarea aplicației Etapă în dezvoltarea aplicațiilor Criterii de eficiență a aplicațiilor	Exersarea instrumentelor de urmărire și depanare pe un program simplu ce conține erori	

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
5.5. Implementarea în limbaj de programare a unor algoritmi cu date simple și tablouri Enumerarea subprogramelor predefinite pentru lucrul cu șiruri de caractere Apelarea corectă a subprogramelor predefinite pentru șiruri de caractere Utilizarea documentației (help) pentru actualizarea cunoștințelor pentru lucrul cu șiruri de caractere Implementarea subprogramelor corespunzătoare principalelor operații cu liste și cu liste speciale Implementarea unor programe pentru prelucrări de liste și sau de liste speciale Implementarea algoritmilor de verificare a unor proprietăți ale grafurilor/arborilor Implementarea unor algoritmi specifici de prelucrare a grafurilor/arborilor	Implementarea în limbaj de programare a unor algoritmi cu date simple și tablouri Enumerarea subprogramelor predefinite pentru lucrul cu șiruri de caractere Apelarea corectă a subprogramelor predefinite pentru șiruri de caractere Utilizarea documentației (help) pentru actualizarea cunoștințelor pentru lucrul cu șiruri de caractere Implementarea subprogramelor corespunzătoare principalelor operații cu liste și cu liste speciale Implementarea unor programe pentru prelucrări de liste și sau de liste speciale Implementarea algoritmilor de verificare a unor proprietăți ale grafurilor/arborilor Implementarea unor algoritmi specifici de prelucrare a grafurilor/arborilor	Implementarea unor algoritmi elementari cu aplicabilitate practică Declararea și utilizarea structurilor de date. Aplicații practice folosind tablouri, șiruri de caractere, înregistrări, liste, grafuri, arbori. Subprograme. Aplicații folosind subprograme Recursivitate	Rularea mai multor implementări pentru o aceeași problemă cu completarea unor fișe de observare Elaborarea și realizarea unei aplicații simple (modelate anterior), folosind un mediu de programare specific Implementarea unor algoritmi fundamentali (unor metode)	Teste scrise (itemi obiectivi și semiobiectivi) Fișe de activitate practică Portofoliu de programe Investigația Referatul de prezentare a unui mediu sau a unei aplicații noi Proiectul
5.6. Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator	Apelarea corectă a unor subprograme predefinite pentru numere, pentru prelucrarea șirurilor de caractere etc. Descrierea parametrilor și rezultatului pentru principalele funcții predefinite Clasificarea funcțiilor predefinite în funcție de specificul prelucrării (numerice, pentru caractere, de conversie etc.) Utilizarea apelului unor proceduri predefinite (ștergerea ecranului, citire, scriere) Diferențierea apelului unei funcții de apelul unei proceduri Deducerea utilității unui subprogram a cărui definiție este dată Definirea unei funcții pentru un rezultat cerut	Alocarea dinamică a memoriei (operații și mecanisme specifice) Elemente de programare orientată pe obiecte Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) Structura unei aplicații Modularizarea aplicației	Implementarea algoritmilor specifici de prelucrare a structurilor de date Învățarea prin descoperire dirijată a tehnicilor moderne de editare oferite de un mediu nou, la prima lui utilizare Prezentarea unei aplicații în fața clasei (pe echipe)	

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
	Definirea unei proceduri pentru o prelucrare cerută Apelul unui subprogram a cărui declarație (proptotip) este dată	Etape în dezvoltarea aplicațiilor Criterii de eficiență a aplicațiilor	Exersarea instrumentelor de urmărire și depanare pe un program simplu ce conține erori	
5.7. Construirea unor subprograme pentru rezolvarea subproblemelor unei probleme	Implementarea corectă a definiției și apelurilor unui subprogram pentru o cerință dată Depanarea și corectarea unor subprograme ce conțin erori Implementarea unui program pentru o cerință dată utilizând unul sau mai multe subprograme definite de utilizator	Implementarea unor algoritmi elementari cu aplicabilitate practică	Rularea mai multor implementări pentru o aceeași problemă cu completarea unor fișe de observații Elaborarea și realizarea unei aplicații simple (modelate anterior), folosind un mediu de programare specific	
5.8. Aplicarea mecanismului recursivității prin crearea unor subprograme recursive	Apelarea unor programe recursive definite Deducerea rolului unui program recursiv dat prin definiția sa Implementarea unor subprograme recursive pentru cerințe simple Implementarea unor programe ce utilizează subprograme recursive Definirea unor subprograme recursive pentru prelucrări de liste înlanțuite/grafuri/arbori Definirea unor subprograme recursive pentru proprietăți ale grafurilor	Declararea și utilizarea structurilor de date. Aplicații practice folosind tablouri, șiruri de caractere, înregistrări, liste, grafuri, arbori. Subprograme. Aplicații folosind subprograme Recursivitate	Implementarea unor algoritmi fundamentali (unor metode) Implementarea algoritmilor specifici de prelucrare a structurilor de date Învățarea prin descoperire dirijată a tehnicilor moderne de editare oferite de un mediu nou, la prima lui utilizare	

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
5.9. Identificarea avantajelor și a dezavantajelor aplicării tehnicii recursive în implementarea unor rezolvări	Trasarea apelurilor pentru definiții recursive neeficiente (Fibonacci, calculul combinărilor cu triunghiul lui Pascal etc) Estimarea numărului de operații pentru un apel pentru o recursie neeficientă Estimarea numărului de operații pentru prelucrarea iterativă corespunzătoare unei definiții recursive neeficiente	Alocarea dinamică a memoriei (operații și mecanisme specifice) Elemente de programare orientată pe obiecte	Prezentarea unei aplicații în fața clasei (pe echipe) Exersarea instrumentelor de urmărire și depanare pe un program simplu ce conține erori	
5.10. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora	Utilizarea diversificată a tipurilor de date și înstrucțiunilor, a subprogramelor predefinite, în scopul rezolvării unei probleme Utilizarea tehnicilor de structurare a datelor, de acces la date, de alocare și dealocare specifice limbajului de programare Utilizarea tehnicilor de programare modulară, definind și apelând subprograme conform cerinței de rezolvare Abstractizarea implementării unei rezolvări utilizând tehnici de structurare a datelor și prelucrărilor (POO)	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) Structura unei aplicații Modularizarea aplicației Etape în dezvoltarea aplicațiilor Criterii de eficiență	Rularea mai multor implementări pentru o aceeași problemă cu completarea unor fișe de observații	
5.11. Elaborarea și implementarea rezolvărilor de probleme din aria curriculară a specializării	Respectarea etapelor de rezolvare (modelare, implementare, testare, depanare) Justificarea alegerii unei metode de rezolvare pentru o problemă Estimarea eficienței metodei de rezolvare alese Enumerarea tehnicilor folosite în implementarea rezolvării	Implementarea unor algoritmi elementari cu aplicabilitate practică Declararea și utilizarea structurilor de date.	Elaborarea și realizarea unei aplicații simple (modelate anterior), folosind un mediu de programare specific	

Competențe specifice	Competențe de evaluat	Conținuturi	Activități de învățare	Forme de evaluare
5.12. Utilizarea tehnicilor moderne în implementarea aplicațiilor	Exploatarea tehnicilor de implementare folosind programarea vizuală Gestionarea evenimentelor și a prelucrărilor asociate acestora Enumerarea principalelor bibliotecile de obiecte predefinite existente în mediul de dezvoltare utilizat Utilizarea componentelor uzuale din principalele biblioteci Definirea unei sau unor clase proprii, specifice aplicației	Aplicații practice folosind tablouri, șiruri de caractere, înregistrări, liste, grafuri, arbori. Subprograme: Aplicații folosind subprograme Recursivitate	Implementarea unor algoritmi fundamentali (unor metode) Implementarea algoritmilor specifici de prelucrare a structurilor de date	
5.13. Utilizarea instrumentelor de dezvoltare a unei aplicații	Identificarea tehnicilor moderne de editare, depanare, documentare etc. Utilizarea instrumentelor oferite de mediul de dezvoltare pentru design-ul interfeței Utilizarea principalelor instrumente de editare (Intellisense, comentarii, ascundere, factorizare etc.)	Alocarea dinamică a memoriei (operații și mecanisme specifice) Elemente de programare orientată pe obiecte Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) Structura unei aplicații Modularizarea aplicației Etapă în dezvoltarea aplicațiilor Criterii de eficiență a aplicațiilor	Învățarea prin descoperire dirijată a tehnicilor moderne de editare oferite de un mediu nou, la prima lui utilizare Prezentarea unei aplicații în fața clasei (pe echipe)	

INSTRUMENTE DE EVALUARE A COMPETENȚELOR LA DISCIPLINA INFORMATICA

Tipuri de itemi

Pentru o înțelegere mai bună a acestei problematice vom opera cu următoarea definiție de lucru a **itemului**: **Item = <întrebare> + <formatul acesteia> + <răspunsul așteptat>**

Teoria și practica evaluării evidențiază mai multe criterii pe baza cărora pot fi clasificați itemii. Unul dintre criteriile cel mai des utilizate este acela al **gradului de obiectivitate oferit în corectare**. În funcție de acest criteriu, itemii pot fi clasificați în trei mari categorii:

- **Itemii obiectivi** asigură un grad de obiectivitate ridicat în măsurarea rezultatelor școlare și testează un număr mare de elemente de conținut într-un interval de timp relativ scurt. Răspunsul așteptat este bine determinat, ca și modalitatea de notare a acestuia.
- **Itemii semiobiectivi** permit ca răspunsul așteptat să nu fie totdeauna unic determinat, modalitatea de corectare și notare inducând uneori mici diferențe de la un corector la altul. Aceștia testează o gamă mai variată de capacități intelectuale, oferind în același timp posibilitatea de a utiliza și materiale auxiliare în rezolvarea sarcinilor de lucru propuse.
- **Itemii subiectivi (cu răspuns deschis)** solicită un răspuns amplu, permițând valorificarea capacităților creative ale elevilor. Aceștia sunt relativ ușor de construit, principala problemă constituind-o modul de elaborare a schemei de notare astfel încât să se poată obține unitate și uniformitate la nivelul corectării.

1. ITEMI OBIECTIVI

Itemii obiectivi reprezintă instrumente de evaluare frecvent aplicate la disciplina Informatică, întrucât activitățile de proiectare și programare implică formulări standardizate, lipsite de echivoc.

Itemii obiectivi oferă posibilitatea asocierii cu un sistem de notare extrem de simplu: punctajul aferent se acordă integral, se acordă parțial conform unei reguli (formule) de calcul sau nu se acordă deloc (în funcție de răspunsul așteptat).

Din această categorie fac parte:

1.1. ITEMI CU ALEGERE DUALĂ

Alegerea duală presupune formularea unei cerințe cu două variante complementare de răspuns (*Adevărat/Fals, Da/Nu, Corect/Incorect* etc.).

Se pot verifica prin intermediul itemilor cu alegere duală:

- *cunoștințele legate de corectitudinea sintactică a unor expresii (comenzi, instrucțiuni, notații etc.);*

- ➔ înțelegerea semnificației unor noțiuni din terminologia de specialitate (denumiri, instrumente de prelucrare, metode de rezolvare, proprietăți etc.)
- ➔ recunoașterea unor explicații, definiții sau imagini.

Pentru proiectarea corectă a itemilor cu alegere duală este necesară respectarea următoarelor cerințe:

- ➔ formularea clară a enunțului, fără ambiguități sau formulări incomplete;

Întrebarea: „Este corectă structura **if** ($x < 3 \ \&\& \ x > 9$) **THEN** HALT?” generează o altă întrebare: „Din ce punct de vedere?”. Din punct de vedere sintactic este corectă, în timp ce semantic (expresia logică având întotdeauna valoarea FALS) aceasta nu este corectă. Este incorectă și din punctul de vedere al principiului programării structurate. Ambiguitatea poate fi eliminată prin reformularea întrebării în felul următor: „Este corectă din punct de vedere sintactic structura **if** ($x < 0 \ \&\& \ x > 1$) **exit**?”

- ➔ dacă se solicită aprecierea cu ADEVĂRAT/FALS, se vor evita enunțurile foarte generale;
- ➔ selectarea unor enunțuri relevante pentru domeniul de cunoaștere sau categoria de competențe testată (uneori, efortul de a realiza enunțuri fără echivoc duce la elaborarea de itemi ne semnificativi din punct de vedere educațional sau științific);

De exemplu, propoziția: „Orice secvență de algoritm conține una sau mai multe instrucțiuni. (ADEVĂRAT/FALS)”, este ne semnificativă din punct de vedere științific.

- ➔ se va evita utilizarea unor enunțuri negative, acestea conducând la raționamente ce folosesc dubla negație, inducând un grad înalt de ambiguitate;
- ➔ se vor evita enunțurile lungi și complexe, prin eliminarea elementelor redundante, inutile în raport cu ideea enunțului și cerința itemului; nu se va folosi un limbaj academic, o terminologie foarte specializată sau o construcție lingvistică stufoasă și greoaie;
- ➔ se va evita introducerea a două idei într-un singur enunț, cu excepția cazului în care se dorește evidențierea relației dintre acestea;

De exemplu, enunțul „Parametrii de ieșire ai unui subprogram trebuie să fie transmiși prin adresă, dar cei de intrare trebuie să fie transmiși prin valoare. (Adevărat/Fals)”. Conține o primă afirmație adevărată, a doua fiind falsă. Cum relația logică dintre cele două propoziții nu este formulată explicit (conjuncție—și, disjuncție—sau, disjuncție exclusivă—ori..., ori..., implicație—dacă...atunci... sau echivalență—dacă și numai dacă...atunci...), răspunsul Adevărat sau Fals va fi ales mai mult pe considerente de inspirație decât științifice.

- ➔ enunțurile vor fi aproximativ egale ca lungime;
- ➔ enunțurile adevărate sau false să fie aproximativ egale ca număr, dar nu exact egale, deoarece acesta ar putea constitui un indiciu după care elevul încearcă să ghicească răspunsul corect.

EXEMPLU

Clasa: a IX-a

Competențe: Determinarea corectitudinii unui algoritm în raport cu o cerință dată.

Enunț: Se consideră următorii algoritmi de verificare a proprietății unui număr de a fi prim. S-a folosit notația $a \% b$ pentru restul împărțirii lui a la b . Bifați căsuțele corespunzătoare algoritmilor care rezolvă corect problema pentru orice valoare naturală n citită, $n > 1$. Explicați erorile detectate în programele pseudocod ale căror numere nu le-ați bifat.

1	2	3	4
---	---	---	---

1)
citește n
 $\text{prim} \leftarrow 1$
pentru $i=1, n-1$ **execută**
 dacă $n \% i = 0$ **atunci**
 $\text{prim} \leftarrow 0$
 ■
dacă $\text{prim}=1$ **atunci**
 scrie „NUMAR PRIM”
altfel
 scrie „NUMARUL NU ESTE PRIM”
■

3)
citește n
 $i \leftarrow 2$
 $\text{prim} \leftarrow 1$
repetă
 dacă $n \% i = 0$ **atunci**
 $\text{prim} \leftarrow 0$
 ■
 $i \leftarrow i+1$
■ **până când** $i > [\sqrt{n}]$ **sau** $\text{prim}=0$
dacă $\text{prim}=1$ **atunci**
 scrie „NUMAR PRIM”
altfel
 scrie „NUMARUL NU ESTE PRIM”
■

Barem de evaluare și notare:

Răspuns: se bifează 2 și 4.

Notarea se realizează pe principiul testării cu alegere duală, fiecare alegere corectă fiind punctată cu 0.50: dacă 1 nu este bifat se acordă 0.50; dacă 2 este bifat se acordă 0.50; dacă 3 nu este bifat se acordă 0.50; dacă 4 este bifat se acordă 0.50. În total 2 puncte.

Total (2 puncte)

2)
citește n
 $\text{prim} \leftarrow 1$
pentru $i=2, [\sqrt{n}]$ **execută**
 dacă $n \% i = 0$ **atunci**
 $\text{prim} \leftarrow 0$
 ■
dacă $\text{prim}=1$ **atunci**
 scrie „NUMAR PRIM”
altfel
 scrie „NUMARUL NU ESTE PRIM”
■

4)
citește n
 $i \leftarrow 2$
cât timp $n \% i \neq 0$ **execută**
 $i \leftarrow i+1$
■
dacă $i = n$ **atunci**
 scrie „NUMAR PRIM”
altfel
 scrie „NUMAR NEPRIM”
■

1.2. ITEMI DE TIP PERECHE

Itemii de tip pereche solicită stabilirea unor corespondențe între informațiile distribuite pe două coloane. Prima coloană conține informații de tip enunț (*premise*), iar cea de-a doua coloană conține informații de tip răspuns. Elevului i se solicită să asocieze fiecare enunț cu un unic răspuns.

Cele două coloane sunt precedate de instrucțiuni de asociere în care i se explică elevului tehnica de formare a perechilor (să unească printr-o linie, să rescrie perechile asociate sau doar elementele lor de identificare etc.) și se precizează dacă un răspuns poate fi folosit la mai mult de un enunț (dacă funcția de asociere este injectivă sau nu), eventual dacă există răspunsuri care nu vor fi folosite niciodată (dacă funcția de asociere este surjectivă sau nu).

Se verifică prin intermediul itemilor de tip pereche capacitatea elevului de a stabili corelații între:

- funcții și instrumente;
- simboluri și concepte;
- termeni și definiții;
- probleme și metode de rezolvare.

Itemii de acest tip permit abordarea unui volum mare de informație într-un interval de timp relativ redus. Factorul de discriminare este ceva mai mare decât în cazul itemilor cu alegere duală, strategia de asociere „la întâmplare” neconducând decât în situații foarte rare la un rezultat acceptabil privind rezultatul testului.

Pentru proiectarea corectă a itemilor de tip de pereche este necesară respectarea următoarelor cerințe:

- *utilizarea unui material omogen, dintr-o sferă relativ restrânsă;*
- *utilizarea unui număr inegal de premise și răspunsuri, astfel încât, dacă elevul asociază corect $n-1$ enunțuri dintre cele n date, să nu rezulte automat răspunsul pentru cel de-al n -lea enunț;*
- *aranjarea listei de răspunsuri (mai ales dacă sunt multe) într-o ordine logică, astfel încât căutarea răspunsului în listă să se realizeze cât mai comod;*
- *aranjarea enunțurilor în listă astfel încât să nu se poată intui o regulă de asociere (referințele să fie „încrucișate”);*
- *aranjarea coloanelor astfel încât acestea să încapă în întregime pe aceeași pagină.*

EXEMPLU

Clasa: a IX-a

Competențe: Recunoașterea tipurilor de constante în limbajul C/C++

Enunț: Alegeți pentru fiecare constantă scrisă în coloana stângă, tipul corespunzător din dreapta. Atragem atenția că este posibil ca unele constante să aibă același tip, iar unele dintre tipurile propuse pot să nu fie folosite niciodată.

1	întreg
3.6	real
.12	
'9'	caracter
"123.1"	șir de caractere
4E-2	
'\n'	logic
'A'	
"B"	
-23	

Barem de evaluare și notare: Se acordă câte 0.50 puncte pentru fiecare asociere corectă.

Constante

- întregi: 1; -23
- reale: .12 ; 3.6 ; 4E-2
- caracter: '9'; '\n'; 'A'
- șir de caractere: "123.1" ; "B"

Total (5 puncte)

1.3. ITEMI CU ALEGERE MULTIPLĂ

Un item cu alegere multiplă este format dintr-un enunț numit **premisă** sau **bază** și un număr de opțiuni din care elevul trebuie să aleagă un singur răspuns numit **cheie**. Celelalte răspunsuri, neconforme cu cerința, dar plauzibile poartă numele de **distractori**.

Se verifică prin intermediul itemilor de tip pereche capacitatea elevului de a identifica:

- definiții și notații;
- secvențe de program care realizează o anumită prelucrare;
- expresii cu o valoare dată;
- termeni și expresii de specialitate;
- metode de rezolvare și tehnici de implementare.

Itemii de acest tip permit abordarea unui volum mare de informație într-un interval de timp relativ redus.

O categorie de itemi cu alegere multiplă solicită răspunsul corect, celelalte variante fiind greșite, în timp ce alți itemi solicită cel mai bun răspuns, pe baza unei discriminări complexe. În aceste cazuri trebuie manifestată atenție la formularea cerinței astfel încât criteriul de discriminare a „celui mai bun răspuns” să reiasă clar din enunț.

Pentru proiectarea corectă a itemilor cu alegere multiplă este necesară respectarea următoarelor cerințe:

- ➔ stabilirea clară a cerinței, în concordanță cu obiectivul de evaluare;
- ➔ furnizarea tuturor informațiilor necesare în premisă, eliminându-se materialul irelevant;
- ➔ formularea premisei folosind afirmații sau întrebări pozitive;
- ➔ construirea unor alternative plauzibile, aflate în concordanță cu premisa;
- ➔ construirea itemului astfel încât să existe o singură alternativă „corectă” sau „cea mai bună”;
- ➔ construirea unor alternative astfel încât distractorii să fie în mod cert „greșiți” sau „mai puțin buni”, iar varianta cheie să fie în mod cert „corectă” sau „cea mai bună”;
- ➔ aranjarea listei de răspunsuri într-o ordine logică, astfel încât căutarea răspunsului în listă să se realizeze cât mai comod;
- ➔ construirea ansamblurilor de itemi cu alegere multiplă astfel încât răspunsurile să ocupe poziții diferite în lista de variante (să nu fie în mod constant al doilea răspuns, de exemplu)

EXEMPLU

Clasa: a IX-a

Obiective: Alegerea din mai multe posibilități de prelucrare a celei care corespunde cerinței

Enunț: Care dintre următoarele expresii **C/C++** are valoarea 1 dacă și numai dacă numărul natural memorat de variabila **x** de tip **int** are exact două cifre?

- a. $x/100==0$ b. $x/100==0 \ \&\& \ x\%10==0$
 c. $x/10!=0$ d. $x/100==0 \ \&\& \ x/10!=0$

Barem de evaluare și notare: Se acordă 2 puncte pentru alegerea variantei d.

2. ITEMI SEMIOBIECTIVI

Itemii semiobiectivi formează o categorie de instrumente de evaluare ce solicită construirea parțială sau totală a unui răspuns pe baza unei sarcini definite.

Itemii semiobiectivi sunt caracterizați prin:

- ➔ posibilitatea de a testa o gamă mai largă de capacități intelectuale și rezultate ale învățării;
- ➔ crearea unor situații cognitive de nivel mai ridicat prin solicitarea de elaborare a răspunsului și nu de alegere a lui dintr-o mulțime prestabilită, ca în cazul itemilor obiectivi;
- ➔ raportarea parțial subiectivă a profesorului în raport cu răspunsul formulat (răspunsul poate fi scris ordonat sau dezordonat, formularea poate fi mai clară sau mai neclară, termenii folosiți se pot încadra în niște standarde științifice sau pot fi variante particulare ale acestora etc.);
- ➔ posibilitatea asocierii unui sistem de notare în care pot să intervină situații neprevăzute (răspunsuri neașteptate, care comportă raportări noi la barem).

