

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII
AL REPUBLICII MOLDOVA**

CURRICULUM NAȚIONAL

ARIA CURRICULARĂ
TEHNOLOGII

DISCIPLINA OPȚIONALĂ

PROGRAMAREA ALGORITMILOR ÎN C / C++

Clasele VII-XII

Chișinău, 2020

Aprobat:

- Consiliul Național pentru Curriculum, proces-verbal nr. 25 din 25.06.2020
- Ordinul Ministerului Educației, Culturii și Cercetării nr. 653/2020

COORDONATOR:

Angela PRISĂCARU, consultant principal, Ministerul Educației, Culturii și Cercetării

GRUPUL DE LUCRU:

Sergiu CORLAT (coordonator grup), magistru în științe exacte, grad didactic superior, Universitatea Tehnică a Moldovei, coordonator;

Grigore VASILACHE, grad didactic superior, director IPLT “Mircea Eliade”, profesor de informatică.

Componența grupului de lucru a fost Aprobată prin Ordinul Ministerului Educației, Culturii și Cercetării nr. 431/2020.

Curriculumul a fost elaborat cu suportul proiectului „*Tekwill în Fiecare Școală/TwentyTU*”, în cadrul Memorandumului de colaborare privind implementarea educației digitale la nivel național, semnat la 14 iunie 2018 între Ministerul Educației, Culturii și Cercetării, Ministerul Economiei și Infrastructurii, Asociația Națională a Companiilor din domeniul TIC și Instituția Publică Centrul de Instruire și Inovații TIC – Tekwill.

Preliminarii:

Curriculumul la disciplina opțională ”Programarea algoritmilor în C / C++” este un document normativ și obligatoriu pentru realizarea procesului de predare-învățare în învățământul gimnazial și liceal. Cursul ” Programarea algoritmilor în C / C++” este un curs opțional de extindere în cadrul disciplinei de studiu Informatica, pentru treptele de studii gimnaziu și liceu.

Obiect de studiu: fundamentele programării, aplicării și proiectării algoritmilor.

Scop:

- Context cognitiv: cursul a fost elaborat în scopul facilitării procesului de formare timpurie la elevi a gândirii algoritmice prin:
 - Studiul algoritmilor în context istoric, științific și aplicativ;
 - Analiza instrumentelor informatice pentru implementarea algoritmilor;
 - Analiza și aplicarea modelelor algoritmice pentru rezolvarea problemelor tradițional asociate altor domenii de studii sau activitate decât cel informatic;
 - Proiectarea algoritmilor.
- Context pedagogic: cursul a fost elaborat în scopul extinderii aplicării modelelor algoritmice de predare-învățare prin:
 - dezvoltarea gamei de cunoștințe adaptate la vârsta elevilor, predate în cadrul disciplinelor școlare;
 - facilitarea activităților educaționale cu caracter interdisciplinar;
 - adaptarea metodelor de predare-învățare la instruirea algoritmică;
 - echilibrarea încărcăturii informaționale și continuității între clase și trepte de învățământ;
 - diferențierea și individualizarea predării-învățării;
 - stabilirea unor precondiții minime pentru instruirea de performanță în domeniul Informaticii.

Funcțiile curriculumului la disciplina opțională ” Programarea algoritmilor în C / C++”:

- act normativ al procesului de predare-învățare-evaluare a cursului în contextul pedagogiei axate pe competențe;
- reper pentru proiectarea didactică și desfășurarea procesului educațional din perspectiva pedagogii axate pe competențe;
- componentă de bază pentru elaborarea strategiei de evaluare la curs;
- orientare a procesului educațional spre formare de competențe la elevi;
- componentă fundamentală pentru elaborarea manualelor școlare în format tradițional și digital, suporturilor de curs digitale, ghidurilor metodologice, testelor de evaluare.

Grupuri țintă: curriculumul cursului se adresează cadrelor didactice, autorilor de manuale și resurse digitale educaționale, elevilor și altor categorii de persoane interesate.

Concepția didactică a cursului

Definirea cursului școlar opțional Programarea algoritmilor în C / C++. Obiectul de studiu al cursului este modelarea informatică a proceselor de diversă natură (tehnică, socială, biologică etc) prin prisma algoritmizării lor și descrierii ulterioare cu ajutorul limbajelor de programare procedurale de nivel înalt.

Cursul vine să susțină efortul disciplinelor școlare obligatorii din domeniul matematică și științe (STEAM) în formarea gândirii analitice a elevilor, dezvoltarea logicii, atât de ordin general, cât și pe arii de cunoaștere specializate. Reprezintă o opțiune pentru consolidarea competențelor specifice disciplinei școlare Informatica dar și un mediu de autoinstruire pentru elevii, a căror traseu formativ deviază către domeniul profesional asociat Tehnologiilor Informaționale și Comunicaționale.

Valoarea formativă a cursului constă în:

- extinderea modelului școlar de cunoaștere a conceptelor de bază ale informaticii, care includ elemente de logică, algoritmizare, modelare și programare a proceselor naturale și ale activității umane;
- dezvoltarea deprinderilor practice de utilizare a modelelor informatice pentru optimizarea operațiilor de prelucrare a informației;

Principiile specifice predării-învățării ale cursului. Curriculumul cursului opțional *Programarea algoritmilor în C / C++* propune un model de studiu modular al acestei discipline. Acest model contribuie la abordarea flexibilă a conținuturilor cursului pentru elevii a căror competențe la momentul lansării procesului de studiu permit startul din punctul de joncțiune a modulelor cursului, cu un efort ulterior aprofundat către componenta algoritmică și performanța informatică. În același timp cursul vine să extindă viziunea elevilor asupra metodelor de implementare a conceptelor informatice pentru rezolvarea problemelor generate de societatea contemporană. În acest context se conturează următoarele proprietăți specifice ale cursului:

Abordarea modulară a cursului – structurarea conținuturilor într-un model modular, flexibil, care are ca scop dezvoltarea competențelor elevului de a gândi algoritmic și de a utiliza instrumente informatice de modelare.