Din categoria itemilor semiobiectivi fac parte: **itemii cu răspuns scurt, itemii de completare și itemii structurați.**

1.1. ITEMI CU RĂSPUNS SCURT

Solicită ca elevul să formuleze un răspuns scurt sau să completeze o afirmație astfel încât aceasta să capete sens sau să aibă valoare de adevăr.

Se pot verifica prin intermediul itemilor cu răspuns scurt și de completare:

- ➔ cunoașterea unor noțiuni, expresii de specialitate, simboluri, notații etc.;
- ➔ recunoașterea și nominalizarea unor elemente vizuale specifice unui anumit mediu de lucru;
- ➔ capacitatea de integrare a unor elemente necesare din punct de vedere sintactic sau semantic într-un context dat;
- ➔ schimbarea unor elemente dintr-un context dat astfel încât să se realizeze o finalitate precizată.

Itemii cu răspuns scurt se prezintă cel mai des sub forma unor întrebări. Ei solicită un răspuns sub o formă restrânsă (un număr, un simbol, un cuvânt, o expresie, o propoziție sau frază concisă).

EXEMPLU

Clasa: a IX-a

Competențe: Aplicarea algoritmilor fundamentali pentru date simple.

Enunț: Precizați domeniul de apartenență pentru variabila reală **x** dacă expresia **C/C++** următoare are valoarea 1.

$(x>3 \ \&\& \ x<=100) \ || \ !(x<34 \ || \ x>=12)$

Barem de evaluare și notare: Se acordă un punct pentru răspunsul **x (3,100]** sau pentru orice răspuns echivalent.

1.2. ITEMII DE COMPLETARE

Se prezintă sub forma unui enunț, unei afirmații incomplete. Ei solicită găsirea cuvântului sau sintagmei care completează și dă sens enunțului respectiv.

Pentru proiectarea corectă a itemilor cu răspuns scurt / de completare este necesară respectarea următoarelor cerințe:

- ➔ formularea enunțului astfel încât să admită un răspuns scurt, exprimat cu precizie;
- ➔ formularea enunțului astfel încât acesta să admită un singur răspuns corect, pe cât posibil;
- ➔ rezervarea unor spații pentru răspuns care să sugereze numărul de cuvinte așteptate (dacă acest lucru nu reprezintă un indiciu), nu și dimensiunea lor;
- ➔ vizarea unui răspuns care să reprezinte o sinteză de cunoștințe sau un rezultat al înțelegerii unei situații și mai puțin o reproducere a unor informații.

EXEMPLU

Clasa: a IX-a

Competențe: Analiza valorilor variabilelor prelucrate de un algoritm pseudocod.

Enunț: Completați enunțul următor cu valoarea numerică potrivită (i număr real):

Structura repetitivă

repetă

i i/2

până când i=4

va asigura executarea instrucțiunii de atribuire de exact 4 ori dacă valoarea inițială a lui i este

Barem de corectare și notare: Se acordă un punct pentru răspunsul 64.

EXEMPLU

Clasa: a XI-a

Competențe: Completarea apelului recursiv al unui subprogram cu o cerință dată

Enunț: Se consideră subprogramul recursiv, **S**, definit incomplet. Scrieți expresia care poate înlocui puntele de suspensie, astfel încât pentru x=2 să se afișeze 3 caractere.

```
void S(int x) { cout<<'*';  
  if (...) { cout<<'*';  
    S(x-1);  
  }  
}
```

Barem de evaluare și notare: Se acordă două puncte pentru răspunsul x>1 sau pentru orice răspuns echivalent cu acesta.

1.3. ÎNTREBĂRI STRUCTURATE

Solicita, printr-un sistem de subîntrebări relative la o temă comună, răspunsuri de tip obiectiv, răspunsuri scurte sau de completare prin care se pot evalua cunoștințele complexe referitoare la tema respectivă fără a solicita elaborarea unui răspuns deschis (eseu).

Se pot verifica prin intermediul întrebărilor structurate:

- capacitatea de a urmări, recunoaște, adapta și construi un algoritm pe o temă dată sau un program într-un limbaj de programare;
- capacitatea de a realiza din aproape în aproape o prelucrare complexă utilizând un mediu de lucru informatic.

O întrebare structurată poate să conțină materiale suport și informații suplimentare ce se adaugă treptat, conferind procesului de evaluare varietate, complexitate și graduație. Se pot verifica totodată cunoștințele, dar și priceperi și deprinderi sporind gradul de obiectivitate în raport cu itemii cu răspuns deschis.

Proiectarea itemilor structurați se face gradat în ceea ce privește nivelul de dificultate din cel puțin două motive: pentru a asigura evaluarea unor capacități cu nivele crescânde de complexitate, dar și pentru a încuraja abordarea subiectului de către elev.

Subîntrebările ce formează itemul permit creșterea progresivă a dificultății cerințelor, dar este recomandat ca subîntrebările să fie independente, adică răspunsul la o întrebare să nu depindă de răspunsul la întrebările precedente. Proiectarea lor necesită atenție, pricepere și timp.

Pentru proiectarea corectă a întrebărilor structurate este necesară respectarea următoarelor cerințe:

- redactarea subîntrebărilor astfel încât acestea să solicite răspunsuri simple la început, crescând pe parcurs dificultatea acestora;
- formularea unor subîntrebări autoconținute (al căror răspuns corect să nu depindă de răspunsul corect la una dintre întrebările precedente);
- realizarea concordanței dintre enunțul general (tema întrebării) și subîntrebările formulate.

EXEMPLU

Clasa: a X-a

Obiective: Aplicarea algoritmilor fundamentali pentru date simple

Enunț: Următorii patru itemi se referă la secvența următoare, în care variabilele i și n sunt de tip întreg.

Pascal

```
readln(n);  
for i:=1 to n do  
  if (n mod i<>0) and (i mod 2<>0) then  
    if(i mod 3>0) then write('*')  
    else write('#');
```

C/C++

```
cin>>n;  
for(i=1;i<=n;i++)  
  if(n%i!=0 && i%2!=0)  
    if(i%3>0) cout<<'*';  
    else cout<<'#';
```

1. Stabiliți dacă instrucțiunea **readln(n)/cin>>n** ; este o instrucțiune prin care se afișează pe ecran valoarea variabilei n (Da/Nu).
2. Stabiliți dacă instrucțiunea **if(i mod 3>0)...** / **if(i%3>0)...** este o instrucțiune de decizie (Da/Nu).
3. De câte ori se execută instrucțiunea **write('#');** / **cout<<'#';** la o rulare a programului?
 - a) o dată
 - b) niciodată
 - c) de două sau mai multe ori
 - d) depinde de valoarea lui n
 - e) de o infinitate de ori
4. Ce se va afișa în urma executării secvenței dacă pentru **n** se citește valoarea **16**?
5. Scrieți cea mai mare valoare pe care o poate primi variabila **n** pentru ca numărul caracterelor afișate să fie **0**.
6. Scrieți cea mai mică valoare naturală pe care o poate primi variabila **n** pentru care șirul de caractere afișat începe cu două caractere *****.
7. Modificați expresia **i mod 2<>0** (Pascal) / **i%2!=0** (C/C++) astfel încât șirul afișat să nu conțină nici un caracter *****.

Barem de evaluare și notare: Se acordă

- 1) **1 punct** pentru răspunsul NU.
- 2) **1 punct** pentru răspunsul DA.
- 3) **1 punct** pentru alegerea literei d).
- 4) **2 puncte** pentru *****##**

Se acordă câte 1p pentru numărul corect de caractere și respectiv pentru ordinea corectă a acestora.

- 5) 1 punct pentru răspunsul 3.
 6) 1 punct pentru răspunsul 9.
 7) 2 puncte pentru un răspuns echivalent cu:
Pascal $(n \bmod i < 0) \text{ and } (i \bmod 3 = 0)$
C/C++ $n \% i != 0 \ \&\& \ i \% 3 == 0$

3. ITEMI SUBIECTIVI (CU RĂSPUNS DESCHIS)

Itemii subiectivi formează o categorie de instrumente de evaluare ce vizează creativitatea elevului, originalitatea și caracterul personal al răspunsului. Aceștia sunt relativ ușor de construit, principala problemă constituind-o modul de elaborare a schemei de notare astfel încât să se poată obține unitate și uniformitate la nivelul corectării. Itemii subiectivi sunt caracterizați prin:

- abordare globală a unei sarcini asociate unui obiectiv ce nu poate fi evaluat prin intermediul itemilor obiectivi;
- crearea unor situații cognitive de nivel foarte ridicat prin solicitarea de a realiza interacțiuni reale și complexe între cunoștințe, abilități și deprinderi;
- raportarea subiectivă a profesorului în raport cu răspunsul formulat;
- necesitatea predefinirii unor criterii privind baremul de corectare și notare, criterii clare, judicioase și puternic anticipative;
- posibilitatea, în cazul în care baremul nu a prevăzut toate situațiile de interpretare și construire a răspunsului, a unor elemente noi (răspunsuri neașteptate) care comportă reanalizarea baremului.

În cazul informaticii se pot elabora **itemi subiectivi de tip eseu** (structurat sau liber) și **itemi de tip problemă** (care necesită proiectare, redactare și uneori implementare a rezolvării).

3.1. ITEMI DE TIP ESEU

Itemii de tip eseu pot fi structurați sau liberi. Itemii structurați sunt construiți astfel încât răspunsul așteptat să fie „orientat” cu ajutorul unor elemente din enunț (indicii privind ordinea de tratare, numărul de linii, formularea răspunsului, ideile care trebuie să fie atinse etc.). Un eseu liber nu furnizează în enunț niciun fel de indicații sau constrângeri, elevul având libertatea să-și strutureze cum consideră și cum poate materialul pe care-l solicită enunțul. Acest tip de eseu comportă operații de maximă complexitate (analiză, sinteză, sistematizare și restructurare) lăsând frâu liber fanteziei și capacităților creative ale elevului.

Deoarece la informatică elementele de creativitate se manifestă mai ales prin rezolvări de probleme și proiecte, itemii de tip eseu preferați sunt cei structurați, un eseu liber nefiind necesar decât rar, pentru anumite teme cu un volum mai mare de elemente „informative” în raport cu achizițiile „operaționale”.

Itemii de tip eseu se prezintă sub forma unor cerințe generale însoțite eventual (pentru eseurile structurate) de indicii privind tratarea cerinței. Se pot adăuga restricții privind întinderea în timp sau spațiu (număr rânduri, pagini, paragrafe etc.) sau privind forma de prezentare a răspunsului (descriere, relatare, schemă etc.).

Se pot verifica prin intermediul itemilor de tip eseu:

- cunoștințele legate de algoritmi elementari, de structurile de date, de etapele conceptuale ale proiectării unui „produs” etc.
- capacitățile de sistematizare a unor elemente prin construirea unor scheme sau reprezentări grafice.

EXEMPLU

Clasa: a X-a

Competențe: Cunoașterea sintetică a tipurilor de date în limbajul de programare studiat (Pascal sau C/C++).

Enunț: Tipuri de date. Scrieți definiția tipului de date și cuprindeți tipurile de date învățate în diferite scheme de clasificare. Realizați o scurtă descriere (de cel mult două rânduri) a fiecărui tip în parte.

Timp de lucru: 45 minute

Barem de evaluare și notare:

Criterii și rezolvare	Punctaj	Observații
➤ definirea tipului de date (mulțime de valori și de operatori)	1 punct	
➤ nominalizarea principalelor tipuri de date (caracter, întreg, real, logic, tablou, înregistrare, fișier, adresă); ➤ nominalizarea subtipurilor (întregi, reale, string, tablouri uni-, bi-, multidimensionale)	2 puncte	0.25 pentru fiecare tip și 0.25 pentru „alte,” sau „subtipuri,”
➤ clasificări posibile (simple – structurate, predefinite – definite de utilizator, interne – externe, omogene – eterogene)	1 punct	0.25 puncte fiecare criteriu
➤ încadrarea tipurilor de date în diferitele scheme de clasificare	1 punct	0.50 corectitudine, 0.50 completitudine
➤ descrierea fiecărui tip de date (minimum 5 principale, în total minimum 8)	2 puncte	0.25 pentru fiecare tip.
➤ capacitatea de sinteză	1 punct	
➤ coerența prezentării	1 punct	Erorile științifice nepenalizate la alte secțiuni ale baremului scad punctajul alocat acestei secțiuni

3.2. REZOLVARE DE PROBLEME

Rezolvarea de probleme este o activitate specifică și des utilizată la disciplina Informatică, elementele gândirii algoritmice, metodele de rezolvare și tehnicile de implementare fiind supuse unui „tir” sistematic de probleme prin care acestea să formeze competențe reale de programare.

Obiectivele urmărite prin utilizarea rezolvării de probleme sunt:

- înțelegerea problemei;
- obținerea informațiilor necesare rezolvării problemei;

- ➔ formularea și testarea ipotezelor;
- ➔ descrierea metodei de rezolvare a problemei;
- ➔ elaborarea unui scurt raport despre rezultatele obținute;
- ➔ posibilitatea de generalizare și transfer a tehnicilor de rezolvare.

Cerințe suplimentare asociate unei probleme pot pune în evidență capacitatea elevului de a estima eficiența unei rezolvări, de a construi un algoritm conform unor criterii (limita de memorie, număr de instrucțiuni etc.).

Se pot formula probleme în care se furnizează algoritmul și se cere un enunț de problemă care se rezolvă prin intermediul algoritmului respectiv. Acest tip de item impune o analiză atentă a algoritmului și asocierea lui cu una dintre problemele sau prelucrările numerice întâlnite la matematică, fizică sau în alte domenii, o formulare a enunțului care să se caracterizeze prin coerență.

Enunțurile pot fi formulate abstract, „la obiect” sau pot crea un „context” care trebuie modelat pentru a se ajunge la rezolvarea propriu-zisă. „Povestea” în spatele căreia se ascunde problema are de cele mai multe ori conotații practice, descriind situații concrete de prelucrare, amintind că rolul programatorului este acela de a „ordona” inițial informația și operațiile specifice unui anumit context și abia după aceea de a elabora algoritmul, de a implementa și verifica programul corespunzător.

Evaluarea prin rezolvare de probleme la informatică ridică uneori probleme din punctul de vedere al întocmirii baremului de corectare. Unele tendințe exagerate tind să impună o corectare pe principiul: *problemă=program funcțional corect* (pornind de la premisa că „un program care aproape merge e ca un avion care aproape zboară”). Se recomandă totuși ca baremul de corectare să cuprindă fracțiuni din punctaj pentru diferitele aspecte pe care le comportă rezolvarea unei probleme la informatică: corectitudinea sintactică, structurarea datelor și declararea variabilelor, structurarea programului, corectitudinea algoritmului, eficiența algoritmului, tratarea unor situații limită, eventual explicarea metodei aplicate (chiar dacă a fost aplicată greșit) etc.

Se pot verifica prin intermediul itemilor de rezolvare de probleme:

- ➔ concepția unor algoritmi de rezolvare a problemelor elementare;
- ➔ capacitatea de a înțelege un algoritm general prin adaptarea acestuia astfel încât să rezolve o problemă particulară;
- ➔ capacitatea de a alege structurile de program și de date adecvate rezolvării unei probleme;
- ➔ abilitatea de a implementa programul, de a-l depana, de a-l testa și, în funcție de erorile apărute, de a reconsidera elementele de sintaxă ale programului, strategiile de structurare a datelor sau însuși algoritmul de rezolvare (în partea practică a probei);
- ➔ capacitatea de a organiza volume mari de date cu ajutorul bazelor de date;
- ➔ discernământul în a alege un algoritm mai eficient (conform unuia dintre criteriile studiate: număr operații, spațiu de memorie utilizat)

EXEMPLU

Clasa: a XI-a

Objective: Construirea algoritmilor care să efectueze operații tipice asupra structurilor de date

Enunț: Fișierul **expresie.in** conține un șir de caractere format din cifre {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} și paranteze drepte deschise și închise. Șirul reprezintă o expresie aritmetică în care operațiile sunt codificate astfel:

- un număr aflat între paranteze drepte reprezintă operația de ridicare la pătrat;
- o secvență de cel puțin două paranteze aflate pe poziții consecutive reprezintă operații de însumare a variabilelor.

Scrieți un program **Pascal/C/C++** care citește șirul de caractere din fișierul **expresie.in**, determină valoarea numerică obținută în urma evaluării expresiei menționate mai sus și scrie această valoare pe prima linie a fișierului **expresie.out**.

Date de intrare

Fișierul de intrare **expresie.in** conține o singură linie pe care este scrisă expresia.

Date de ieșire

Fișierul de ieșire **expresie.out** va conține pe prima linie valoarea numerică obținută în urma evaluării expresiei.

Restricții și precizări

- expresia citită este formată din cel mult 100 de caractere;
- valorile numerice care se vor obține au cel mult 9 cifre;
- în expresia citită fiecare cifră este inclusă între paranteze drepte;
- expresia este corectă.

Exemple:

expresie.in	expresie.out	Explicație
[[2][4]][[3][2]]	569	$(2^2+4^4)^2+(3^3+2^2)^2=400+169=569$
[[[2]]]	256	$((2^2)^2)^2$
[[2][4]][3]	409	$(2^2+4^4)^2+3^2=400+9=409$

Timp de lucru: 30 minute

Barem de evaluare și notare:

Pentru program corect se acordă punctaj maxim	10 p.	* datele de intrare se consideră corecte, nefiind necesară validarea lor
➔ declararea corectă a variabilelor	1 p.	
➔ identificarea unui termen al sumei	1 p.	
➔ determinarea puterii unui termen	2 p.	
➔ determinarea valorii expresiei	3 p.	
➔ lucrul corect cu fișiere*	2 p.	
➔ corectitudinea globală a programului	1 p.	

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte greșeli neprecizate în barem

Exemplu de rezolvare C/C++

```
#include <fstream.h>
ifstream f(„expresie.in”);
ofstream g(„expresie.out”);
```

```

char s[100];
long St[100];
int suma, S, x, vf;
void calcul (int i)
{ if(i<=strlen(s)-1)
  { if(s[i]=='(') St[++vf]=-1;
    else
      if(s[i]>='0' && s[i]<='9') St[++vf]=s[i]-'0';
      else
        { suma=0;
          while(St[vf]>-1) {suma+=St[vf];vf--;}
          St[vf]=suma*suma;
        }
      calcul(i+1);
    }
}

int main ()
{f.getline(s,101);
 calcul(0);
 while(vf>=2){St[vf-1]+=St[vf];vf--;}
 g<<St[1]<<'\n';
 f.close();g.close();
 return 0;
}

```

4. METODE COMPLEMENTARE DE EVALUARE

Metodele complementare de evaluare reprezintă instrumente suplimentare, ne-standardizate, de evaluare dispunând de forme specifice cum ar fi: **investigația, referatul, portofoliul, proiectul, observarea sistematică a activității elevului și autoevaluarea.**

Metodele complementare de evaluare permit o evaluare individualizată (observare sistematică), au capacitatea de a educa spiritul de echipă prin activități de grup (investigații, proiecte) și au un caracter profund integrator realizat prin interdisciplinaritate, educare și instruire multilaterală.

4.1. INVESTIGAȚIA

Investigația este o metodă de evaluare și învățare utilizată ocazional la disciplina Informatică.

Organizarea unei activități de evaluare și învățare prin metoda investigației presupune:

- valorificarea metodei de învățare prin descoperire;
- studiul unor documentații complementare, experimentarea unor instrumente de prelucrare nestandard;

- extrapolarea cunoștințelor dobândite și verificarea ipotezelor formulate;
- solicitarea unor cunoștințe sau deprinderi dobândite la alte discipline prin adaptarea creatoare a acestora la cerințele temei de investigație.
- În cele mai multe dintre cazuri investigația trebuie să fie organizată ca muncă independentă depusă de elev, dirijată și sprijinită de profesor, prin:
 - formularea generală a temei;
 - asigurarea surselor bibliografice sau tehnice necesare;
 - formularea unor indicații care să direcționeze activitatea elevilor;
 - urmărirea activității elevului în sensul utilizării eficiente și creatoare a materialului de investigat;
 - sprijinirea elevilor sau grupurilor de elevi care întâmpină dificultăți în înțelegerea temei sau a metodelor specifice de studiu;
 - încurajarea și evidențierea activităților creatoare desfășurate de elevi, a descoperirilor neașteptate.

EXEMPLU

Clasa: a XII-a

Competențe: Descoperirea asemănărilor și deosebirilor existente între mediile de programare corespunzătoare unui același limbaj de programare cunoscut (de exemplu, mediile *Borland Pascal* și *Free Pascal* sau *Borland C++* și *MinGW*)

Enunț: Studiați Help-ul fiecăruia dintre mediile *Borland C++* și *MinGW* și scrieți un program care alege dintr-un meniu una dintre opțiunile următoare rezolvând cerința corespunzătoare:

- a) ordonează crescător un șir de valori
- b) desenează un cerc de rază dată
- c) sfârșit program.

Realizați un raport în care să evidențiați particularitățile fiecăruia dintre mediile studiate, asemănările și deosebirile pe care le considerați importante.

Timp de lucru: 100 minute

Organizarea activității: grupuri de 4-6 elevi cu două calculatoare și o documentație minimală *Borland C++* sau *MinGW*.

Barem de evaluare și notare: Se acordă:

- 4 puncte pentru programul realizat în cele două medii recomandate;
- 2 puncte pentru utilizarea creatoare a informațiilor din Help;
- 2 puncte pentru organizarea informației în raport, pentru coerența și sistematizarea informațiilor;
- 1 punct pentru terminologia științifică utilizată corect;
- 1 punct din oficiu.

Total :10 puncte

4.2. REFERATUL ȘI PROIECTUL

Referatul reprezintă o formă de îmbinare a studiului individual cu activitate de prezentare și argumentare. Tema referatului, însoțită de bibliografie și alte surse de documentare (Internet, vizite etc.), este tratată în mod independent de către elev și susținută apoi în fața colegilor sau altui auditoriu mai larg. Varietatea universului informatic, a limbajelor

și tehnicilor de programare, justifică utilizarea acestei forme de studiu și evaluare la clasă, la disciplina Informatică. Dacă studiul aferent și rezultatul studiului prezintă interes și din punct de vedere practic, rezultatul fiind, de exemplu, un program (o aplicație) sau dacă bibliografia propusă este mai bogată și etapele de proiectare (concepție), implementare și testare necesită un timp mai îndelungat, lucrarea poartă numele de **proiect**.

Organizarea unei activități de evaluare și învățare prin intermediul referatelor și proiectelor presupune:

- valorificarea metodei de învățare prin descoperire;
- studiul unor materiale suplimentare și izvoare de informare diverse în scopul îmbogățirii și activării cunoștințelor din domeniul studiat sau domenii conexe, prin completări de conținut ale programei sau prin aducerea în atenție a unei problematice complet noi;
- structurarea informației corespunzătoare unui referat într-un material ce poate fi scris, ilustrat sau prezentat pe calculator; activitățile de concepere, organizare, experimentare, reproiectare (dacă este cazul), dezvoltare și elaborare a documentației aferente necesită planificarea unor etape de elaborare și o strategie de lucru, în cazul proiectului;
- prezentarea referatului sau proiectului de către elevul sau elevii care l-au elaborat, acesta (sau un reprezentant al grupului) trebuind să-l susțină, să fie capabil să dea explicații suplimentare, să răspundă la întrebări etc.

Referatul este de regulă o lucrare de mai mică amploare, dar mai structurată și mai bogată în informații decât o temă de muncă independentă aferentă lecției curente. Proiectul este o lucrare mai amplă a cărei temă este comunicată sau aleasă din timp, elaborarea unui proiect putând să dureze de la 1-2 săptămâni până la 2-3 luni sau chiar un semestru. Proiectul poate fi elaborat în grup, cu o distribuie judicioasă a sarcinilor între membrii grupului.

Pentru a realiza o evaluare pe bază de referate, profesorul:

- va formula teme clare, de complexitate medie, precizând pe cât posibil amploarea lucrării (câte pagini, durata maximă necesară prezentării etc.);
- va recomanda sau asigura sursele bibliografice și de informare necesare;
- își va rezerva suficient timp (în perioada de evaluare sau la sfârșitul unor unități de învățare) pentru ca elevii însărcinați cu elaborarea referatelor să-și poată prezenta referatul;
- va supraveghea discuțiile purtate cu elevii asupra conținutului referatului.

Pentru a realiza o evaluare pe bază de proiecte, profesorul:

- va formula teme practice, de complexitate sporită, lăsând celor care elaborează proiectul multă libertate în a improviza, adapta și interpreta cerința într-un mod personal;
- va stabili un termen final și, în funcție de modul de evaluare, termene intermediare de raportare;
- va recomanda sau asigura sursele bibliografice și de informare necesare;
- își va rezerva suficient timp (în perioada de evaluare sau la sfârșitul unor unități de învățare) pentru ca elevii însărcinați cu elaborarea proiectelor să-și poată prezenta rezultatul proiectării;
- va supraveghea discuțiile purtate cu elevii asupra proiectului.

EXEMPLU

Clasa: a X-a

Competențe: Alegerea unui algoritm de rezolvare dintr-o mulțime de algoritmi dați, conform unui criteriu;

Estimarea eficienței, ca timp de executare, a unei rezolvări;

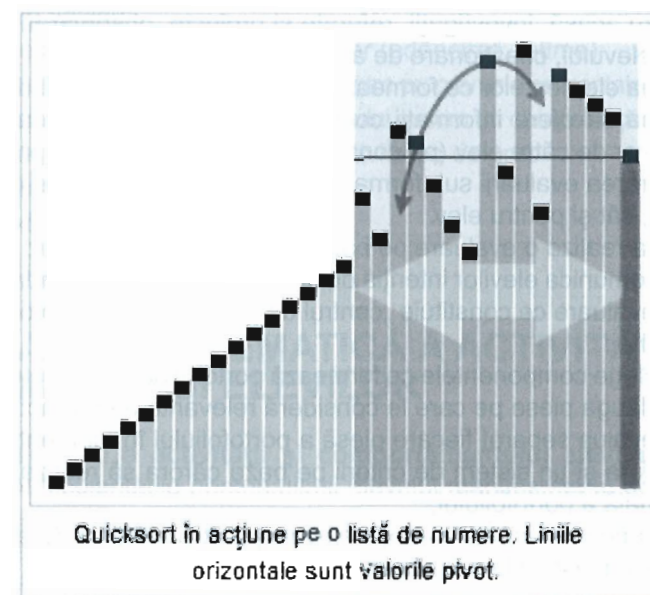
Tema proiectului: Metode de sortare.

Bibliografie: D.E. Knuth – Arta programării calculatoarelor (vol. I)

http://ro.wikipedia.org/wiki/Categorie:Algoritmi_de_sortare

Cerințele proiectului:

- a) Studiul și implementarea a cel puțin 6 metode diferite de sortare (sortarea prin inserție, selecție, BubbleSort, ShakerSort, ShellSort, heapsort, QuickSort, RadixSort);
- b) Analiza obiectivă a fiecărei metode în parte;
- c) Reprezentarea grafică a modificărilor efectuate pe parcursul ordonării. Exemplu:



d) Analiza eficienței fiecărei metode în raport cu date experimentale variate structurate în diferite moduri;

e) Sistematizarea și prezentarea concluziilor.

Termen de realizare: 9 săptămâni

Barem de evaluare și notare:

- | | |
|---|----------|
| a) Implementarea metodelor (câte 0.5 fiecare metodă) | 3 puncte |
| b) Complexitatea și varietatea metodelor alese | 1 punct |
| c) Ilustrarea grafică a comportamentului metodelor de sortare | 2 puncte |
| d) Contribuții originale la tematica proiectului | 1 punct |
| e) Capacitatea de autoanaliză a proiectului | 1 punct |
| f) Încadrarea în timp | 1 punct |
| g) Puntaj acordat din oficiu | 1 punct |

4.3. PORTOFOLIUL

Potofoliul reprezintă o metodă complexă de evaluare în care un rezultat al evaluării este elaborat pe baza aplicării unui ansamblu variat de probe și instrumente de evaluare.

Prin multitudinea de forme și momente în care se desfășoară testarea elevului, rezultatul final „converge” către valoarea reală a acestuia, sesizând elementele de progres sau regres, ilustrând preocuparea pentru lămurirea neclarităților, oferind o imagine de ansamblu asupra nivelului cunoștințelor, gradului de formare a abilităților și gradului de raportare atitudinală pe care acesta o are față de tema evaluată. Portofoliul este realizat pe o perioadă mai îndelungată, de la un semestru, un an, până la un ciclu de învățământ.

Conținutul unui portofoliu este reprezentat de rezultatele la: lucrări scrise sau practice, teme pentru acasă, investigații, referate și proiecte, observarea sistematică la clasă, autoevaluarea elevului, chestionare de atitudini etc.

Alegerea elementelor ce formează portofoliul este realizată de către profesor (astfel încât acestea să ofere informații concludente privind pregătirea, evoluția, atitudinea elevului) sau chiar de către elev (pe considerente de performanță, preferințe etc.)

Structurarea evaluării sub forma de portofoliu se dovedește deosebit de utilă, atât pentru profesor, cât și pentru elev.

Pentru a realiza o evaluare pe bază de potofoliu, profesorul:

- va comunica elevilor intenția de a realiza un portofoliu, adaptând instrumentele de evaluare ce constituie „centrul de greutate” ale portofoliului la specificul disciplinei;
- va alege componentele ce formează portofoliul, dând și elevului posibilitatea de a adăuga piese pe care le consideră relevante pentru activitatea sa;
- va evalua separat fiecare piesă a portofoliului în momentul realizării ei, dar va asigura și un sistem de criterii pe baza cărora să realizeze evaluarea globală și finală a portofoliului;
- va pune în evidență evoluția elevului, particularitățile de exprimare și de raportare a acestuia la aria vizată;
- va integra rezultatul evaluării portofoliului în sistemul general de notare.

EXEMPLU

Tematica: Grafuri (portofoliu realizat pe parcursul semestrului I, clasa a XI-a)

Competențe:

- descrierea în limbaj natural a algoritmilor de verificare a proprietății de graf;
- aplicarea unor metode de verificare pentru grafuri date;
- reprezentarea unor algoritmi de verificare prin calculul gradelor vârfurilor (graf complet, graf regulat), prin parcurgere (graf conex, tare conex);
- descrierea unei metode pentru verificarea proprietății de lanț/ciclu/graf hamiltonian, de ciclu/graf eulerian;
- utilizarea corectă a termenilor specifici;
- descrierea uneia sau mai multor metode pentru drumuri optime;
- aplicarea unor metode de prelucrare pentru grafuri date;
- reprezentarea unor secvențe specifice de prelucrare.

Conținutul portofoliului:

- teste și lucrări scrise;
- teme de activitate independentă scrisă;
- programe implementate independent (parcurea grafurilor, determinarea componentelor conexe) sau în grupe de doi elevi;
- referat prezentat sau participare la un proiect;
- fișe de evaluare a elevului;
- fișa de observare a profesorului privind implicarea în activități și discuții;
- studiu de caz (grafuri ponderate);
- fișa de autoevaluare a elevului.

Criterii de evaluare:

- | | |
|---|--------------|
| ➤ Proiectarea algoritmilor fundamentali (reprezentare, parcurgere, conexitate, optimalitate) | 4x0.5 puncte |
| ➤ Realizarea câte unui tip de reprezentare pentru grafuri (matrice de adiacență, liste de vecini) | 2x0.5 puncte |
| ➤ Utilizarea metodelor de parcurgere a grafurilor (adâncime, lățime) | 2x1 punct |
| ➤ Realizarea, cu resurse proprii, a cel puțin două dintre programele prezentate | 2x1 punct |
| ➤ Observarea unui progres sau atingerea standardului maxim de competență | 1 punct |
| ➤ Manifestarea unei atitudini constructive în activitatea de grup | 0.5 puncte |
| ➤ Capacitatea de autoevaluare | 0.5 puncte |
| ➤ Punctaj acordat din oficiu | 1 punct |

4.4. OBSERVAREA SISTEMATICĂ A ACTIVITĂȚII ȘI COMPORTAMENTULUI ELEVILOR

Fișa de observare a activității și comportamentului elevului înregistrează informații legate de particularitățile personalității elevului manifestate în procesul didactic, de achizițiile evaluate spontan (răspunsuri sporadice, atitudini semnificative etc.), de progresul înregistrat de acesta. Profesorul construiește această fișă în vederea individualizării procesului sumativ de evaluare, dar și a celui de învățare.

Fișa de observare poate să surprindă:

- modul în care elevul își expune cunoștințele și capacitățile de investigare;
- atitudinea față de evenimentele studiate;
- interesul;
- adaptarea socială.

Un model orientativ de fișă de observare conține:

- date generale despre elev (nume, prenume, vârstă, climat educativ, condiții materiale, particularități socio-comportamentale);
- particularități ale proceselor intelectuale (gândire, limbaj, imaginație, memorie, atenție, spirit de observație etc.);
- aptitudini și interese manifestate;
- particularități afectiv-motivaționale;
- trăsături de temperament;
- atitudini și relaționare (cu sine însuși, cu materia studiată, cu colegii);
- considerații privind evoluția aptitudinilor, atitudinilor, intereselor și nivelului de integrare.

Completarea fișei se realizează în timp într-un ritm adecvat specificului activităților de la disciplină, din anul și de la clasa respectivă, dar și în funcție de implicarea și de ritmul individual al elevului.

EXEMPLU

Clasa: a IX-a

Competențe:

- rezolvarea de probleme;
- realizarea unui program;
- construirea de aplicații;
- independența cognitivă;
- comunicarea și integrarea.

Criterii de evaluare: Fiecare aspect urmărit prin fișa de observare trebuie să fie cuantificat utilizând standarde de nivel: nivel minimal, mediu, superior.

De exemplu, pentru obiectivele formulate mai sus se poate întocmi o detaliere a criteriilor de evaluare:

Obiectiv	Standard minim	Standard mediu	Standard superior
Realizarea unui program	Editare, compilare și rulare	Urmărire și depanare	Identificarea algoritmului, metodelor și tehnicilor folosite
Rezolvarea de probleme	Reproducere și înțelegere enunț	Identificare metodă și tehnici de realizare	Identificare algoritm optim conform criteriilor prestabilite
Construirea de aplicații	Identificare specificații	Implementare și testare	Dezvoltare și personalizare
Independența cognitivă	Cunoașterea termenilor	Utilizarea contextuală	Adaptarea la context
Comunicarea și integrarea	Comunicare de idei	Argumentare, raportare la sarcina proprie	Motivare, acte decizionale, corelare cu grupul

Prin stabilirea achizițiilor cognitive fișa de observare poate să conțină o detaliere a comportamentului elevului:

Comportament	DA	NU
Reproduce și înțelege		
Identifică specificații		
Cunoaște termenii		
Identifică metode și tehnici de realizare		
Implementează și testează		
Utilizează contextual instrumentele de lucru		
Identifică metode de rezolvare optimă, conform criteriilor prestabilite		
Dezvoltă și personalizează lucrările primite ca temă de lucru		
Adaptează noțiunile potrivindu-le la context		
Memorează numai după mai multe repetări ale aceleași noțiuni		
Pentru a înțelege are nevoie de material vizual		
Memorează mecanic reguli		
Sesizează legături logice între noțiuni		
Produce rapid mai multe idei pe o temă dată		
Repetă numai ce au spus ceilalți		
Respectă proporții reale ale obiectelor desenate sau modelate		
Manipulează cu ușurință instrumentele		
Înțelege cu ușurință noile aplicații		

4.5. HĂRȚILE CONCEPTUALE

Hărțile conceptuale (conceptual maps) sau hărțile cognitive (cognitive maps) se definesc ca fiind o imagine a modului de gândire, simțire și înțelegere ale celui sau celor care le elaborează, devenind o modalitate, o procedura de lucru la diferite discipline, dar și inter și transdisciplinar.

Această procedură poate fi folosită: în predare, în învățare dar și în **evaluare**.

Esența cunoașterii constă în modul cum se structurează cunoștințele. Important este nu cât cunoști, ci relațiile care se stabilesc între cunoștințele asimilate.

Avantaje ale hărților conceptuale:

- organizează cunoștințele existente în mintea elevului;
- pregătește noile asimilări;
- ajută la organizarea planificării sau proiectării unei activități;
- elimină memorizarea și simpla reproducere a unor definiții sau algoritmi de rezolvare a unei probleme;
- învățarea devine activă și constantă;
- permit **vizualizarea** relațiilor dintre cunoștințele elevului;
- **evaluarea** pune în evidență modul cum gândește elevul și cum **folosește ceea ce a învățat**;

Hărțile conceptuale solicită mult timp (deci un alt mod de organizare a evaluării), nivelul standardelor este ridicat (deci evaluarea se face pe finalități ale curriculumului) iar elevul trebuie să respecte o rigoare și o ordine deosebite.

Procesul elaborării hărților conceptuale în grup cuprinde 6 etape:

Etapa 1: PREGĂTIREA

- selectarea partenerilor;
- stabilirea temei de lucru;

Etapa 2: GENERAREA IDEILOR, A AFIRMAȚIILOR

- definirea conceptelor, argumentarea folosirii lor;

Etapa 3: STRUCTURAREA AFIRMAȚIILOR

- selectarea ideilor;
- clasarea lor;

Etapa 4: REPREZENTAREA GRAFICĂ

- elaborarea hărții conceptuale;

Etapa 5: INTERPRETAREA, EVALUAREA COLECTIVĂ A HĂRȚII CONCEPTUALE

- verificarea listei de concepte;
- analiza relevanței conceptelor pentru scopurile propuse;
- analiza legăturilor și a afirmațiilor ce leagă conceptele;

Etapa 6: UTILIZAREA HĂRȚII CONCEPTUALE

- pentru planificarea, proiectarea activității, a proiectelor de dezvoltare și evaluare;

EXEMPLU

Clasa: a XI-a

Timp de lucru: 100 min

Competență specifică: Caracterizarea metodelor studiate (Backtracking, Divide et impera, Greedy, Programare dinamică)

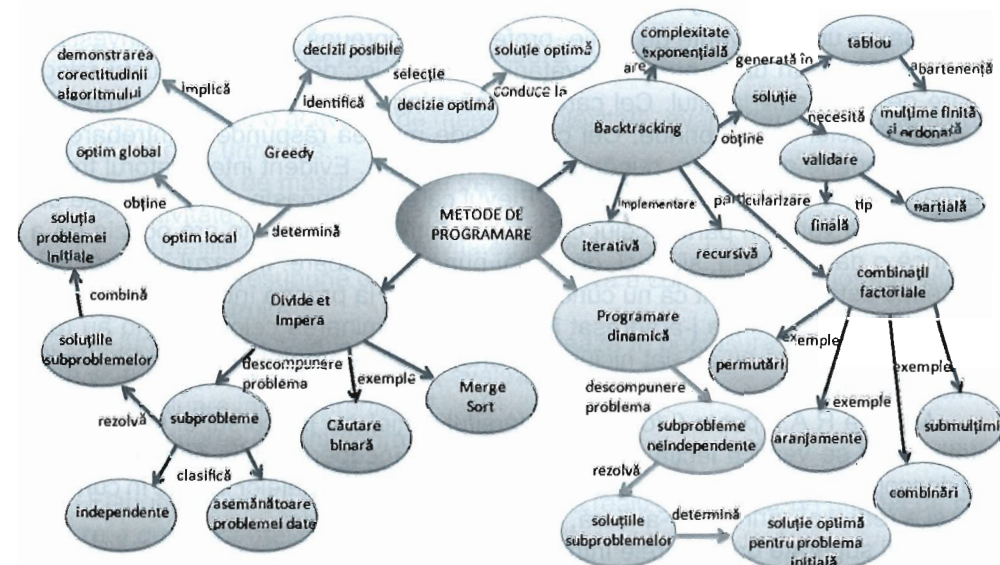
Competențe de evaluat:

- Caracterizarea metodelor studiate (Backtracking, Divide et impera, Greedy, Programare dinamică)
- Descrierea etapelor de rezolvare a unei probleme din fiecare clasă
- Identificarea unor probleme cunoscute dintr-o clasă dată
- Clasificarea unui set de probleme date în mulțimi corespunzătoare metodelor învățate

Se prezintă tema și obiectivele urmărite pe parcursul evaluării. Se cere elevilor să identifice conceptele utilizate în caracterizarea celor patru metode de programare. Vor fi selectate conceptele de bază:

- | | |
|--------------------|-----------------------|
| x Alegere | x Programare dinamica |
| x Backtracking | x Recursivitate |
| x Combinare | x Revenire |
| x Combinatorică | x Similare |
| x Divide et impera | x Soluție optimă |
| x Generare | x Spațiul soluțiilor |
| x Greedy | x Subprobleme |
| x Înaintare | x Subsoluții |
| x Încercări | x Subsoluții optime |
| x Independente | x Validare finală |
| x Optim global | x Validare parțială |
| x Optim local | x Vector |

Se grupează informațiile pe categorii, fiecare dintre ele caracterizând una dintre metodele de programare studiate. Fiecare grupă (4-5 elevi) realizează, în format digital, o reprezentare grafică a relațiilor dintre conceptele selectate și adaugă pe hartă expresii verbale care să indice relațiile dintre acestea.



Fiecare grupă prezintă colegilor interpretarea hărții. Se analizează legăturile și afirmațiile ce leagă conceptele.

Exemple:

Utilizarea metodei *Greedy* presupune selectarea deciziei optime pe baza deciziilor posibile, decizie care conduce la obținerea unei soluții optime.

Rezolvarea unei probleme cu ajutorul metodei *Divide et Impera* presupune descompunerea problemei în subprobleme independente sau asemănătoare problemei date, soluția problemei inițiale obținându-se prin combinarea soluțiilor subproblemelor.

Barem de evaluare și notare:

- a) corectitudine – $1 \times 4 = 4$ puncte
- b) completitudine (cuvinte, verbe) – $0.5 \times 4 = 2$ puncte
- c) organizarea informației (grupări, distribuii) – $0.25 \times 4 = 1$ punct
- d) calitatea grafică a prezentării – $0.25 \times 4 = 1$ punct
- e) calitatea prezentării orale – 2 puncte

Total :10 puncte

4.6. METODA R. A. I.

Metoda R. A. I. are la bază stimularea și dezvoltarea capacităților elevilor de a comunica (prin întrebări și răspunsuri) ceea ce tocmai au învățat. Denumirea provine de la inițialele cuvintelor **R**ăspunde – **A**runcă – **I**nteroghează și se desfășoară astfel: la sfârșitul unei lecții sau a unei secvențe de lecție, profesorul, împreună cu elevii săi, investighează rezultatele obținute în urma predării-învățării, printr-un joc de aruncare a unei mingi mici și ușoare de la un elev la altul. Cel care aruncă mingea trebuie să pună o întrebare din lecția predată celui care o prinde. Cel care prinde mingea răspunde la întrebare și apoi aruncă mai departe altui coleg, punând o nouă întrebare. Evident interogatorul trebuie să cunoască și răspunsul întrebării adresate. Elevul care nu cunoaște răspunsul iese din joc, iar răspunsul va veni din partea celui care a pus întrebarea. Acesta are ocazia de a mai arunca încă o dată mingea, și, deci, de a mai pune o întrebare. În cazul în care, cel care interoghează este descoperit că nu cunoaște răspunsul la propria întrebare, este scos din joc, în favoarea celui căruia i-a adresat întrebarea. Eliminarea celor care nu au răspuns corect sau a celor care nu au dat niciun răspuns, conduce treptat la rămânerea în grup a celor mai bine pregătiți.

Metoda R.A.I. poate fi folosită la sfârșitul lecției, pe parcursul ei sau la începutul activității, când se verifică lecția anterioară, înaintea începerii noului demers didactic, în scopul descoperirii, de către profesorul ce asistă la joc, a eventualelor lacune în cunoștințele elevilor și a reactualizării ideilor-ancoră.

Pot fi sugerate următoarele întrebări:

- Ce știi despre.....?
- Care sunt ideile principale ale lecției.....?
- Despre ce ai învățat în lecția.....?
- Care este importanța faptului că.....?
- Cum justifici faptul că.....?
- Care crezi că sunt consecințele faptului.....?
- Ce ai vrea să mai afli în legătură cu tema studiată (predată).....?
- Ce întrebări ai în legătură cu subiectul propus.....?
- Cum consideri că ar fi mai avantajos să.....sau să.....?
- Ce ți s-a părut mai dificil din.....?
- Cum poți aplica cunoștințele învățate.....?
- Ce ți s-a părut mai interesant.....?
- De ce alte experiențe sau cunoștințe poți lega ceea ce tocmai ai învățat?

Metoda R.A.I. este adaptabilă oricărui tip de conținut.

Este o metodă de a realiza un feed-back rapid, într-un mod plăcut, energizant și mai puțin stresant decât metodele clasice de evaluare. Se desfășoară în scopuri constataktiv-ameliorative și nu în vederea sancționării prin notă sau calificativ.

Metoda R.A.I. poate fi folosită și pentru verificarea cunoștințelor pe care elevii și le-au dobândit independent prin studiul bibliografiei recomandate. Accentul se pune pe ceea ce s-a învățat și pe ceea ce se învață în continuare prin intermediul creării de întrebări și de răspunsuri

EVALUAREA COMPETENȚELOR SPECIFICE. EXEMPLE DE PROBE DE EVALUARE

Pașii alcătuirii unei probe de evaluare

Reprezentând o activitate de mare complexitate, evaluarea subsumează două demersuri, după cum urmează:

- activitate de măsurare care trebuie să fie foarte riguroasă și foarte precisă;
- activitate de apreciere care trebuie să acorde semnificațiile curente versus informațiile obținute prin activitatea de măsurare.

Măsurarea presupune o descriere cantitativă a comportamentelor formate la elevi în urma realizării instruirii.

În funcție de momentul în care se integrează în desfășurarea procesului didactic, precum și în funcție de scopul urmărit, evaluarea poate fi:

- evaluare inițială/predictivă
- evaluarea continuă/formativă
- evaluare finală/sumativă

Elaborarea **competențelor de evaluat** se realizează printr-un proces de **operationalizare a competențelor** din programa școlară.

Competențele de evaluat:

- au caracter derivat în raport cu competențele specifice din programa școlară;
- au un grad de specificitate care permite o evaluare educațională validă și fidelă, deoarece acest lucru se realizează pe baza aprecierii, cu ajutorul unui instrument de evaluare, a comportamentului cognitiv și formativ al elevilor.

Elaborarea unei competențe de evaluat presupune:

- specificarea comportamentului vizat, pe care elevul trebuie să-l demonstreze;
- precizarea condițiilor în care comportamentul se poate produce sau poate deveni vizibil, măsurabil, evaluabil;
- stabilirea unui nivel al performanței acceptabile, prin enunțarea unui criteriu de reușită direct măsurabil – exprimabil prin niveluri de cunoaștere

Pornind de la competențele generale, relaționate cu competențele specifice în **procesul de evaluare se identifică următoarele etape:**

- stabilirea competențelor de evaluat;
- stabilirea metodelor și instrumentelor de evaluare, însoțite de criteriile de notare și de baremele de evaluare;
- acordarea notelor.

Criteriile de notare joacă un rol important atât în ceea ce privește evaluarea obiectivă a elevilor, cât și diminuarea diferențelor de notare dintre elevii aceleiași clase, dintre elevii aceluiași an de studiu, la nivel de școală și la nivel național.

Evaluarea rezultatelor și progreselor obținute de elevi la informatică urmărește să măsoare și să aprecieze progresele elevilor în materie de cunoștințe, priceperi și deprinderi informatice, ca rezultate ale procesului de instruire, precum și aspecte educative ale activității școlare la informatică, materializate în atitudinile și comportamentul elevilor.

O probă de evaluare trebuie să îndeplinească anumite exigențe de elaborare (calități tehnice) în vederea atingerii scopului pentru care acesta a fost proiectată.

O probă de evaluare este compusă dintr-un număr de itemi care, pe de o parte au reguli precise de elaborare, iar pe de altă parte sunt selectați pe baza unei matrice de specificații.

Principalele calități ale unui instrument de evaluare sunt: **validitatea, fidelitatea, obiectivitatea și aplicabilitatea**.

Validitatea reprezintă calitatea **testului de a măsura ceea ce este destinat să măsoare** (competențele de evaluat).

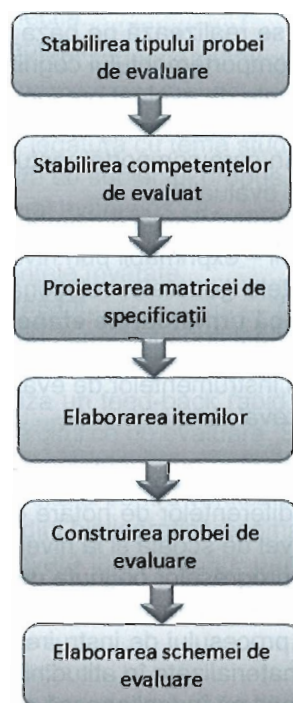
Fidelitatea reprezintă calitatea unui test de a produce rezultate comparabile în cursul aplicării sale repetate.

Obiectivitatea reprezintă gradul de concordanță între aprecierile făcute de către evaluatori independenți în ceea ce privește un răspuns corect pentru fiecare din itemii testului. Cu alte cuvinte, un test are calitatea de a fi obiectiv, dacă evaluatori diferiți aplică în mod unitar baremul de evaluare și de notare.

Aplicabilitatea reprezintă calitatea testului de a fi administrat și interpretat cu ușurință. Criteriile de selectare a testelor cu o bună aplicabilitate sunt :

- specificul competențelor evaluate prin test;
- concordanța dintre forma și conținutul testului, pe de o parte, și nivelul de vârstă al elevilor, pe de altă parte;
- timpul și costurile necesare pentru administrarea testului;
- obiectivitatea în notare și interpretarea rezultatelor.

În proiectarea unei probe de evaluare trebuie avute în vedere următoarele etape:



Proiectarea matricei de specificații

„Matricea de specificații constă într-un tabel cu două intrări care servește la proiectarea și organizarea itemilor dintr-un test docimologic, în care sunt precizate, pe de o parte, conținuturile care vor fi vizate, și, pe de altă parte, nivelurile taxonomice la care se plasează competențele de evaluat.”¹

Matricea de specificații indică ceea ce urmează a fi testat - competențele de evaluat prin raportare la conținuturile învățării.

O matrice de specificații detaliată trebuie să precizeze competențele formate prin procesul didactic pentru fiecare unitate tematică parcursă într-o anumită perioadă de timp.

Profesorul evaluator stabilește procentele ce urmează a fi evaluate din fiecare domeniu/conținut/temă raportate la nivelurile cognitive/competențele specificate în matrice.

Exemplu:

Niveluri cognitive	Achiziția informației	Înțelegere	Aplicare	Analiză	Pondere %
Competențe de evaluat / Conținuturi	c1	c2	c3	c4	
Element de conținut 1	1	3	3	3	10
Element de conținut 2	2,5	7,5	7,5	7,5	25
Element de conținut 3	3,5	10,5	10,5	10,5	35
Element de conținut 4	3	9	9	9	30
Pondere %	10	30	30	30	100

Completarea celulelor matricei se realizează prin înmulțirea valorilor de pe ultima linie cu valorile de pe ultima coloană ($A_{ij} = A_{ni} \cdot A_{js}$). De exemplu, ponderea „Achiziției de informații” raportată la elementul de conținut 1 este: $10\% \times 10\% = 1$.

Profesorul stabilește numărul total de itemi pe care dorește să îl conțină testul (de exemplu, 20 de itemi), după care completează fiecare celulă a matricei utilizând formula: **procentaj/100 x nr.total de itemi**.

Astfel, rezultă o **a doua matrice care specifică numărul itemilor** care trebuie elaborați în funcție de competențele de evaluat stabilite și elementele de conținut abordate.

Niveluri cognitive	Achiziția informației	Înțelegere	Aplicare	Analiză	Pondere %
Competențe de evaluat / Conținuturi	c1	c2	c3	c4	
Element de conținut 1	0,2	0,6 (1 item)	0,6 (1 item)	0,6	2
Element de conținut 2	0,5	1,5 (1 item)	1,5 (1 item)	1,5 (1 item)	5
Element de conținut 3	0,7 (1 item)	2,1 (2 item)	2,1 (2 item)	2,1 (2 item)	7
Element de conținut 4	0,6 (1 item)	1,8 (2 item)	1,8 (2 item)	1,8 (2 item)	6
Pondere %	2	6	6	6	20

¹ Mason și Bramble, 1997; Schreerens, Glas și Thomas, 2003; Gall, Gall și Borg, 2007

1. EXEMPLE DE PROBE DE EVALUARE PREDICTIVĂ

Calitatea unei învățări noi depinde de calitatea învățărilor anterioare și de nivelul motivațional. Dacă la un moment dat vrem să continuăm instruirea unui elev, trebuie să știm exact ce știe să facă elevul până în acel moment.

Înainte de a declanșa un nou proces instructiv-educativ, profesorul trebuie să examineze minuțios starea inițială a pregătirii elevilor și capacitatea lor de învățare.

Calitatea achizițiilor dobândite pe parcursul instruirii anterioare condiționează calitatea și eficiența instruirii ce va urma.

Profesorul trebuie

- să determine precis nivelul de atingere a obiectivelor materiei anterior parcurse de elev;
- să elaboreze, să aplice și să examineze detaliat rezultatele unui test predictiv;
- să stabilească programe compensatorii.

Testul predictiv înseamnă un test inițial – aplicat la începutul unei noi etape de instruire pentru a identifica nivelul de realizare a obiectivelor studiului într-o etapă anterioară, riguros delimitată și lacunele intervenite în pregătirea fiecărui elev al clasei pe parcursul instruirii sau ulterior.

Lista de competențe asociate conținuturilor studiate în etapa anterioară constituie baza derivării itemilor care alcătuiesc testul predictiv. Regula simplă de elaborare a unui test predictiv este următoarea :

Pentru fiecare competență de evaluat asociată unui conținut studiat în etapa precedentă trebuie elaborat un item care verifică atingerea acestuia la un nivel de performanță suficient pentru ca elevul să poată continua adecvat instruirea.

Baza testului predictiv o constituie competențele atinse prin instruirea deja desfășurată, dar, în elaborarea testului predictive trebuie ținut seama și de ceea ce urmează să învețe elevii.

Numai în acest fel se pot stabili performanțe minimal acceptabile pe baza cărora se anticipează posibilitatea continuării instruirii în ritmul impus de parcurgerea programei școlare.

Calitățile unui test predictiv:

- validitatea predictivă a testului inițial este asigurată dacă acesta este astfel construit încât să poată indica în ce fel se poate continua instruirea fiecăruia dintre cei care au fost testați;
- un test predictiv poate fi considerat reprezentativ dacă verifică esențialul întregii materii parcurse anterior;
- un test predictiv este eficient dacă identifică exact nivelul de performanță de care este capabil și toate lacunele esențiale care au intervenit în instruirea anterioară a elevului;
- un test predictiv este aplicabil dacă oferă datele necesare diagnosticului și remedierii.

1.1. EVALUARE PREDICTIVĂ LA ÎNCEPUT DE CICLU LICEAL

Întrucât studiul disciplinei informatică devine obligatoriu din prima clasă de liceu, evaluarea predictivă trebuie să aibă în vedere cunoștințele dobândite la celelalte discipline care ar putea oferi informații referitoare la capacitatea de analiză și sinteză a informației, identificarea ordinii de efectuare a unor operații matematice și logice, dezvoltarea unei „gândiri algoritmice”.

Clasa: a IX-a

Timp de lucru: 30 min

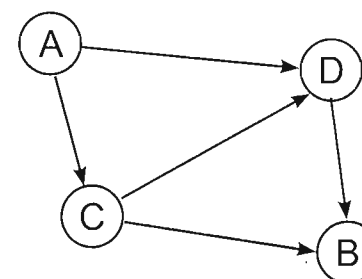
Competență specifică: Descrierea unei succesiuni de operații prin care se obțin, din datele de intrare, datele de ieșire

Competențe de evaluat:

- Identificarea relațiilor dintre date
- Analizarea enunțului unei probleme
- Descompunerea rezolvării unei probleme în pași
- Identificarea pașilor de prelucrare a datelor
- Identificarea tuturor pașilor de prelucrare posibili la un moment dat
- Alegerea din mai multe posibilități de prelucrare a celei care corespunde cerinței
- Alegerea unei ordini de prelucrare a datelor care să corespundă cerinței
- Stabilirea valorii unor date specifice (rezultate, date de intrare pentru un rezultat specificat etc.) pentru un algoritm dat
- Descrierea în limbaj natural a unor secvențe de prelucrare date
- Legarea coerentă a descrierilor pentru obținerea unui enunț

Itemi semiobiectivi (cu răspuns scurt, de completare)

- 1) Ce număr urmează imediat după 97 în sirul **17, 37, 47, 67, 97, ...** ?
- 2) Nicușor se află în orașul A și dorește să ajungă în orașul B. Nu există o legătură directă între cele două orașe. Din orașul A poate să meargă doar în orașul C sau în orașul D. Din orașul C poate pleca numai către orașele D și B iar din orașul D doar B. Scrieți toate modurile în care Nicușor poate să ajungă în orașul B, plecând către din orașul A.



- 3) Se consideră suprafața din imaginea alăturată. Se dorește secționarea acestei suprafețe în patru părți egale (de aceeași formă și aceeași arie). În cazul în care suprafața poate fi secționată conform cerinței, colorați diferit fiecare dintre cele patru suprafețe formate.



- 4) Câte numere cu trei cifre se pot scrie folosind cifrele 0, 1 și 2 ?
- 5) Alina face clătite și are toate ingredientele (ouă, lapte, făină, ulei, sare, zahăr). A pierdut însă rețeta și a reținut doar primii doi pași:
a) se pune laptele la fiert;
b) după ce s-a răcit laptele, se adaugă sarea și zahărul;
Scrieți în continuare o succesiune de pași care să conducă la reconstituirea rețetei.
- 6) Toate figurile geometrice din imaginea alăturată respectă aceeași regulă.
Ce număr trebuie scris în triunghi pentru ca acesta să respecte regula?
- 7) Se consideră un număr natural n și operațiile elementare:
 O_1) adună 2
 O_2) împarte la 3
Indicați o succesiune de astfel de operații, care, pornind de la valoarea inițială $n = 10$ să determine obținerea valorii 8.
- 8) Precizați elementele mulțimii $A = \{n \in \mathbb{N}^* | n \leq 10, (\text{rest}(n/3) - 2) \cdot (\text{rest}(n/3) - 1) = 0\}$.



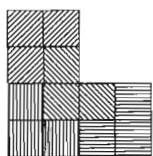
Barem de evaluare și notare:

Se acordă câte un punct pentru fiecare dintre itemi și 2p din oficiu.

1) 107;

2) ACB, ACDB, ADB

3)



- 4) 18 (cifra sutelor poate fi 1 sau 2, cifra zecilor oricare din cele 3, cifra unităților oricare din cele 3 deci 18 numere: $18 = 2 \times 3 \times 3$);
5) se punctează orice succesiune corectă;
6) 5 (suma dintre numărul înscris în fiecare figură geometrică și numărul de unghiuri din acea figură geometrică trebuie să fie 8);
7) $O_1 O_2 O_1 O_1$ ($((((10+2)/3)+2)+2)$);
8) $A = \{1, 2, 4, 5, 7, 8, 10\}$ (elementele din intervalul $[1, 10]$ care nu sunt divizibile cu 3)

Observație. Testul predictiv la început de ciclu liceal poate să conțină și întrebări referitoare la cunoșterea limbajelor de programare.

1.2. PROBĂ DE AMELIORARE ȘI PROGRES

Un test predictiv este aplicabil dacă și numai dacă oferă date utile atât elevului cât și profesorului, asigurând un feed-back diferențial. Un test predictiv este inaplicabil dacă nu oferă datele necesare ameliorării și progresului.

Programele compensatorii sunt programe de instruire suplimentară care urmăresc ameliorările care se vor produce în comportamentul de învățare al elevilor, în vederea atingerii sau depășirii standardelor de performanță solicitate de programele școlare.

Acestea pot fi :

- ➔ programe de recuperare : programe suplimentare destinate elevilor cu lacune esențiale în instruirea anterioară, organizate în vederea atingerii performanțelor minimal acceptabile;
- ➔ programe de îmbogățire: programe suplimentare destinate elevilor capabili de performanțe superioare standardelor prevăzute de programa școlară.

Clasa a X-a

Timp de lucru: 30 min

Competențe specifice:

Identificarea tipurilor de date necesare pentru rezolvarea unei probleme.

Identificarea relațiilor dintre date

Competențe de evaluat :

Identificarea numărului, numelui și valorilor posibile pentru datele de intrare și de ieșire

Enumerarea rezultatelor ce se pot obține dintr-un set de date

Identificarea structurii de date necesare pentru memorarea unor date

Itemi obiectivi cu alegere duală:

Enunț: Citiți următoarele afirmații. Încercuiți varianta **A** dacă afirmația este adevărată și varianta **F** dacă aceasta este falsă.

1. Setul de caractere folosit pentru descrierea programelor în C++ este format din: - literele alfabetului englez; - cifrele 0, ..., 9; - caractere speciale sau acestea sunt codificate prin întregi din intervalul $[1, 127]$.	A F
2. Cuvintele cheie în limbajul C++ pot fi folosite ca identificatori oarecare în descrierea unui program C++.	A F
3. <i>tip de date</i> se caracterizează prin mulțimea valorilor pe care le pot lua datele respective, modul de reprezentare în memoria calculatorului precum și operațiile și funcțiile ce se pot efectua cu datele respective.	A F
4. Comentariile NU pot fi considerate separatori universali ai unităților sintactice.	A F
5. Pentru a specifica absența oricărei valori există tipul special <i>void</i> . Mulțimea valorilor sale este mulțimea vidă.	A F
6. Modificatorii de tip <i>unsigned</i> și <i>long</i> pot fi aplicați tipului de date <i>int</i> : <i>long</i> va modifica dimensiunea reprezentării, iar <i>unsigned</i> va face ca în reprezentare să fie folosit și bitul de semn rezultând un interval de numere naturale.	A F

Barem de evaluare și notare: 1. – A.; 2. – F.; 3. – A.; 4. – F.; 5. – A.; 6. – A.

Se acordă câte 5 puncte pentru fiecare dintre itemii pentru care răspunsul a fost corect.
Total 30 de puncte.

Itemii obiectivi de tip pereche:

Enunț: Alegeți pentru elementele descrise în prima coloană elementele corespunzătoare din a doua coloană.

1. Tipurile de date standard în C++	a. 2 octeți
2. Numărul de octeți necesari pentru a reprezenta o variabilă de tip int sau de tip unsigned int în memoria calculatorului	b. un singur tip de dată
3. „O secvență de caractere scrisă între ghilimele”	c. 4 octeți
4. „O variabilă poate reține valori de”	d. Tipuri întregi și tipuri reale
5. Numărul de octeți necesari pentru a reprezenta în memoria calculatorului variabila e , definită astfel long e=24*32+10;	e. constantă șir de caractere
6. Caracterele speciale neimprimabil pot fi folosite în limbajul C++	f. codul ASCII cuprins în intervalul [0, 32]

Barem de evaluare și notare: 1. – d.; 2. – a.; 3. – e.; 4. – b.; 5. – c.; 6. – f.

Se acordă câte 5 puncte pentru fiecare dintre itemii pentru care răspunsul a fost corect. Total 30 de puncte

Itemii obiectivi cu alegere multiplă:

Enunț: Stabiliți pentru fiecare dintre enunțurile date mai jos care este varianta de răspuns corectă.

1. NU constituie constante de tip șir de caractere:	a. "xYz" b. '123' c. "x56\x24"
2. Fiind date două variabile de tip int a și b , a=32760 , iar b=10 , expresia a+b va avea valoarea:	a. – 32766 b. nu se poate calcula c. 32770
3. Fie declarația de variabile: int x=15426, y=0x3c42; ce relație există între valorile celor două variabile?	a. a<b b. a>b c. a=b
4. Dacă c este o variabilă de tip char c = 'd' indicați valoarea lui c după atribuirea: c = c - 'a' + 'A';	a. atribuirea nu este corectă b. c = 'D' c. c = 'd'
5. Se definesc variabilele float a, b, c, d; Indicați atribuirea care determină memorarea în d a mediei aritmetice a celor trei valori reținute în variabilele a, b, c.	a. d=(a+b+c)/3; b. d=a+b+c/3; c. d=(a+b+c)/2.
6. Fie declarația de variabile int a=15, b=17, c=19, d=21, e; Indicați secvența de instrucțiuni în urma executării căreia variabilele au valorile a=21, b=15, c=17, d=19.	a. e=a; a=b; b=c; c=d; d=e; b. e=a; a=d; b=a; c=b; d=e; c. e=d; d=c; c=b; b=a; a=e;

Barem de evaluare și notare: 1. – b.; 2. – a.; 3. – c.; 4. – b.; 5. – a.; 6. – c.

Se acordă câte 5 puncte pentru fiecare dintre itemii pentru care răspunsul a fost corect.

Total 30 de puncte

Din oficiu: 10 puncte

1.3. EVALUARE PREDICTIVĂ LA ÎNCEPUTUL CICLULUI SUPERIOR AL LICEULUI

O primă direcție de acțiune în procesul complex al evaluării rezultatelor școlare la informatică este impusă de necesitatea diagnosticării competențelor elevilor, în vederea proiectării eficiente și realiste a unei secvențe de instruire. De exemplu, abordarea capitolului „Aplicarea mecanismului recursivității prin crearea unor subprograme recursive” poate fi precedată de un test diagnostic, prin care vor fi cunoscute achizițiile și eventualele lacune în cunoștințele legate de utilizarea subprogramelor:

Clasa a XI-a

Timp de lucru: 30 min

Competențe specifice: Utilizarea subprogramelor

Competențe de evaluat :

Descompunerea unei cerințe compuse în cerințe simple.

Recunoașterea metodelor de refolosire a codului unei secvențe de prelucrare

Identificarea datelor de intrare, a datelor de ieșire și a celor de intrare-ieșire într-o prelucrare modulară

Diferențierea definirii unui subprogram de apelul acelui subprogram

Itemii semiobiectivi (cu răspuns scurt, de completare, întrebări structurate)

Enunț:

Fie funcția definită în limbajul C++

```
long f(long x)
{ long y;
  y=x/1000*100+x%100;
  return y;
}
```

- Dați un exemplu de valoare pe care o poate avea parametrul **x** astfel încât rezultatul obținut în urma apelului funcției să fie 1245.
- Pentru funcția dată parametrul **x** este transmis prin.....?
- Scrieți numărul afișat în urma executării instrucțiunii `cout<<f(8765432);`
- Scrieți ce se va afișa în urma executării programului următor:

```
int main()
{ long a=12345, b;
  cout<<a<<'n';
  b=f(a);
  cout<<a<<'n'<<b<<'n';
  return 0;
}
```


e. Dacă funcția devine:

```
long f(long &x)
{
    x=x/1000*100+x%100;
    return x;
}
```

Scrieți ce se va afișa în urma executării programului descris la punctul d.

Barem de evaluare și notare:

Se acordă câte 18 puncte pentru fiecare dintre itemii pentru care răspunsul a fost corect.

Total 90 de puncte
Din oficiu: 10 puncte

a. Orice număr de forma $x=12a45$ unde $x=\{0, 1, 2, \dots, 9\}$; rezultatul returnat de funcție este un număr natural din care s-a eliminat cifra miilor, dacă $x \geq 1000$ sau valoarea x dacă $x < 1000$.

b. valoare.

Întrebări sugerate:

„Ce anume se reține în cazul parametrilor transmiși prin valoare în segmentul de stivă?”

„În ce ordine se rețin parametrii unei funcții?”

„Ce se întâmplă la revenirea în blocul apelant cu valorile variabilelor memorate în stivă?”

c. 876532.

d. 12345

12345

1245

e. 12345

1245

1245

Întrebări sugerate:

„Cum se vor transmite parametrii efectivi la apelul funcției în acest caz?”

Observații.

Clasa va fi organizată pe grupe formate din 3-4 elevi.

Întrebările vor avea ca scop o mai bună înțelegere a transmiterii parametrilor unei funcții.

Obiectivul urmărit este ca răspunsurile date să reprezinte o sinteză de cunoștințe sau un rezultat al înțelegerii unei situații și nu reproducerea unor informații prezentate anterior.

2. EXEMPLE DE PROBE DE EVALUARE CONTINUĂ

Evaluarea continuă la informatică vizează confirmarea atingerii obiectivelor propuse pentru o secvență didactică, stabilirea nivelului la care a ajuns fiecare elev în procesul formării setului de competențe implicat de aceste obiective.

2.1. EVALUARE ORALĂ

Metoda principală de evaluare, evaluarea orală poate fi realizată în diferite momente ale desfășurării lecției și permite aprecierea participării elevilor, precum și calitatea acestei participări la lecție. În cadrul acestei metode de examinare, elevii sunt puși în situația de a reproduce definiții, proprietăți, reguli, de a efectua exerciții cu grade diferite de dificultate, de a rezolva și compune probleme, verbalizând, exprimând într-un limbaj informatic corect și coerent judecățile făcute. Nu este necesar ca toate examinările orale să fie sancționate prin note, la sfârșitul lecției. Profesorul poate păstra o evidență a rezultatelor examinării orale zilnice ale elevilor, ce se pot transforma, o dată sau de două ori pe semestru, în calificative ce oglindesc activitatea elevilor.

Organizarea grafică a informațiilor în evaluarea orală

Studiile de psihologie cognitivă susțin: cu cât este mai adâncă procesarea unui stimul cu atât mai bine este reținut în memoria de lungă durată.

Prin organizarea grafică a unui conținut, elevii analizează ideile, le compară pentru a extrage esențialul, deduc conceptele, apoi ordonează informațiile în diverse ansambluri, după diferite criterii (temporale, ierarhice, spațiale, etc.), pe baza cerințelor logice și a domeniului studiat.

Elaborarea organizatorilor grafici în evaluare prezintă multiple avantaje pentru elevi:

- învață să extragă informațiile esențiale dintr-un conținut;
- identifică unele conexiuni existente între diferite aspecte din realitate;
- stabilesc ierarhii, cronologii și alte ansambluri organizate de cunoștințe;
- exersează principalele operații ale gândirii (analiza, comparația, sinteza, generalizarea, abstractizarea, etc)
- stabilesc conexiuni logice între vechile cunoștințe și noile înforații;
- prin reflecții și restructurări succesive ale informațiilor acestea sunt învățate mai ușor și devin cunoștințe;
- cunoștințele dobândite prin procesare profundă sunt durabile și operaționale.

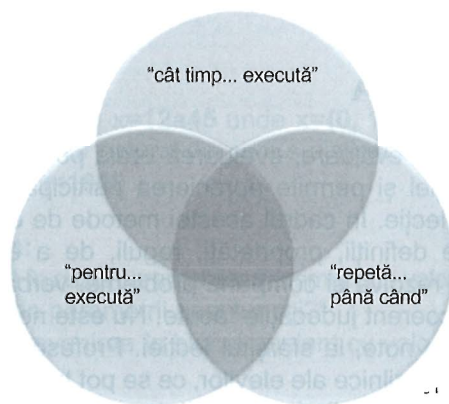
DIAGRAMA VENN este formată din cercuri care se suprapun parțial. În arealul în care cercurile se suprapun se grupează asemănările, iar în arealele rămase libere deosebirile dintre aspectele sau conceptele evaluate.

Clasa a IX-a

Competență specifică. Reprezentarea algoritmilor în pseudocod

Competențe de evaluat: Analiza comparativă a structurilor repetitive

Comunicarea sarcinii de lucru. Imaginea de pe tablă/monitor este formată cu ajutorul a trei cercuri care se intersectează. Fiecare cerc corespunde structurii pseudocod de control scrise în interior. Completați imaginea astfel încât în zonele care se suprapun să se afle elementele comune structurilor pseudocod corespunzătoare, iar în zonele care nu se suprapun să se găsească elemente proprii structurii descrise.



Organizarea activității.

Activitatea se poate organiza:

- *pe grupe:* fiecare grupă prezintă caracteristicile uneia dintre structuri. Acestea sunt scrise în cercuri, inițial în partea necomună acestora. Proprietățile comune sunt apoi mutate în zonele de intersecție.
- *frontal:* elevii prezintă o caracteristică, aceasta este analizată și plasată pe diagramă.

Reflecție și acțiune.

Cum vor fi grupați elevii?

- Exemplu. Elevii se grupează în trei grupe, fiecare grupă stabilește proprietățile unei structuri. Câte un reprezentant din fiecare grupă completează diagrama
- Care sunt întrebările care vor dirija evaluarea?
- Exemplu. Se poate adresa direct întrebarea asupra numărului de repetiții ale buclei sau se poate prezenta un exemplu de utilizare a structurii din care să se deducă numărul de repetiții (Enunțul „Citește poezia până când ai să o știi”, indică faptul că trebuie să citești cel puțin o dată poezia pentru ca să o știi.)

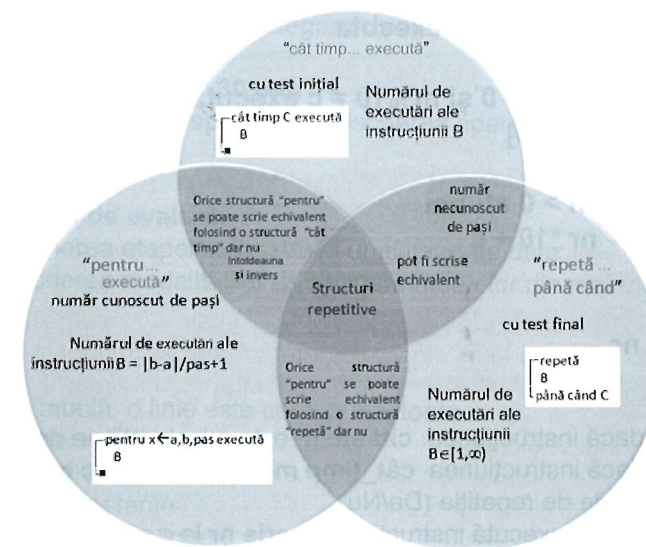
Barem de evaluare și de notare:

Fiecare grupă va primi o apreciere din partea profesorului/ colegilor.

se acordă 6p pentru completarea elementelor distincte ale structurilor (3x2p)

- se acordă 3p pentru completarea zonelor de intersecție a două cercuri (3x1p)
- se acordă un 1p pentru completarea zonei de intersecție a tuturor cercurilor

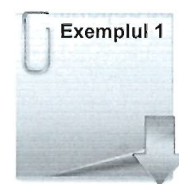
Un exemplu de completare a diagramei Venn:



2.2. EVALUARE SCRISĂ

Lucrările scrise reprezintă metoda (și chiar forma) fundamentală de evaluare a nivelului de pregătire a elevilor. Variante sub care se prezintă pot fi, de exemplu:

- testele sumative, probele de control de la sfârșitul unei teme complexe (unitate de învățare); acestea sunt anunțate din timp, se pot întinde pe durata unei ore, sunt direcționate prin sublinierea inițială a obiectivelor propuse;
- testele formative rezolvate pe parcursul uneia sau mai multor lecții;
- lucrări efectuate ca activitate independentă în clasă (efectuarea unui set de exerciții, rezolvarea sau compunerea unor probleme).



Exemplul 1

Itemi obiectivi

Itemi semiobiectivi (cu răspuns scurt, de completare, întrebări structurate)

Clasa: a IX-a

Timp de lucru: 50 min

Competență specifică. Aplicarea algoritmilor fundamentali pentru date simple

Competențe de evaluat:

Recunoașterea unor algoritmi care prelucreză cifrele unui număr, de divizibilitate, de calcul etc.

Adaptarea unor algoritmi dați pentru a răspunde unor cerințe suplimentare de prelucrare a datelor.

Enunț: Se consideră următorul algoritm pseudocod:

citește n (număr natural)

$nr \leftarrow 0$

pentru $c \leftarrow 9, 0, -1$ executa

$m \leftarrow n$

cât timp $m > 0$ și $m \% 10 \neq c$ execută

$m \leftarrow [m/10]$

dacă $m > 0$ atunci

$nr \leftarrow nr * 10 + c$

scrie nr

1. Stabiliți dacă instrucțiunea **citește n** este o instrucțiune de atribuire (Da/Nu)
 2. Stabiliți dacă instrucțiunea **cât timp $m > 0$ și $m \% 10 \neq c$ execută...** eface parte din structurile de repetiție (Da/Nu)
 3. De câte ori se execută instrucțiunea **scrie nr** la o rulare a programului ?
 - a) o dată
 - b) niciodată
 - c) de o infinitate de ori
 - d) depinde de valoarea lui n
 4. Expresia **($m > 0$ și $m \% 10 \neq c$)** este adevărată dacă
- Când această expresie devine falsă?
5. Dacă se citește valoarea **100**, pentru n stabiliți ce se va afișa prin execuția programului de mai sus.
 6. Ce se afișează dacă pentru n se citește valoarea 1132963?
 7. Care este semnificația valorii afișate în urma executării algoritmului?
 8. Scrieți două valori distincte care pot fi citite pentru variabila n astfel încât, pentru fiecare dintre acestea să se afișeze aceeași valoare în urma executării algoritmului.
 9. Dacă numărul n citit este format din cifre diferite de zero, scrieți semnificația valorii afișate în urma executării algoritmului obținut prin înlocuirea instrucțiunii.

pentru $c \leftarrow 9, 0, -1$ execută

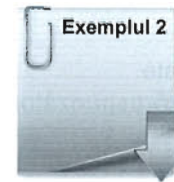
cu instrucțiunea

pentru $c \leftarrow 1, 9$ execută ...

10. Rescrieți algoritmul astfel încât prin înlocuirea instrucțiunii **pentru $c \leftarrow 9, 0, -1$ execută** cu o altă instrucțiune repetitivă condiționată anterior să obțineți un algoritm echivalent.

Barem de evaluare și notare:

Item	1	2	3	4	5	6	7	8	9	10	Total
Punctaj	0.5	0.5	0.5	2	0.5	1	2	1	1	1	10



Itemii obiectivi (cu alegere duală, multiplă)

Clasa: a XII-a

Timp de lucru: 30 min

Competență specifică. Descrierea operațiilor specifice bazelor de date

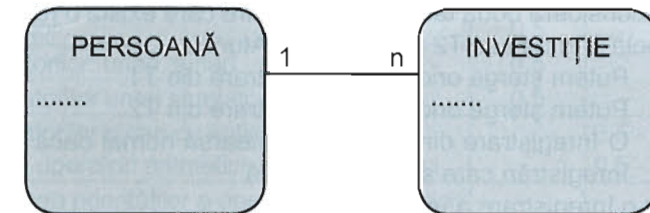
Competențe de evaluat:

Descrierea etapelor construirii unei baze de date

Descrierea modalităților de obținere a datelor din mai multe tabele legate

Enunț:

- 1) Într-o tabelă, o linie este corespunzătoare
 - a. unei relații
 - b. unui atribut
 - c. unei instanțe
 - d. unei constrângeri
- 2) Se mapează relația 1-n dintre entitățile **PERSOANA** și **INVESTITIE**.



Atunci

- a. cheia primară din tabela **PERSOANE** devine cheie primară în tabela **INVESTIȚII**;
 - b. cheia primară din tabela **PERSOANE** devine cheie străină în tabela **INVESTIȚII**;
 - c. cheia primară din tabela **INVESTIȚII** devine cheie primară în tabela **PERSOANE**;
 - d. cheia primară din tabela **INVESTIȚII** devine cheie străină în tabela **PERSOANE**;
- 3) La inițierea unei conexiuni, sursa de date la care se face conectarea trebuie să existe.
 - a. adevărat
 - b. fals
 - 4) Un fișier cu extensia implicită **.mdf** memorează o bază de date.
 - a. adevărat
 - b. fals
 - 5) Construirea unei aplicații legate la o bază de date presupune:
 - a. realizarea mai întâi a modelului conceptual și după aceea modelului fizic
 - b. realizarea modelului fizic și apoi a modelului conceptual
 - c. realizarea modelului fizic în paralel cu a modelului conceptual

- 6) Dintre următoarele afirmații despre relațiile stabilite între tabele este falsă:
- Sunt implementate cu ajutorul cheilor străine
 - Se stabilesc numai între două tabele diferite
 - Se pot defini numai între tabele aflate în aceeași bază de date
- 7) Este recomandat ca numele unei constrângeri să aibă sufixul corespunzător unei relații:
- PK_
 - FK_
 - PFK_
 - FP_
- 8) O tabelă poate avea mai multe chei secundare
- Adevărat
 - Fals
- 9) Care dintre următoarele afirmații referitoare la o cheie primară, este adevărată:
- O cheie primară trebuie să fie formată numai dintr-un singur câmp
 - Cheia primară realizează o indexare automată a înregistrărilor din tabelă
 - O tabelă poate avea mai multe chei primare
- 10) Dintre următoarele nu reprezintă tipuri de relații:
- 1-1 (one to one)
 - 0-n (zero to many)
 - n-n (many to many)
- 11) Se consideră două tabele T1 și T2 între care există o relație de tip 1-n (T1 este tabela părinte sau T2 tabela copil). Atunci:
- Putem șterge oricând o înregistrare din T1
 - Putem șterge oricând o înregistrare din T2
 - O înregistrare din T1 poate fi ștearsă numai dacă în tabela T2 nu există înregistrări care să-i corespundă
- 12) Într-o înregistrare a unei tabele
- orice câmp de tip numeric poate avea valoarea NULL
 - un câmp care face parte dintr-o cheie primară poate avea valoarea NULL
 - un câmp care face parte dintr-o cheie străină poate avea valoarea NULL
- 13) O constrângere corespunzătoare unei relații între două tabele se poate defini la nivelul
- tabelei părinte
 - tabelei copil
 - oricăreia dintre cele două tabele

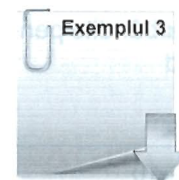
Barem de evaluare și notare:

1. – c.; 2. – b.; 3. – a.; 4. – a.; 5. – a.; 6. – b.; 7 – b;
8. – a.; 9. – b.; 10. – b.; 11. – c.; 12. – c.; 13. – b.;

Se acordă câte 5 puncte pentru fiecare dintre itemii cu un singur răspuns corect și câte 10 puncte pentru itemii cu două răspunsuri corecte.

Total 70 de puncte (12x5 + 1x10)

Din oficiu: 30 puncte



Exemplul 3 Itemi subiectivi (Eseu)

Clasa a IX-a

Timp de lucru: 50 min

Competență specifică. Identificarea și clasificarea datelor necesare pentru rezolvarea unei probleme

Competențe de evaluat: Identificarea datelor care intervin în evaluarea unei expresii aritmetice

Enunț:

Expresii. Descrieți ce reprezintă o expresie, relația dintre operanzi și operatori în reprezentarea pseudocod. Realizați o clasificare a operatorilor descriind și prioritățile.

Barem de evaluare și de notare:

Criterii și rezolvare	Punctaj acordat	Observații
1. definirea unei expresii, tipul și valoarea unei expresii	0.5	
2. definirea operandului: constantă, constantă simbolică, nume de variabilă, o altă expresie scrisă între paranteze	0.5	
3. definirea operatorilor	0.5	
4. clasificarea operatorilor: unari, binari	0.5	
5. prezentarea operatorilor unari studiați: +, -, negație	0.5	
6. prezentarea operatorilor binari cu aplicabilitate de la stânga la dreapta: operatori aritmetici, relaționali, logici	1	0.5 - corectitudine 0.5 - completitudine
7. descrierea în ordinea priorităților a operatorilor aritmetici: multiplicativi, aditivi; rezultatul obținut în urma aplicării operatorului aritmetic, restricții pentru operatorul „restul împărțirii întregi”	1	0.5 - corectitudine 0.5 - completitudine
8. descrierea operatorilor relaționali, rezultatul unei expresii ce conține operatori relaționali	1	
9. descrierea operatorilor logici, rezultatul expresiei ce conține operatori logici	1	
10. prezentarea priorităților operatorilor	0.5	
11. capacitatea de sinteză	1	
12. coerența prezentării, greșelile științifice nesancționate la alte secțiuni ale baremului vor fi penalizate aici	1	
13. punctaj acordat din oficiu	1	

Descriptori de performanță pe niveluri de realizare a sarcinii.

Profesorul trebuie să țină seama în acordarea notelor și de:

- volumul cunoștințelor însușite, raportat și la nivelul clasei respective;
- gradul de înțelegere a materiei, temeinicia cunoștințelor, priceperea de a aplica cunoștințele dobândite;
- numărul și natura erorilor cuprinse în cadrul răspunsurilor date;

➔ priceperea de a-și prezenta răspunsul într-o succesiune logică, prin expunerea cunoștințelor teoretice pe care le are, prin deprinderile de calcul dobândite pentru lecția curentă și deprinderile de calcul dobândite anterior.

nota 5-6	nota 7-8	nota 9-10
<ul style="list-style-type: none"> *identifică tipul și valoarea unei expresii *recunoaște operandul atunci când acesta este o constantă sau o variabilă *definește operatorii *descrie operatorii binari cu aplicabilitate de la stânga la dreapta: operatori aritmetici, relaționali 	<ul style="list-style-type: none"> *recunoaște operandul atunci când acesta este constantă, constantă simbolică, nume de variabilă, o altă expresie scrisă între paranteze *clasifică operatorii *descrie operatorii binari cu aplicabilitate de la stânga la dreapta: operatori aritmetici și relaționali, logici 	<ul style="list-style-type: none"> *clasifică operatorii în funcție de prioritate *descrie funcționalitatea operatorilor prin comparație și exemple concludente *identifică rolul operatorilor pe biți în expresii aritmetice *utilizează exemple care demonstrează capacitatea de sinteză



Itemi obiectivi cu alegere multiplă
Itemi semiobiectivi cu răspuns scurt
Itemi subiectivi (rezolvare de probleme)

Clasa: a XI-a

Timp de lucru: 50 min

Competențe specifice: Identificarea proprietăților grafurilor neorientate

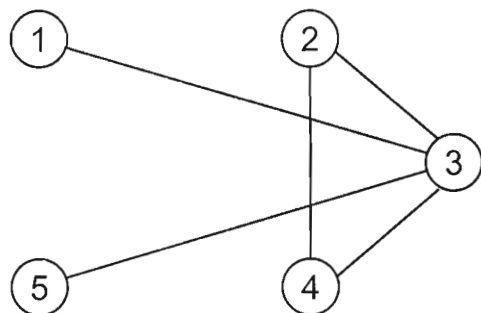
Competențe de evaluat:

Verificarea unor proprietăți ale grafurilor

Implementarea unor algoritmi specifici de prelucrare a grafurilor

Implementarea algoritmilor de verificare a unor proprietăți ale grafurilor

1. Se consideră graful din figura următoare.



- Enumerați nodurile de grad par.
- Care este numărul de muchii ce trebuie adăugate în graf pentru a obține un graf complet?
- Care este numărul de muchii ale celui mai lung lanț, format din noduri distincte, ce are ca extremități nodurile 1 și 4?
- Care este numărul minim de muchii ce pot fi eliminate adăugate în astfel încât graful parțial obținut să fie eulerian?

2. Care este numărul de grafuri neorientate, cu mulțimea nodurilor $\{1, 2, 3, \dots, n\}$? Justificați. Două grafuri sunt distincte dacă matricele lor de adiacență sunt diferite.

3. Care este numărul maxim de noduri de grad 3 într-un graf neorientat cu 5 noduri?

- a. 5 b. 4 c. 3 d. 2

4. Scrieți secvența de program care, pornind de la matricea de adiacență asociată unui graf neorientat cu n noduri:

a) afișează listele de adiacență;

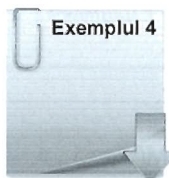
b) afișează toate ciclurile elementare de lungime 3.

Barem de evaluare și notare:

item	Răspuns corect	Punctaj	Observații
1a	2, 3, 4	1 p	
1b	5	1 p	$ M(K_5) = 5 \cdot 4 / 2 = 10$, $ M(G) = 5$
1c	3	1 p	lanțul 1-3-2-4
1d	1	1p	muchia [1, 5]
2	$2^{n(n-1)/2}$	2p	se acordă 1p pentru formulă corectă și 1p pentru demonstrație
3	b) 4	1p	
4a	pentru rezolvare corectă -parcursarea tuturor nodurilor -identificarea unui vecin al unui nod -afișarea listei de adiacență a unui nod	1p 0.25p 0.25p 0.50p	
4b	pentru rezolvare corectă - identificare ciclu elementar de lungime 3 (*) -afișarea tuturor ciclurilor elementare de lungime 3	1p 0.5p 0.5p	(*) Se acordă numai 0,25p dacă s-a identificat un ciclu, dar nu este elementar sau nu are lungimea 3
4b	Punctaj acordat din oficiu	1p	

Descriptori de performanță pe niveluri de realizare a sarcinii.

nota 5-6	nota 7-8	nota 9-10
<ul style="list-style-type: none"> * cunoaște noțiunile de graf neorientat, nod, muchie, graf complet, lanț * reprezintă graful cu ajutorul matricei de adiacență * identifică vecinii unui nod * calculează gradul unui nod 	<ul style="list-style-type: none"> * cunoaște noțiunile de graf parțial, subgraf, ciclu * calculează numărul de muchii pentru un graf neorientat cu n noduri * utilizează matricea de adiacență asociată unui graf 	<ul style="list-style-type: none"> * cunoaște noțiunile de graf eulerian, hamiltonian * calculează numărul de grafuri neorientate cu n noduri * determină ciclurile unui graf * modifică un graf astfel încât să fie îndeplinite condiții de maxim/minim



Exemplul 4

Tehnica 3-2-1

Tehnica 3-2-1 este folosită pentru a aprecia rezultatele unei secvențe didactice sau a unei activități. Denumirea provine din faptul că elevii scriu:

- **trei termeni** (concepte) din ceea ce au învățat,
- **doi idei** despre care ar dori să învețe mai mult în continuare și
- **o capacitate**, o pricepere sau o abilitate pe care consideră ei că au dobândit-o în urma activităților de predare-învățare.

Tehnica 3-2-1 poate fi considerată drept o bună **modalitate de autoevaluare** cu efecte formative în planul învățării realizate în clasă. Este o cale de a afla rapid și eficient care au fost efectele proceselor de predare și învățare, având valoare constatativă și de feed-back. Pe baza conexiunii inverse externe, profesorul poate regla procesele de predare viitoare, îmbunătățindu-le și poate elabora programe compensatorii, dacă rezultatele sunt sub așteptări ori programe în concordanță cu nevoile și așteptările elevilor.

Această modalitate complementară de evaluare are scopul de ameliorare și nicidecum de sancționare. **Este un instrument al evaluării continue**, formative, ale cărei funcții principale sunt de constatare și de sprijinire continuă a elevilor.

Clasa: a XI-a

Timp de lucru: 15 min

Competențe specifice: Descrierea operațiilor specifice structurilor arborescente

Competențe de evaluat:

Utilizarea corectă a termenilor specifici

Enunțarea principalelor proprietăți ale unei structuri arborescente

Enunț:

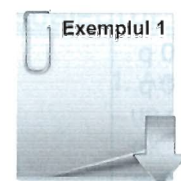
Referitor la *Structurile arborescente*, scrieți:

- * **trei concepte** învățate,
- * **două idei** despre care doriți să învățați mai mult în continuare și
- * **o capacitate**, o pricepere sau o abilitate pe care considerați că ați dobândit-o

Un răspuns posibil:

- * **trei concepte** învățate: arbore, arbore cu rădăcină, arbori binari
- * **două idei** despre care doresc să învăț mai mult în continuare: parcurgerea arborilor, utilizarea structurilor arborescente în aplicații practice
- * **o capacitate:** reprezentarea arborilor pe baza vectorului de tați

2.3. EVALUARE PRACTICĂ



Exemplul 1

Item subiectiv (rezolvare de probleme)

Clasa : a IX-a

Timp de lucru: 50 min

Competențe specifice: Aplicarea creativă a algoritmilor fundamentali în rezolvarea unor probleme concrete

Competențe de evaluat:

Reproducerea enunțului unei probleme concrete-practice

Evidențierea elementelor din enunț care sugerează un algoritm fundamental

Evidențierea elementelor din enunț care-l diferențiază de aplicarea directă a algoritmului sugerat

Adaptarea algoritmului pentru a răspunde cerințelor unei probleme noi

Implementarea algoritmului adaptat

Enunțul problemei:

Se citește de la tastatură un număr natural n . Scrieți un program care să determine toate numerele naturale formate din n cifre cu proprietatea că sunt numere prime și, de asemenea, numerele obținute prin oglindirea acestora sunt numere prime. Pentru valori ale lui n mari stabiliți un algoritm eficient de rezolvare a problemei.

Exemplu:

$n=2$, numerele prime, formate din 2 cifre sunt:

11, 13, 17, 31, 37, 71, 73, 79, 97

Managementul timpului. Barem de evaluare și notare.

Sub.	Enunț	Componente	Punctaj	Timp
1.	<ul style="list-style-type: none"> - înțelegerea enunțului problemei - stabilirea etapelor de rezolvare a problemei - specificarea tipului de date folosit - stabilirea datelor de intrare - stabilirea datelor de ieșire 	<ul style="list-style-type: none"> - citirea cu atenție a problemei - înțelegerea sarcinilor de lucru, aprofundarea exemplului - stabilirea valorilor pe care le poate lua variabila n astfel încât problema să poată avea soluție, știind că numerele naturale pot avea cel mult 9 cifre - stabilirea tipului de date long ca fiind tipul necesar pentru generalizarea problemei - data de intrare: n - date de ieșire: valorile numere prime pentru care și numărul obținut prin oglindire este număr prim 	0.25 p 0.25 p 0.25 p 0.25 p 0.25 p 0.25 p	5 min
2.	<ul style="list-style-type: none"> - stabilirea proprietății de număr prim - realizarea oglinzării unui număr natural 	<ul style="list-style-type: none"> - definiția numărului prim - stabilirea unei secvențe de algoritmi care în mod eficient va stabili dacă un număr este prim - determinarea unui număr pe baza cifrelor sale - obținerea cifrelor pentru construirea numărului natural obținut prin oglindire - copie de siguranță a numărului natural inițial, din cauza extragerii cifrelor sale 	0.50 p 0.50 p 0.50 p 0.25 p 0.25 p	15 min
3.	<ul style="list-style-type: none"> - stabilirea limitelor între care se verifică valorile dacă au proprietatea de număr prim 	<ul style="list-style-type: none"> - stabilirea numărului minim format din n cifre, ca fiind 10^{n-1} - stabilirea pașilor pentru obținerea acestui număr - stabilirea numărului maxim format din n cifre, ca fiind $10^n - 1$ - realizarea legăturii dintre numerele minim și maxim pentru optimizarea determinării valorii maxime 	0.50 p 0.50 p 0.25 p 0.25 p	10 min
4.	<ul style="list-style-type: none"> - optimizarea alegerii valorii ce trebuie verificată ca având proprietatea de număr prim 	<ul style="list-style-type: none"> - parcurgerea doar a valorilor impare - evitarea valorilor care au pe prima poziție o cifră pară și care prin oglindire nu pot fi numere prime 	0.50 p 0.50 p	5 min
5.	<ul style="list-style-type: none"> - verificarea corectitudinii algoritmului ales 	<ul style="list-style-type: none"> - realizarea algoritmului pseudocod - implementarea într-un limbaj de programare - verificarea algoritmului - verificare din punct de vedere a timpului de executare pentru date de intrare mari 	0.50 p 0.50 p 0.50 p 0.50 p	15 min

SUBIECTUL I

(1.5 puncte)

1. Citirea cu atenție a problemei. 0.25 p
2. Înțelegerea sarcinilor de lucru, aprofundarea exemplului. 0.25 p
3. Stabilirea valorilor pe care le poate lua variabila **n** astfel încât problema să poată avea soluție, știind că numerele naturale pot avea cel mult 9 cifre.
 repetă
 citește **n**
 până_când ($n \geq 1$ and $n \leq 9$) 0.25 p
 Stabilirea tipului de date **long** ca fiind tipul necesar pentru generalizarea problemei. 0.25 p
4. Data de intrare: **n**. 0.25 p
5. Date de ieșire: numere prime pentru care și numărul obținut prin oglindire este prim 0.25 p

SUBIECTUL al II-lea

(2 puncte)

1. Amintirea definiției numărului prim.
 - un număr cu exact doi divizori este număr prim. Un algoritm posibil, pentru $x > 1$:
 $ok \leftarrow 1$
 $d \leftarrow 2$
 cât_timp $d * d \leq x$ and $ok = 1$ execută
 dacă $x \% d = 0$ atunci $ok \leftarrow 0$
 altfel $d \leftarrow d + 1$
 sfârșit_cât_timp
 dacă $ok = 1$ atunci scrie **x „prim“**
 sfârșit_dacă 0.50 p
2. Stabilirea unei secvențe de algoritmi care în mod eficient va stabili dacă un număr este prim 0.50 p
3. Determinarea unui număr pe baza cifrelor sale 0.50 p
4. Obținerea cifrelor pentru construirea numărului natural obținut prin oglindire
 $y \leftarrow 0$
 $a \leftarrow x$
 cât_timp $a \neq 0$ execută
 $y \leftarrow y * 10 + a \% 10$
 $a \leftarrow [a / 10]$
 sfârșit_cât_timp 0.25 p
5. Copie de siguranță numărului natural inițial, din cauza generării cifrelor sale 0.25 p

SUBIECTUL al III-lea

(1.5 puncte)

1. Stabilirea numărului minim format din n cifre, ca fiind 10^{n-1} 0.50 p
 $\text{min} \leftarrow 1$
pentru $i \leftarrow 1, n-1$ **execută**
 $\text{min} \leftarrow \text{min} * 10$
sfârșit 0.50 p
3. Stabilirea numărului maxim format din n cifre, ca fiind $10^n - 1$ 0.25 p
4. Realizarea legăturii dintre numerele minim și maxim pentru optimizarea determinării valorii maxime 0.25 p
 $\text{max} \leftarrow \text{min} * 10 - 1$

SUBIECTUL al IV-lea

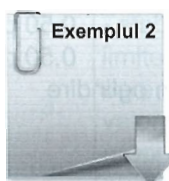
(1 punct)

1. Parcurgerea doar a valorilor impare
pentru $x \leftarrow \text{min} + 1, \text{max} 2$ **execută** 0.50 p
2. Evitarea valorilor care au pe prima poziție o cifră pară și care prin oglindire nu pot fi numere prime
dacă $x/\text{min}=2$ **or** $x/\text{min}=4$ **or** $x/\text{min}=6$ **or** $x/\text{min}=8$
atunci. 0.50 p

SUBIECTUL al V-lea

(2 puncte)

1. Realizarea algoritmului pseudocod 0.50 p
2. Implementarea într-un limbaj de programare 0.50 p
3. Verificarea algoritmului 0.50 p
4. Verificare din punct de vedere a timpului de executare pentru date de intrare mari 0.50 p



Exemplul 2

Itemi subiectivi

Clasa : a X-a/ a XI-a

Timp de lucru: 50 min

Competențe specifice: Identificarea metodei de programare adecvate pentru rezolvarea unei probleme
 Alegerea unui algoritm eficient de rezolvare a unei probleme

Competențe de evaluat:

Estimarea eficienței, ca timp de executare, a unei rezolvări

Modificarea unui algoritm pentru a îndeplini o anumită condiție de eficiență (număr de variabile, număr de structuri repetitive, număr de executări ale unui pas etc.)

Enunț: Scrieți un program care citește de la tastatură un număr natural n ($100 \leq n \leq 1000$) și în continuare un șir de n numere naturale nenule cu cel mult 5 cifre. Programul construiește în memorie un tablou unidimensional care să conțină șirul de numere și elimină din tablou toate aparițiile ultimului termen al șirului. Se cere utilizarea unui algoritm care să conțină cât mai puține structuri repetitive.

Exemplu. Pentru $n = 10$ și numerele: 2, 8, 4, 8, 8, 3, 8, 8, 9, 8 se obține șirul 2, 4, 3, 9.

Pentru program corect se acordă punctaj maxim	10 p.	*datele de intrare se consideră corecte, nefiind necesară validarea lor
x declararea corectă a variabilelor	1p.	
x citire corectă, afișare corectă *	1p.	
x eliminarea unui element din tablou	1p.	
x eliminarea tuturor aparițiilor ultimului element din șir	2p.	
x utilizarea unui algoritm eficient din punct de vedere al timpului de executare**	3p.	
x corectitudinea globală a programului	1p.	
x punctaj acordat din oficiu	1p.	

**Se acordă cele 3 puncte dacă secvența de eliminare evită deplasările repetate ale elementelor tabloului.

O soluție posibilă:

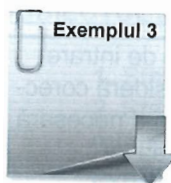
```

citește n
pentru  $i \leftarrow 1, n$  execută
    citește  $v[i]$ 
sfârșit
 $x \leftarrow v[n]$ 
 $n \leftarrow n - 1$ 
 $p \leftarrow 0$ 
pentru  $i \leftarrow 1, n$  execută
    dacă  $x \neq v[i]$  atunci
         $p \leftarrow p + 1$ 
         $v[p] \leftarrow v[i]$ 
        scrie  $v[p]$ 
    sfârșit
 $n \leftarrow p$ 

```

Descriptori de performanță pe niveluri de realizare a sarcinii.

nota 5-6	nota 7-8	nota 9-10
<ul style="list-style-type: none"> x declară variabile de tip întreg și tablou unidimensional x citește datele de intrare x afișează datele de ieșire x identifică pozițiile pe care se află o valoare 	<ul style="list-style-type: none"> x elimină un element din tablou x elimină toate aparițiile unui element într-un tablou 	<ul style="list-style-type: none"> x utilizează un algoritm eficient pentru eliminarea unei valori din tablou

**Exemplul 3****Itemii subiectivi****Clasa : a X-a / a XI-a****Timp de lucru: 50 min****Competențe specifice:**

Aplicarea creativă a algoritmilor fundamentali în rezolvarea unor probleme concrete.

Elaborarea unor algoritmi de prelucrare a datelor structurate

Competențe de evaluat:

Utilizarea corectă a parametrilor subprogramelor predefinite

Utilizarea funcțiilor și procedurilor specifice de prelucrare a șirurilor de caractere

Apelarea corectă a subprogramelor predefinite pentru șiruri de caractere

Aplicarea unor metode de prelucrare pe seturi de date concrete

Construirea algoritmilor care să efectueze operații tipice asupra structurilor de date

Enunț:

Scrieți un program care citește de la tastatură un text format din cel mult 200 de caractere, litere, cifre și spații și realizează, în ordine, următoarele prelucrări:

a) șterge spațiile inutile din text, astfel încât două cuvinte să fie separate printr-un singur spațiu;

b) înlocuiește fiecare secvență maximală de cifre aflate pe poziții consecutive în șir, care formează codul ASCII al unei litere cu litera ce are codul ASCII astfel determinat;

c) afișează pe ecran numărul de subșiruri formate numai din litere;

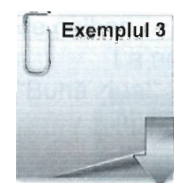
d) șterge din șir toate cuvintele care au cel puțin un caracter care nu este literă.

Exemplu: dacă de la tastatură se citește șirul:**97st97zi este m97i m789i 98in102** se obțin rezultatele:**a) 97st97zi este m97i m789i 98in102****b) astazi este mai m789i bine****c) 4****d) astazi este mai bine****Barem de evaluare și notare :**

Pentru program corect se acordă punctaj maxim	10 p.	*datele de intrare se consideră corecte, nefiind necesară validarea lor
x declararea corectă a variabilelor	1p.	
x citire corectă, afișare corectă *	1p.	
x eliminarea spațiilor inutile din text	1p.	
x înlocuirea secvențelor ce reprezintă coduri ASCII	2p.	
x determinarea subșirurilor formate numai din litere	1p.	
x ștergerea cuvintelor care au cel puțin un caracter diferit de literă	1p.	
x corectitudinea globală a programului	1p.	
x punctaj acordat din oficiu	1p.	

Descriptori de performanță pe niveluri de realizare a sarcinii.

nota 5-6	nota 7-8	nota 9-10
<ul style="list-style-type: none"> x declară variabile de tip întreg și șir de caractere x citește datele de intrare x afișează șiruri de caractere x identifică tipul unui caracter (literă, cifră, spațiu) 	<ul style="list-style-type: none"> x șterge un caracter din șir x elimină spațiile inutile din text x șterge un subșir dintr-un șir x numără subșiruri formate din litere 	<ul style="list-style-type: none"> x extrage un subșir maximal format doar din cifre x înlocuiește un subșir cu alt subșir

**Exemplul 3****Itemii subiectivi****Clasa : a XII-a (Microsoft SQL Server)****Timp de lucru: 30 min****Competențe specifice:** Descrierea operațiilor specifice bazelor de date**Competențe de evaluat:**

Descrierea etapelor construirii unei baze de date

Aplicarea metodelor de creare și actualizare a unei tabele

Enunț:Să se proiecteze baza de date **ORGANIZATOR** pentru gestionarea activității unei firme care organizează evenimente.

Interesează informații referitoare la persoane care au fost, sunt sau vor fi clienții firmei. Pentru fiecare astfel de persoană memorăm numele, prenumele, adresa și numărul de telefon.

De asemenea, interesează detalii referitoare la ornamentele care pot fi alese de un client pentru buna organizare a unui eveniment. Fiecare ornament are o denumire, o scurtă descriere, un cod și un preț.

Fiecare comandă este dată de un client, pentru un singur eveniment. Se memorează data comenzii, locul de desfășurare și o scurtă descriere a evenimentului.

Un client poate da mai multe comenzi. Baza de date trebuie să poată evidenția toate ornamentele utilizate în organizarea unui eveniment și toate evenimentele la care a fost folosit un ornament.

a) Să se construiască modelul conceptual atașat bazei de date ORGANIZATOR.

b) Să se implementeze pe baza de date, proiectând tabelele aceia

Barem de evaluare și notare:

a) construirea modelului conceptual

- stabilirea unor entități care să descrie clienții, ornamentele, detaliile comenzilor
- stabilirea relațiilor corecte între entități
- stabilirea corectă a cardinalității relațiilor
- posibilitatea obținerii tuturor informațiilor cerute pe baza datelor descrise în modelul ales
- elemente de creativitate, originalitate

3p

2p

2p

2p

1p

b) construirea tabelor bazei de date

- stabilirea numelui bazei de date a conexiunii
- crearea tabelor
- definirea câmpurilor fiecărei table
- relaționarea tabelor pe baza cheilor străine

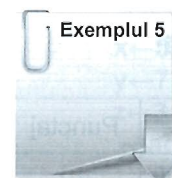
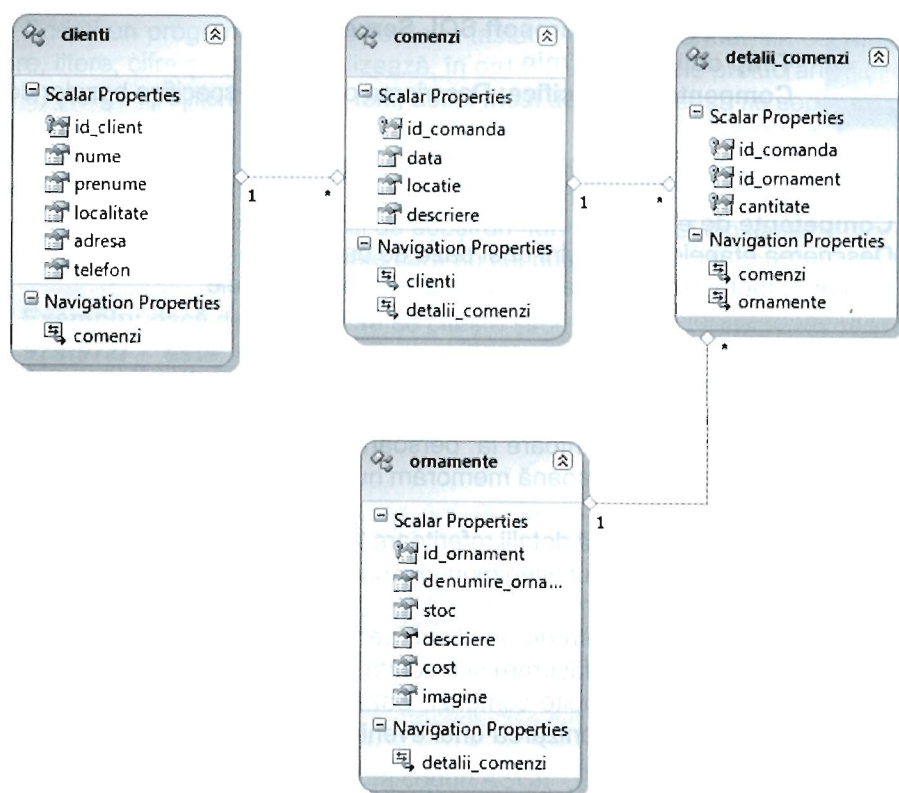
1p

3p

3p

3p

O soluție posibilă este modelul:



Itemi subiectivi

Clasa a XII-a (Visual C#)

Timp de lucru: 50 min

Competențe specifice. Elaborarea strategiei de realizare a unei aplicații

Competențe de evaluat:

Recunoașterea proprietăților specifice Form-ului și uneltei Button în Visual C#. Utilizarea corectă a funcției Show din clasa MessageBox

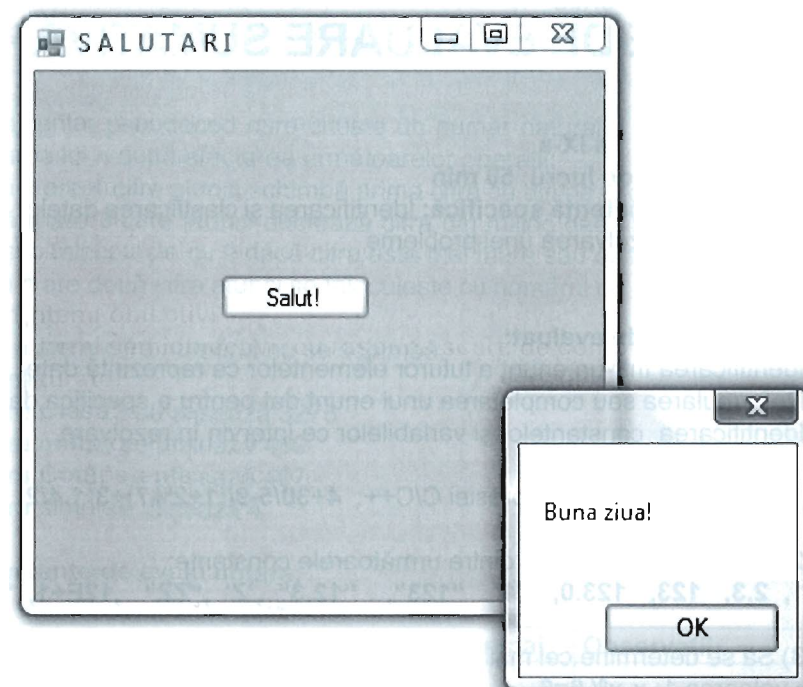
Enunț:

Creați o aplicație Windows cu numele Salutări care să conțină fereastra, cu titlu "SALUTĂRI" pe care se află un buton, cu eticheta "Salut!".

Culoarea fundalului ferestrei este nuanțată de albastru, iar a butonului e nuanță de galben.

La acționarea butonului se deschide o fereastră de mesaje care să conțină textul "Bună ziua!"

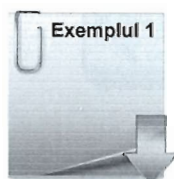
Aplicația poate avea aspectul următor:



Barem d evaluare și de notare pentru aplicația practică realizată:

Descriere, soluție posibilă	Punctaj
- accesare meniu File New Project Windows Forms Application , inserare nume proiect „Salutari”	1p
- în fereastra Properties, se alege proprietatea Text în care se șterge „Form1” și se introduce „S A L U T A R I”	1p
- în fereastra Properties se selectează culoarea dorită de la proprietatea BackColor	1p
- din fereastra Toolbox se inserează un „Button” prin dublu clic de mouse sau drag and drop	1p
- în fereastra Properties corespunzătoare butonului creat, se selectează culoarea dorită de la proprietatea BackColor	1p
- în fereastra Properties corespunzătoare butonului creat, se selectează proprietatea Text în care se înlocuiește „button1” cu „Salut!”	1p
- prin dublu clic de mouse pe butonul creat se completează funcția corespunzătoare MessageBox.Show(„Buna ziua!”);	1p

3. EXEMPLE DE EVALUARE SUMATIVĂ



Exemplul 1

Clasa : a IX-a

Timp de lucru: 50 min

Competență specifică: Identificarea și clasificarea datelor necesare pentru rezolvarea unei probleme

Competențe de evaluat:

Identificarea într-un enunț a tuturor elementelor ce reprezintă date

Reformularea sau completarea unui enunț dat pentru a specifica datele implicate

Identificarea constantelor și variabilelor ce intervin în rezolvare

1) Calculați valoarea expresiei C/C++: $4+30/5-9/(1+2\%7)+3*1.4/2$

2) Precizați tipul fiecărei dintre următoarele constante:

1, 2.3, 123, 123.0, '1', "123", "12.3", 'Z', "ZZ", 12E+1, "13E2"

3) Să se determine cel mai mic număr natural x cu proprietatea că expresia următoare are valoarea 1: $x-x\%6=6$

4) Care este numărul de atribuiri efectuate la executarea algoritmului:

```

x ← 5
y ← 7
z ← 3
dacă (y-z)%2=0 atunci
    y ← 4
    dacă x+y>3 atunci
        z ← x+y
    altfel
        x ← 2
        y ← 1

```

5) Se consideră algoritmul alăturat, descris în pseudocod, în care numărul citit pentru k este format dintr-o singură cifră.

a) Scrieți numărul care se va afișa dacă se citește pentru n valoarea 526791 și pentru k valoarea 6.

b) Dacă pentru n se citește valoarea 76895 scrieți toate valorile ce pot fi citite pentru k astfel încât valoarea afișată să fie 0.

c) Scrieți un enunț scurt pentru problema rezolvată de acest algoritm.

d) Scrieți programul Pascal/C/C++ corespunzător algoritmului

```

x ← 0
citește n, k
(numere naturale nenule)
cât timp n ≠ 0 execută
    dacă n%10 < k atunci
        x ← x*10+n%10
    n ← [n/10]
scrie x

```

6) Scrieți un algoritm pseudocod care citește un număr natural n cu maximum 4 cifre și afișează valoarea lui n după efectuarea următoarelor operații:

- dacă n are 4 cifre atunci schimbă prima cifră cu ultima și a doua cifră cu a treia

- dacă n are 3 cifre dublează cifra din mijloc dacă aceasta este strict mai mică decât 5 și o înlocuiește cu 0 dacă cifra este mai mare sau egală cu 5

- dacă n are două cifre atunci se înlocuiește cu numărul de numere divizibile cu 10 din intervalul $[1, n]$;

- dacă n are o cifră numărul nu se schimbă.

Exemple.

- pentru $n=6723$ se afișează 3276

- pentru $n=439$ se afișează 469

- pentru $n=487$ se afișează 407

- pentru $n=41$ se afișează 4

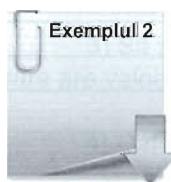
Barem de evaluare și notare:

Item	Răspuns corect	punctaj	Observații
1.	9.1	1	
2.	intregi: 1, 123 reale: 2.3, 123.0, 1.2E+1 caracter: '1', 'Z' sir de caractere: "123", "12.3", "ZZ", "13E2"	0.25 0.25 0.25 0.25	
3.	6	1	

4.	5	1	
5.	a. 125	1	se obține numărul format cu cifrele mai mici decât k (în ordinea inversă apariției începând cu cifra unităților)
	b. $k \in \{1,2,3,4,5\}$	1	condiție necesară : cifrele din n să fie mai mari sau cel mult egale cu k se acordă 0.20 pentru fiecare valoare corectă
	c. O soluție posibilă: se citește un număr natural n și o cifră nenulă k. Afișați, numărul format cu cifrele numărului n care sunt strict mai mici decât k, în ordine inversă apariției lor în n	1	
	d. declarare variabile	0.20	
	structură repetitivă	0.20	
	structură alternativă	0.20	
	citire corectă	0.10	
	scriere corectă	0.10	
	corectitudine sintactică	0.20	
6.	citire, scriere date	0.20	
	rezolvare corectă pentru număr cu 4 cifre	0.20	
	rezolvare corectă pentru număr cu 3 cifre	0.20	
	rezolvare corectă pentru număr cu 2 cifre	0.20	
	corectitudine sintactică	0.20	

Matricea de specificații:

Niveluri cognitive/ Conținuturi	Achiziția informației	Înțelegere	Aplicare	Analiză	Total
Tipuri de date	5%	5%	-	-	10%
Operatori aritmetici	10%	10%	10%	10%	40%
Structura alternativă	5%	5%	5%	5%	20%
Structura repetitivă	5%	5%	10%	10%	30%
Pondere %	25%	25%	25%	25%	100%



Clasa a X-a / clasa a XI-a

Timp de lucru: 50 min

Competențe specifice: Utilizarea subprogramelor

Competențe de evaluat:

Identificarea datelor de intrare, a datelor de ieșire și a celor de intrare-ieșire într-o prelucrare modulară

Diferențierea definirii unui subprogram de apelul aceluia subprogram

Diferențierea unui subprogram procedural de unul funcțional

Identificarea variabilelor locale, variabilelor globale și a parametrilor

- Se consideră subprogramul **f**, definit în continuare. Care este numărul valorilor întregi, distincte pe care le poate avea parametrul **x** astfel încât **f(x)=1**?

Pascal

```
function f(x:integer):integer;
begin
  f:=x div 2 div 3;
end;
```

C/C++

```
int f(int x)
{ return x/2/3; }
```

- a. 1 b. 2 c. 4 d. 6

- Pentru câte valori ale lui **n** din intervalul **[1,100]** apelul **p(n)** determină afișarea valorii 9?

Pascal

```
procedure p(n:integer);
var s: integer;
begin
  s:=n div 10+n mod 10;
  n:=s div 10+s mod 10;
  write(n);
end;
```

C/C++

```
void p(int n)
{ int s=n/10+n%10;
  n=s/10+s%10;
  cout<<n; | printf(„%d“, n);
}
```

- a. 1 b. 10 c. 11 d. 99

- Se consideră subprogramul **test**, definit în continuare. Ce rezultat se reține în variabila **k** dacă **a=207** și **b=23**?

Pascal

```
function test(a,b:integer):integer;
var k:integer;
begin
  k:=0;
  repeat
    if a mod 2 <>0 then k:=k+b;
    a:=a div 2;
    b:=b*2;
  until a=1;
  k:=k+b;
  test:=k;
end;
```

C/C++

```
int test(int a, int b)
{ int k=0;
  do
  { if(a%2!=0) k=k+b;
    a=a/2;
    b=b*2;
  }while(a!=1);
  k=k+b;
  return k;
}
```

- a. 23 b. 207 c. 4761 d. 5421

4. Următorii cinci itemi se referă subprogramul următor :

Pascal
function
f(n:longint;x:integer):longint;
var m:longint;
begin
m:=0;
while n>0 do
begin if n mod 10<>x then
m:=m*10+n mod 10;
n:=n div 10
end;
f:=m
end;

C/C++
long f(long n, int x)
{ long m=0;
while(n>0)
{if(n%10!=x) m=m*10+n%10;
n=n/10;}
return m;
}

5. Ce valoare are **f(52324,2)**?

6. Care este cel mai mic număr natural cu exact trei cifre ce poate fi atribuit parametrului **n** astfel încât **f(n,2)=0** ?

7. Dacă **n** este un număr natural cu exact 5 cifre iar **x** este o cifră, pentru câte perechi distincte **(n,x)** valoarea **f(n,x)** este 2009?

8. Care este cel mai mare număr natural **n**, cu exact trei cifre, pentru care **f(n,7)=f(n,9)**?

9. Scrieți numărul de valori naturale distincte cu cel mult două cifre pe care le poate primi variabila **n** astfel încât **f(n,n)=n**.

Barem de evaluare și notare

item	1	2	3	4a	4b	4c	4d	4e	of
raspuns	d	c	c	435	200	42	888	10	
punctaj	1	1	1	1	1	2	1	1	1

Observații

4b) Se acordă 0.5p dacă valoarea verifică relația dată dar nu este cea mai mică

4c) $n \in \{9002, 90002, 900020, 9000200\}$ pentru $x \in \{1, 3, 4, 5, 6, 7, 8\}$

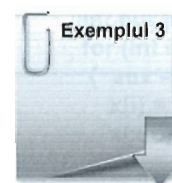
și $n=90020$ pentru $x \in \{1, 3, 4, 5, 6, 7, 8\}$

4d) Se acordă 0.5p dacă valoarea verifică relația dată dar nu este cea mai mare

4e) $n \in \{0, 11, 22, 33, 44, 55, 66, 77, 88, 99\}$

Matricea de specificații

Obiective/Niveluri cognitive	Achiziția informației	Înțelegere	Aplicare	Analiză	Total
Identificarea datelor de intrare, a datelor de ieșire și a celor de intrare-ieșire într-o prelucrare modulară	10%	10%	10%	-	30%
Diferențierea definirii unui subprogram de apelul acelui subprogram	10%	10%	-	-	20%
Diferențierea unui subprogram procedural de unul funcțional	10%	-	-	-	10%
Identificarea variabilelor locale, variabilelor globale și a parametrilor	10%	10%	10%	10%	40%
Pondere %	40%	30%	20%	10 %	100%



Clasa a X-a / clasa a XI-a

Timp de lucru: 100 min

Competențe specifice. Elaborarea și implementarea rezolvărilor de probleme din aria curriculară a specializării

Competențe de evaluat:

Respectarea etapelor de rezolvare (modelare, implementare, testare, depanare)

Justificarea alegerii unei metode de rezolvare pentru o problemă

Estimarea eficienței metodei de rezolvare alese

Enumerarea tehnicilor folosite în implemetarea rezolvării

Enunț: Să se realizeze un proiect cu tema:

„Operații matematice cu numere naturale având mai mult de 10 cifre fiecare”.

Etapele realizării proiectului:

Nr crt	Etapele realizării proiectului	Modalitatea de realizare	Observații
1.	Înțelegerea problemei	<ul style="list-style-type: none"> - numerele conțin până la 1000 de cifre; ele se vor găsi într-un fișier de date scrise pe câte o linie, între cifre neexistând niciun separator. - cifrele numerelor se vor reține în doi vectori, numărul de componente ale vectorului constituind lungimea numărului; prelucrarea cifrelor se realizează transformând caracterul cifră în valoare numerică 	- fișierul va fi redactat corect, după fiecare număr va fi caracterul de trecere la linie nouă
2.	proiectarea unei soluții	<ul style="list-style-type: none"> - adunarea - se determină suma folosind algoritmul matematic însumând cifră cu cifră și pornind de la cifra unităților - se realizează un nou vector care reține rezultatul - scăderea - se folosește tot algoritmul matematic având în vedere ca diferența cifrelor corespunzătoare să se poată efectua - se va realiza și un subprogram care să determine care din cele două numere este mai mare pentru a efectua diferența - înmulțirea - se are în vedere că înmulțirea reprezintă realizarea unui vector obținut pas cu pas prin înmulțirea primului număr cu câte o cifră a celui de-al doilea număr 	<ul style="list-style-type: none"> - se pun în discuție cazuri de diferențe: 100078 - 76296v - se consideră cazul în care $a-b \geq 0$

Nr crt	Etapele realizării proiectului	Modalitatea de realizare	Observații
3.	realizarea programului	- se are în vedere declararea variabilelor globale - rezultatele fiind scrise în fișier este suficient să fie folosit doar un vector pentru a reține pe rând suma, diferența și respectiv produsul	
4.	testarea programului	- se vor da exemple de numere astfel încât să se realizeze o verificare în cât mai multe situații	
5.	elaborarea documentației	- se va realiza un manual scris atașat care va aduce explicații subprogramelor folosite	

În realizarea proiectului, obiectivele urmărite sunt legate de cunoașterea din partea elevilor a:

- operațiilor cu fișiere
- prelucrarea caracterelor și folosirea codului ASCII pentru a determina valoarea numerică asociată
- inversarea elementelor unui vector pentru o parcurgere mai ușoară a elementelor în vederea efectuării operațiilor
- protecția, în cazul diferenței, la rezultatele obținute

Proiectul poate fi reluat în clasele terminale pentru a i se asocia o interfață mai frumoasă, într-un mediu vizual.

Model pentru implementare în limbajul C++:	Model pentru implementarea în limbajul Pascal:
<pre>#include<fstream.h> #define Fin „Op_nr.in“ #define Fout „Op_nr.out“ #define Max 1000 int a[Max], b[Max]; ifstream in(Fin); ofstream out(Fout); void read(int x[]) { char ch; int k=0; in.get(ch); while(ch!='\n') { x[++k]=ch-'0'; in.get(ch); } x[0]=k; } void invers(int x[])</pre>	<pre>Program Numere_Mari; Const Fin = 'Op_nr.in'; Const Fout = 'Op_nr.out'; Const Max = 1000; Type Vector = Array[0..2*Max] Of Integer; Var a, b, r : Vector; f, g : Text; er, i, j, k, v : Integer; procedure citire(var x: Vector); var ch: char; begin Assign(f, Fin); Reset(f); k:=0; while Not Eoln(f) Do begin Read(f, ch); inc(k); val(ch,x[k],er); end; x[0]:=k; ReadLn(f); end;</pre>

<pre>{ int aux; for (int i=1; i<=x[0]/2; i++) { aux = x[i]; x[i] = x[x[0]-i+1]; x[x[0]-i+1] = aux; } } void write(int x[]) { for(int i=x[0]; i>=1; i--) out<<x[i]; out<<endl; } void suma(int x[]) { int ct = 0, i; if(a[0]>b[0]) x[0]=a[0]; else x[0]=b[0]; for(i=1; i<=x[0]; i++) { x[i] = a[i] + b[i] + ct; ct = x[i]/10; x[i] = x[i]%10; } if(ct) { x[0]++; x[x[0]] = ct; } } int nr_mare() { if(a[0]>b[0]) return 1; else if(a[0]<b[0]) return -1; else { int i=a[0]; while(i>=1 && a[i]==b[i]) i--; if(i<1) return 0; else if(a[i]>b[i]) return 1; else return -1; } } void diferenta(int a[],int b[],int x[]) { int i, j, k; int d[Max]; for(i=0; i<=a[0]; i++) { d[i]=a[i]; x[i]=0; } x[0]=d[0]; for(i=1; i<=x[0]; i++) { if(d[i]>=b[i]) x[i]=d[i]-b[i]; else { j=i+1; while(d[j]==0) j++; d[j]--; } } }</pre>	<pre>procedure invers(var x: Vector); var aux : Integer; begin for i:=1 To x[0] div 2 Do begin aux := x[i]; x[i] := x[x[0]-i+1]; x[x[0]-i+1] := aux; end end; procedure afis(var x: Vector); begin for i:=x[0] DownTo 1 Do write(g, x[i]); writeln(g); end; procedure suma(var x: Vector); Var ct : Integer; begin ct := 0; if(a[0]>b[0]) Then x[0] := a[0] else x[0] := b[0]; for i:=1 To x[0] Do begin x[i] := a[i] + b[i] + ct; ct := x[i] Div 10; x[i] := x[i] Mod 10; end; if ct<>0 Then begin inc(x[0]); x[x[0]] := ct; end; end; function nr_mare: Integer; begin if (a[0]>b[0]) Then nr_mare := 1 Else if(a[0]<b[0]) Then nr_mare := -1 Else begin i := a[0]; while (i>=1) and (a[i]=b[i]) Do Dec(i); if(i<1) Then nr_mare := 0 Else if (a[i]>b[i]) Then nr_mare := 1 Else nr_mare := -1; end; end; procedure diferenta(a, b : Vector; var x : Vector); var d : vector; begin for i:=0 To a[0] Do begin d[i]:=a[i]; x[i]:=0; end;</pre>
--	--

```

for(k=i+1; k<j; k++) d[k]=9;
x[i]=10+d[i]-b[i];
}

while(x[x[0]]==0) x[0]--;
}
void produs(int x[])
{
int cp, cs, i, j, pr;
for(i=0; i<2*Max; i++) x[i]=0;
for(i=1; i<=b[0]; i++)
{
cp=cs=0;
for(j=1; j<=a[0]; j++)
{
pr = a[j]*b[i] + cp;
cp = pr/10;
pr = pr%10;
x[i+j-1] = x[i+j-1] + pr + cs;
cs = x[i+j-1]/10;
x[i+j-1] = x[i+j-1]%10;
}
if(cp+cs) x[i+j-1] = cs+cp;
}
x[0]=2*Max-1;
while(x[x[0]]==0) x[0]--;
}

int main()
{
int r[2*Max];
read(a); read(b);
invers(a); invers(b);
suma(r);
write(r);
int v=nr_mare();
if(v==0) out<<0<<endl;
else
{ if(v==1) diferenta(a, b, r);
else diferenta(b, a, r);
write(r);
}
produs(r);
write(r);out.close();
return 0;
}

```

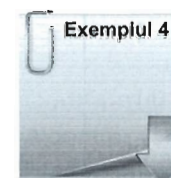
```

for i:=1 To x[0] Do
if (d[i]>=b[i]) Then
x[i]:= d[i]-b[i]
Else
begin j:=i+1;
while (d[j]=0) Do Inc(j);
Dec(d[j]);
for k:=i+1 To j-1 Do d[k]:=9;
x[i]:= 10 + d[i] - b[i];
end;
while x[x[0]]=0 Do Dec(x[0]);
end;
procedure produs(var x : Vector);
var cp, cs, pr : Integer;
begin
for i:=0 To 2*Max Do x[i]:=0;
for i:=1 To b[0] Do
begin
cp:=0; cs:=0;
for j:=1 To a[0] Do
begin
pr := a[j]*b[i] + cp;
cp := pr Div 10;
pr := pr Mod 10;
x[i+j-1] := x[i+j-1] + pr + cs;
cs := x[i+j-1] Div 10;
x[i+j-1] := x[i+j-1] Mod 10;
end;
if cp+cs<>0 Then x[i+j] := cs+cp;
end;
x[0]:=i+j+2;
while x[x[0]]=0 Do Dec(x[0]);
end;
begin
Assign(g, Fout);Rewrite(g);
citire(a); citire(b);
invers(a); invers(b);
suma(r); afis(r);
v := nr_mare;
if(v=0) Then writeln(g, 0)
Else begin
if(v=1) Then diferenta(a, b, r)
else diferenta(b, a, r);
afis(r);
end;
produs(r);afis(r);close(g);
end.

```

Matricea de specificații:

Niveluri cognitive / Conținuturi	Achiziția informației	Înțelegere	Aplicare	Analiză	Total
Înțelegerea problemei	10%	5%	-	-	15%
Proiectarea unei soluții	10%	5%	10%	10%	35%
Realizarea programului	5%	5%	5%	5%	20%
Testarea programului și elaborarea documentației	5%	5%	10%	10%	30%
Pondere %	30%	20%	25%	25%	100%



Exemplul 4

Clasa a XII-a

Timp de lucru: 50 min

Competențe specifice. Elaborarea strategiei de realizare a unei aplicații

Competențe de evaluat:

Recunoașterea proprietăților specifice Form-ului, TextBox-ului, ListBox-ului și uneltei Button în Visual C#.

Alegerea și aplicarea corectă a funcțiilor corespunzătoare clasei ListBox.

Enunț:

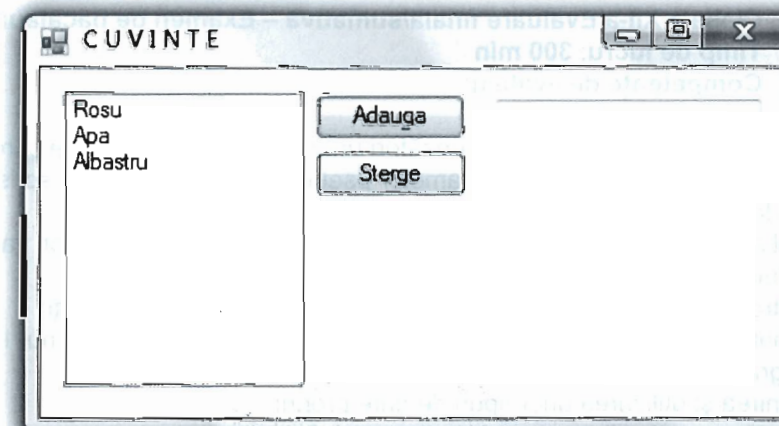
Creați o aplicație Windows cu numele Cuvinte, care să conțină o fereastră cu titlul "CUVINTE" și fundal de o nuanță de galben.

Fereastra cuprinde o listă care inițial conține trei cuvinte. Conținutul listei poate fi modificat la acționarea a unuia dintre cele două butoane aflate pe fereastra, cu etichetele "Sterge" și respectiv "Adaugă":

Cuvântul care se va adăuga în listă este editat într-o regiune de tip adecvat și va fi șters din aceasta după includerea în listă.

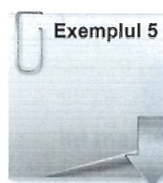
Cuvântul care se șterge din listă este selectat cu butonul mouse-ului.

Aplicația poate avea aspectul următor:



Barem de evaluare și de notare pentru aplicația practică realizată:

Cerință	Descriere, soluție posibilă	Punctaj	Obs.
1.	- accesare meniu File New Project Windows Forms Application , inserare nume proiect „Cuvinte”	1	
2.	- în fereastra Properties, se alege proprietatea Text în care se șterge „Form1” și se introduce „C U V I N T E”; - se selectează culoarea dorită de la proprietatea BackColor	1	
3.	- din fereastra Toolbox se inserează două obiecte de tip „Button”, un ListBox și un TextBox	1	
4.	- se accesează ListBox Tasks pentru a adăuga itemi listei	2	
5.	- din fereastra Properties corespunzătoare primului sau celui de-al doilea buton se actualizează valorile proprietăților.	1	
6.	a. private void btnAdauga_Click(object sender, EventArgs e) { listBox1.Items.Add(textBox1.Text); textBox1.Text = „”; }	1	
	b. private void btnSterge_Click(object sender, EventArgs e) { listBox1.Items.Remove(listBox1.SelectedItem); }	1	
7.	- funcționalitatea butoanelor create	1	



Clasa a XII-a Evaluare finală/sumativă – Examen de bacalaureat

Timp de lucru: 300 min

Competențe de evaluat:

- construirea algoritmilor corespunzători unor prelucrări elementare și reprezentarea lor prin intermediul programelor pseudocod și programelor scrise în limbaj de programare (Pascal sau C/C++, la alegere);
- analiza rezolvării unei probleme prin urmărirea evoluției valorilor variabilelor prelucrate de algoritmul corespunzător;
- abstractizarea rezolvării prin construirea unor algoritmi echivalenți;
- identificarea și utilizarea tipurilor de date predefinite specifice unui limbaj de programare;
- definirea și utilizarea unor tipuri de date proprii;
- identificarea și utilizarea operatorilor predefiniți elementari;
- identificarea și utilizarea subprogramelor predefinite elementare;
- identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;
- definirea și apelul unor subprograme proprii cu înțelegerea mecanismelor de transfer prin intermediul parametrilor;

- identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;
- organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate;
- organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control și module de program;
- folosirea unor metode sistematice de rezolvare pentru probleme de generare;
- analiza unor algoritmi echivalenți de rezolvare a unei probleme în vederea alegerii algoritmului optim.

Întrucât proba scrisă este structurată pe trei subiecte și un număr de 15 itemi, s-a realizat o distribuie a conținuturilor prevăzute în programa de bacalaureat, pe subiecte, astfel:

	Subiectul	Subiect I	Subiect II	Subiect III
Conținutul				
Algoritmi		x		
Elementele de bază ale unui limbaj de programare (Pascal sau C, la alegere)		x		
Subprograme predefinite			x	
Tipuri structurate de date			x	
Fișiere text				x
Algoritmi elementari		x		
Subprograme definite de utilizator				x
Recursivitate				x
Metoda backtracking (iterativă sau recursivă)				x
Generarea elementelor combinatoriale				x
Liste			x	
Grafuri			x	

Niveluri cognitive	Cunoaștere	Înțelegere	Aplicare	Analiză/ Sinteză	Pondere
Competențe de evaluat	-construirea algoritmilor corespunzători unor prelucrări elementare și reprezentarea lor	-identificarea și utilizarea tipurilor de date predefinite specifice unui limbaj de programare;	-identificarea și utilizarea subprograme- lor predefinite elementare;	-definirea și apelul unor subprograme proprii cu înțelegerea mecanismelor de transfer prin intermediul parametrilor;	
Subiect	- analiza rezolvării unei probleme prin urmărirea evoluției valorilor variabilelor prelucrate de algoritmul corespunzător	-identificarea și utilizarea operatorilor predefiniți elementari;	- identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;	- organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control și module de program;	
		-identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;	- organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate.	- folosirea unor metode sistematice de rezolvare pentru probleme de generare;	
		-definirea și utilizarea unor tipuri de date proprii;	- analiza unor algoritmi echivalenți.	- analiza unor algoritmi echivalenți de rezolvare a unei probleme în vederea alegerii algoritmului optim.	
Subiect I	6,7%	10%	10%	6,7%	33,4%
Subiect II	6,65%	10%	10%	6,65%	33,3%
Subiect III	6,65%	10%	10%	6,65%	33,3%
Pondere	20%	30%	30%	20%	100%

Niveluri cognitive	Cunoaștere	Înțelegere	Aplicare	Analiză/ Sinteză	Pondere
Competențe de evaluat	-construirea algoritmilor corespunzători unor prelucrări elementare și reprezentarea lor	-identificarea și utilizarea tipurilor de date predefinite specifice unui limbaj de programare;	-identificarea și utilizarea subprograme- lor predefinite elementare;	-definirea și apelul unor subprograme proprii cu înțelegerea mecanismelor de transfer prin intermediul parametrilor;	
Subiect	- analiza rezolvării unei probleme prin urmărirea evoluției valorilor variabilelor prelucrate de algoritmul corespunzător.	-identificarea și utilizarea operatorilor predefiniți elementari;	- identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;	- organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control și module de program;	
		-identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;	- organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate.	- folosirea unor metode sistematice de rezolvare pentru probleme de generare;	
		-definirea și utilizarea unor tipuri de date proprii;	- analiza unor algoritmi echivalenți.	- analiza unor algoritmi echivalenți de rezolvare a unei probleme în vederea alegerii algoritmului optim.	
Subiect I	0,99 (1 item)	1,53(2 itemi)	1,53 (1 item)	0,99(1 item)	5 itemi
Subiect II	0,99(1 item)	1,49(2 itemi)	1,49(1 item)	0,99(1 item)	5 itemi
Subiect III	0,99(1 item)	1,49(1 item)	1,49(2 itemi)	0,99(1 item)	5 itemi
Pondere	3 itemi	5 itemi	4 itemi	3 itemi	15 itemi

Proba de evaluare:
SUBIECTUL I (30 de puncte)

- Indicați expresia **Pascal|C/C++** care are valoarea **true|1** dacă și numai dacă numărul natural memorat de variabila **x** de tip **integer|int** are exact două cifre. (4p.)
 - x div 100=0** **x/100==0**
 - (x div 100 =0) and (x mod 10=0)** **x/100==0 && x%10==0**
 - x div 10<>0** **x/10!=0**
 - (x div 100 =0) and (x div 10<>0)** **x/100==0 && x/10!=0**

2. Se consideră algoritmul alăturat descris în pseudocod.

S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

Scrieți valoarea care se afișează dacă se citește numărul $n=5172$. (6p.)

Scrieți programul **Pascal|C/C++** corespunzător algoritmului dat. (10p.)

Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **cât timp...execută** cu o structură repetitivă de un alt tip. (6p.)

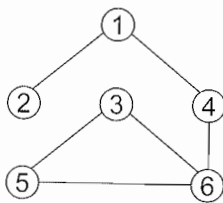
Scrieți toate valorile distincte, fiecare având exact patru cifre, care pot fi citite pentru variabila n astfel încât să se afișeze valoarea **2008**, pentru fiecare dintre acestea (4p.)

```

citește n (număr natural)
m ← 0
P ← 1
cât timp n > 0 execută
    c ← n % 10
    dacă c > 0 atunci
        c ← c - 1
    m ← m + c * P; P ← P * 10
    n ← [n / 10]
scrie m
    
```

SUBIECTUL al II-lea (30 de puncte)

1. Indicați numărul **minim** de muchii ce trebuie mutate în graful din figura alăturată astfel încât graful obținut să fie conex și fiecare nod să aparțină unui ciclu. (6p.)

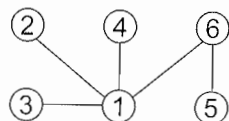


- a. 0 b. 1 c. 2 d. 4

2. Indicați valoarea expresiei **length(s)|strlen(s)** pentru variabila s de tip șir de caractere, declarată **var s:string[15];** și inițializată **s:="Proba_E";** | **char s[15]="Proba_E";** (4p.)

- a. 7 b. 15 c. 6 d. 5

3. Care sunt nodurile de tip frunză din arborele alăturat dacă se alege ca rădăcină nodul 6. (6p.)



4. O listă liniară simplu înlanțuită cu cel puțin 5 elemente, alocată dinamic, reține în câmpul **nr** al fiecărui element câte un număr natural, iar în câmpul **urm**, adresa elementului următor din listă sau **nil|NULL** dacă nu există un element următor. Variabila **prim** memorează adresa elementului aflat pe prima poziție în listă, **ultim** adresa elementului aflat pe ultima poziție în listă, iar **p** și **q** sunt două variabile de același tip cu **prim**. Scrieți poziția pe care se va găsi în lista modificată în urma executării secvenței alăturate, elementul aflat pe poziția a doua, în lista inițială. (4p.)

5. Scrieți un program **Pascal|C/C++** care citește de la tastatură un număr natural n ($2 < n \leq 15$) și construiește în memorie o matrice A cu n linii și n coloane în care orice element aflat pe prima linie sau pe prima coloană are valoarea 1 și oricare alt element A_{ij} din matrice este egal cu suma a două elemente din matrice, primul aflat pe linia i și pe coloana $j-1$, iar cel de-al doilea pe coloana j și pe linia $i-1$. Matricea va fi afișată pe ecran, linie cu linie, numerele de pe aceeași linie fiind separate prin câte un spațiu. Exemplu: pentru $n=4$, se afișează matricea alăturată. (10p.)

SUBIECTUL al III-lea (30 de puncte)

1. Pentru generarea numerelor cu n cifre formate cu elementele mulțimii $\{0,4,8\}$ se utilizează un algoritm backtracking care, pentru $n=2$, generează, în ordine, numerele **40,44,48,80,84,88**.

Dacă $n=4$ și se utilizează același algoritm, indicați numărul generat imediat după numărul **4008**. (4p.)

- a. 4040 b. 4024 c. 4080 d. 8004

2. Subprogramul **f** este definit mai jos. Scrieți ce se afișează pe ecran ca urmare a apelului **f(1,3)**. (6p.)

```

procedure f(x,y:integer);
var i:integer;
begin
    for i:=x to y do
        begin write(i);f(i+1,y) end
    end;

void f(int x,int y)
{ for (int i=x;i<=y;i++)
  {cout<<i;|printf(„%d“,i);
   f(i+1,y);}
}
    
```

3. Scrieți definiția completă a subprogramului **suma**, care primește prin cei 4 parametri v, n, i, j :

- v , un tablou unidimensional cu maximum 100 de elemente întregi din intervalul $[-1000, 1000]$ numerotate de la 1 la n ;
- n , un număr natural reprezentând numărul de elemente din tabloul v ;
- i și j , două valori naturale $1 \leq i \leq j \leq 100$; și returnează suma elementelor $v_1, v_2, \dots, v_{i-1}, v_{j+1}, \dots, v_n$ din tabloul v . (10p.)

4. Fișierul **NUMERE.IN** conține pe prima linie un număr natural nenul n ($1 \leq n \leq 100$) și pe următoarea linie, n numere reale pozitive, **ordonate crescător**, separate prin câte un spațiu.

a) Scrieți un program **Pascal|C/C++** care citește din fișierul **NUMERE.IN** numărul natural n și determină, utilizând un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate, numărul **minim** de intervale închise de forma $[x; x+1]$, cu x număr natural, a căror reuniune include toate numerele reale din fișier.

Exemplu: Dacă fișierul **NUMERE.IN** are conținutul:

6 2.3 2.3 2.8 5.7 5.9 6.3

atunci se afișează **3** (intervalele $[2;3], [5;6], [6;7]$ sunt cele 3 intervale de forma cerută care conțin numere din șir). (6p.)

b) Descrieți în limbaj natural metoda utilizată și explicați în ce constă eficiența ei. (4p.)

Barem de evaluare și notare și sugestii de rezolvare

Subiectul I

item	Rezolvare	Punctaj	Observații
1	Răspuns corect: d.	4 p	
2	a) Răspuns corect: 4061	6 p	
	b)		
	declararea variabilelor	1 p	
	citire date	1 p	
	scrierea date	1 p	
	atribuiri corecte	1 p	
	structura repetitivă	2 p	
	structură alternativă	3 p	
	corectitudine globală	1 p	
	c)		
	utilizarea unei structuri echivalente	3 p	
	algoritm echivalent	3 p	
	d. Răspuns corect: {3009,3019,3109,3119}	4 p	

Subiect II

item	Rezolvare	Punctaj	Observații
1	Răspuns corect: b.	6 p	Putem muta muchia [3,6] astfel încât să unească nodurile 2 și 4.
2	Răspuns corect: a.	4 p	
3	Răspuns corect: 2,3,4,5	6 p	(în orice ordine)
4	Răspuns corect: 3	4 p	Secvența mută ultimul element la începutul listei
5.	Pentru rezolvare corectă	10p	
	-declararea variabilelor	1 p	
	-citire date	1 p	
	-scriere date	1 p	
	-completarea elementelor de pe prima linie și prima coloană	2 p	
	-completarea elementelor din interior	2 p	
	-afișare matrice	2p	
	-corectitudine globală	1 p	

O soluție posibilă pentru itemul 5.

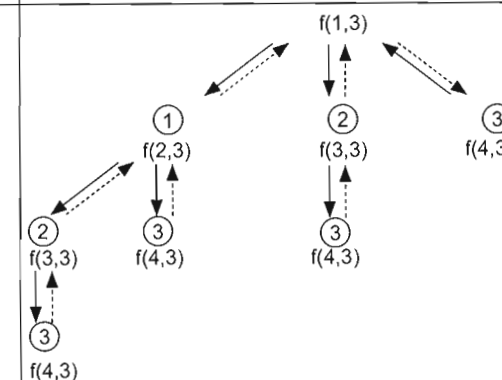
Se completează cu valoarea 1 elementele aflate pe prima linie sau pe prima coloană. Celelalte elemente A_{ij} primesc valoarea $A_{i,j-1} + A_{i-1,j}$.

```
var n,i,j:byte;
a:array[1..15,1..15] of longint;
begin
  readln(n);
  for i:=1 to n do begin
    A[1,i]:=1;A[i,1]:=1 end;
  for i:=2 to n do
    for j:=2 to n do
      A[i,j]:=A[i,j-1]+A[i-1,j];
  for i:=1 to n do
    begin
      for j:=1 to n do
        write(A[i,j], ' ');
      writeln
    end
  end.
```

```
# include <iostream.h>
unsigned long A[16][16];
unsigned n,i,j,k;
void main()
{ cin>>n;
  for(i=1;i<=n;i++)
    A[1][i]=A[i][1]=1;
  for(i=2;i<=n;i++)
    for(j=2;j<=n;j++)
      A[i][j]=
        A[i][j-1]+A[i-1][j];
  for(i=1;i<=n;i++)
    { for(j=1;j<=n;j++)
      cout<<A[i][j]<<' ';
      cout<<endl;
    }
}
```

Subiectul al III-lea

item	Rezolvare	Punctaj	Observații
1	Răspuns corect: a.	4 p	
2	Răspuns corect : 1233233	6 p	În figura următoare este reprezentată schema apelurilor recursive și sunt încercuite valorile afișate.
3	Pentru rezolvare corectă	10p	
	-tip subprogram	1 p	
	- antet corect	2 p	
	- suma elementelor $v_1 \dots v_{i-1}$	3 p	
	- suma elementelor $v_{j+1} \dots v_n$	3p	
	- returnare rezultat	1 p	
	- corectitudine globală	1 p	



	<p>O soluție posibilă. Suprogramul va calcula suma primelor $i-1$ elemente și a ultimelor $(n-j)$ elemente din tabloul unidimensional transmis ca parametru.</p> <p>Pascal:</p> <pre> type vector=array[1..100] of integer; function suma(v:vector;n,i,j:byte):longint; var s:longint;k:byte; begin s:=0; for k:=1 to i-1 do s:=s+v[k]; for k:=j+1 to n do s:=s+v[k]; suma:=s end; </pre> <p>C/C++</p> <pre> long suma (int v[101],unsigned n,unsigned i,unsigned j) { long s=0;unsigned k; for(k=1;k<=i-1;k++)s=s+v[k]; for(k=j+1;k<=n;k++)s=s+v[k]; return s; } </pre>		
4	<p>a. Pentru rezolvare corectă</p> <ul style="list-style-type: none"> - declarare variabile - operații cu fișiere - determinarea intervalelor - eficiență* 	<p>6 p.</p> <p>1 p</p> <p>1 p</p> <p>2 p</p> <p>2 p</p>	<p>*Se acordă punctajul dacă algoritmul este liniar și nu utilizează date structurate</p>
	<p>b. -descrierea unei metode corecte de rezolvare</p> <ul style="list-style-type: none"> -justificarea eficienței din punct de vedere al timpului și al spațiului de memorie de executare* 	<p>2 p</p> <p>2 p</p>	<p>*Se acordă punctajul dacă algoritmul este liniar și nu utilizează date structurate</p>

	<p>Sugestie de rezolvare.</p> <p>Se pornește de la partea întreagă p a primului număr și se citesc succesiv valori din fișier cât timp acestea fac parte din intervalul $[p,p+1]$. Apoi se alege un nou p, partea întreagă a numărului care nu mai face parte din interval, numărându-se un nou interval etc..</p> <p>Pascal</p> <pre> var f:text;n,i,nr,p:word;x:real; begin assign(f,'numere.in');reset(f); read(f,n,x); p:=trunc(x);nr:=1; for i:=2 to n do begin read(f,x); if (trunc(x)<>p) and (x<>p+1) then begin nr:=nr+1;p:=trunc(x); end end; close(f); write(nr) end. </pre> <p>C/C++</p> <pre> #include <fstream.h> #include <math.h> ifstream f(„numere.in“); unsigned n,i,nr,p;float x; void main() { f>>n>>x; p=floor(x);nr=1; for(i=2;i<=n;i++) { f>>x; if(floor(x)!=p&&x!=p+1) {nr++;p=floor(x);} } f.close();cout<<nr; } </pre>
--	--

EVALUARE DIGITALĂ – CONCLUZII

Evaluarea digitală permite realizarea unui contact mai strâns cu elevii, un control temeinic și operativ, constituindu-se într-o tehnică de evaluare complementară celei clasice.

Rețeaua de calculatoare permite stabilirea unui dialog permanent cu elevii: se prezintă informații, se verifică asimilarea cunoștințelor, se propun exerciții, după cum se dorește ca dominantă transmiterea de cunoștințe, evaluarea sau formarea de priceperi și deprinderi.

Când are rolul de a evalua competențele, calculatorul afișează mai întâi cerința care apare proiectată pe ecran și la care cel examinat trebuie să răspundă. Dispozitivul memorează răspunsul dat și timpul de gândire. În funcție de modul în care este proiectat testul, se poate alege obținerea unui feedback imediat (după fiecare întrebare) sau la sfârșitul testului.

Evaluarea digitală permite realizarea unei rețele complexe de instruire, existând posibilitatea controlului simultan al unui număr mare de elevi, care utilizează terminale conectate la un server, de unde se poate urmări și dirija procesul de evaluare.

Totodată, utilizarea calculatoarelor în aprecierea cunoștințelor, prezintă avantajele obiectivității și insensibilității la reacții afective legate de notare.

BIBLIOGRAFIE

1. *Abernot, Yvan*, 1998, Les méthodes d'évaluation scolaire, Nouvelle édition, DUNOD, Paris;
2. *Abrecht, Roland*, 1991, L'évaluation formative. Une analyse critique, Bruxelles, De Boeck;
3. *Barlow Michel*, 1992, L'évaluation scolaire. Decoder son langage, Chronique sociale, Lyon;
4. *Belair, Louis M.*, 1999, L'évaluation dans l'école. Nouvelles pratiques, ESF éditeur;
5. *Bosman Christian, Gerard François-Marie, Roegiers Xavier*, 2000, Quel avenir pour les compétences? De Boeck Université;
6. *Cardinet, Jean*, 1998, Pour apprécier le travail des élèves, De Boeck;
7. *Cerghit, Ioan*, 2008, Sisteme de instruire alternative și complementare, Editura Polirom, Iași;
8. *Cucoș, Constantin*, 2008, Teoria și metodologia evaluării, Editura Polirom, Iași;
9. *De Lansheere, G.*, 1975; Evaluarea continuă a elevilor și examenele, București, EDP;
10. *Figari, Gerard; Achouche, Mohamed*, 2001, L'activité évaluative reinterrogée, Bruxelles, De Boeck;
11. *Hadji, Charles*, 1992, L'évaluation des actions éducatives, PUF;
12. *Hadji, Charles*, 1989, L'évaluation, règles du jeu, Paris, ESF éditeur;
13. *Jinga. I., Petrescu, A.*, Evaluarea performanței școlare, București, Editura Delfin, 1996;
14. *Joița, Elena*, 2002, Educația cognitivă, Editura Polirom, Iași;
15. *Ketele, Jean Marie de*, 1986, L'évaluation: approche descriptive ou prescriptive? Bruxelles, De Boeck ;
16. *Landsheere, Viviane*, 1992, L'éducation et la formation, PUF, Paris;
17. *Lisievi, Petru*, 2002, Evaluarea în învățământ. Teorie, practică, instrumente, Editura ARAMIS, București;
18. *Manolescu, Marin*, Activitatea evaluativă între cogniție și metacogniție, București, Editura Meteor, 2004;
19. *MEN*, Reforma sistemului de evaluare și examinare, București, Editura Școala Românească, 1998;
20. *Meyer Geneviève*, 2000, De ce și cum evaluăm, Polirom , Iași;
21. *Neacșu, I.; Stoica, A. (coord)*, 1998, Ghid general de evaluare și examinare, M.I., CNEE, Aramis, București;
22. *OCDE*, 1999, Mesurer les connaissances et compétences des élèves. Un nouveau cadre d'évaluation;
23. *Peretti, André de*, 1996, Educația în schimbare, Iași, Editura Spiru Haret;
24. *Perrenoud Philippe*, 1998, L'évaluation des élèves. De la fabrication de l'excellence à la régulation des apprentissages. Entre deux logiques, Bruxelles, De Boeck;
25. *Potolea Dan, Manolescu, Marin*, 2006, Teoria și practica evaluării educaționale, Proiectul pentru Învățământul Rural, București

26. Potolea, Dan, Neacșu, Ioan; Radu, I.T., 1996, Reforma evaluării în învățământ, București, EDP;
27. Potolea, Dan, Păun E. Coord), Pedagogie, Editura Polirom, Iași, 2002;
28. Radu, I. T., 2000, Evaluarea în procesul didactic, EDP, București;
29. Rogiers Xavier, 1997, Analyser une action d'éducation ou de formation, De Boeck Université;
30. Scallon Gerard, 2000, L'evaluation formative, Bruxelles, De Boeck;
31. SNEE, 2001, Ghid de evaluare. Limba și literatura română, București, Aramis;
32. Stan, Cristian, Evaluarea și autoevaluarea în procesul didactic, în Ionescu Miron, Chiș Vasile (coord), 2001, în Pedagogie, Editura Presa Universitară Clujeană, Cluj-Napoca;
33. Stoica, Adrian (coord), 2001, Evaluarea curentă și examenele. Ghid pentru profesori, Prognosis, 2001;
34. Strungă, Constantin, 1999, Evaluarea școlară, Editura Universității de Vest, Timișoara;
35. Vogler Jean (coord.), 2000, Evaluarea în învățământul preuniversitar, Polirom, Iași;
36. Voiculescu Elisabeta, 2001, Factorii subiectivi ai evaluării școlare. Cunoaștere și control, Aramis;
37. *** CNCEIP, Programul Național de Dezvoltare a Competențelor de Evaluare ale Cadrelor Didactice (DeCeE), 2008
38. Pinte, Rodica; Lițoiu, Nicolae, 2001, Ghid de evaluare Informatică și Tehnologia informației, Editura Aramis, București;
39. Hussar, Elena, 2007, Școala incluzivă-Școală europeană, Editura CCD Bacău;
40. Cerghit, Ioan, 2008, Sisteme de instruire alternative și complementare, Editura Polirom, Iași;
41. Cucoș, Constantin, 2008, Teoria și metodologia evaluării, Editura Polirom, Iași
42. Jinga. I., Petrescu, A., Evaluarea performanței școlare, București, Editura Delfin, 1996;
43. Neacșu, I.; Stoica, A. (coord), 1998, Ghid general de evaluare și examinare, M.I., CNEE, Aramis, București;
44. *** CNCEIP, Programul Național de Dezvoltare a Competențelor de Evaluare ale Cadrelor Didactice (DeCeE), 2008
45. Cojocariu, V.M., 2003, Educația pentru schimbare și creativitate, E.D.P.
46. Șoitu, L., Cherciu, R.D., 2006, Strategii educaționale centrate pe elev, www.supradotați.ro
47. Dumitriu, C., 2003, Strategii alternative de evaluare. Modele teoretico-experimentale, EDP.