Centrarea activității / demersului didactic pe elev – acceptarea unui model de învățare activă, centrat pe elev, orientat către activități individuale sau în grup, care să permită dezvoltarea independenței de acțiune, originalității, creativității, capacității de lucru în echipă, combinând acestea cu individualizarea ritmului de învățare – similar modelului pentru disciplina școlară Informatica.

Asigurarea funcționalității sociale a procesului didactic – dezvoltarea aptitudinilor și competențelor necesare pentru integrarea organică a elevilor în societatea informațională. Principiul se realizează prin activități practice de studiere și utilizare ale algoritmilor și elementelor limbajelor de programare de nivel înalt.

Asigurarea legăturilor interdisciplinare – abordarea unor modele complexe, corelate cu alte discipline școlare, dezvoltarea și implementarea algoritmilor pentru identificarea soluțiilor problemelor din cele mai diverse domenii.

Orientări generale de predare-învățare a cursului. Procesul general de predare-învățare a cursului *Introducere în programarea algoritmilor* este elaborat în contextul sistemului de competențe pentru învățământul gimnazial și liceal, cu diferențiere a conținuturilor specifice fiecărei trepte.

Competențe-cheie/transversale

- Competențe de învățare/de a învăța să înveți.
- Competențe de comunicare în limba română.
- Competențe de comunicare într-o limbă străină.
- Competențe acțional-strategice.
- Competențe de autocunoaștere și autorealizare.
- Competențe interpersonale, civice, morale.
- Competențe de bază în matematică, științe și tehnologie.
- Competențe digitale în domeniul tehnologiei informației și comunicațiilor (TIC).
- Competențe culturale, interculturale (de a recepta și a crea valori).
- Competențe antreprenoriale.

Competențele trans-disciplinare

Competențe de învățare/de a învăța să înveți:

- stăpânirea metodologiei de integrare a cunoștințelor de bază despre natură, om și societate în scopul satisfacerii nevoilor și acționării pentru îmbunătățirea calității vieții personale și sociale.

Competențe de comunicare în limba română:

- comunicare argumentată în limba română în situații reale ale vieții;
- comunicare într-un limbaj științific argumentat.

Competențe de comunicare într-o limbă străină:

- cunoașterea unei limbi străine;
- comunicare argumentată într-o limbă străină în situații reale de viață.

Competențe de bază în Matematică, Științe și Tehnologie:

- organizarea activității personale în condițiile unor tehnologii aflate în permanentă schimbare;
- dobândirea și stăpânirea de cunoștințe fundamentale din domeniile matematică, informatică, științe ale naturii și tehnologii în coraport cu nevoile sale;
- propunerea de idei noi în domeniile științifice.

Competențe acțional-strategice:

- proiectarea propriilor activități, orientarea către rezultatul final, propunerea de soluții de rezolvare a situațiilor-problemă din diverse domenii;
- acționarea autonomă și creativă în diverse situații de viață în scopul protecției mediului ambiant.

Competențe digitale, în domeniul tehnologiei informației și comunicațiilor (TIC):

- utilizarea în situații reale a instrumentelor cu acțiune digitală;
- crearea de documente în domeniul comunicativ, informațional și utilizarea în situații reale a serviciilor electronice, inclusiv a Internetului.

Competențe interpersonale, civice, morale:

- colaborarea în grup/echipă, prevenirea situațiilor de conflict, respectarea opiniilor semenilor săi;
- manifestarea unei poziții civice active, solidaritate și coeziune socială pentru o societate nondiscriminatoare;
- acționarea în diferite situații de viață în baza normelor și valorilor moral-spirituale.

Competențe de autocunoaștere și autorealizare:

- gândire critică asupra activității sale în scopul autodezvoltării continue și autorealizării personale;
- asumarea responsabilității pentru un mod sănătos de viață;
- adaptarea la condiții și situații noi.

Competențe culturale, interculturale (de a recepta și de a crea valori):

- orientarea în sistemele de valori ale culturii naționale și culturilor altor etnii în scopul aplicării lor creative și autorealizării personale;
- toleranță în receptarea valorilor interculturale.

Competențe antreprenoriale:

- cunoștințe și abilități de autorealizare în domeniul antreprenorial în condițiile economiei de piață;
- alegerea conștientă a ariei de activitate profesională în viitor.

Competențe-cheie dezvoltate în cadrul cursului *Introducere în programarea algoritmilor*

Dezvoltare directă

Competențe de bază în matematică, științe și tehnologie – dobândirea și utilizarea cunoștințelor de bază din domeniul matematică, științe și tehnologii pentru rezolvarea unor probleme și situații din cotidian, pentru continuarea studiilor profesionale tehnice sau superioare după absolvirea învățământului liceal.

Competențe acțional-strategice – identificarea problemelor acțional-strategice și propunerea soluțiilor de rezolvare. Planificarea propriilor activități și prognozarea rezultatei așteptate. Elaborarea propriilor planuri și a strategiilor de activitate individuală și în grup.

Competențe digitale, în domeniul tehnologiei informației și a comunicațiilor (TIC) – utilizarea în situații reale a instrumentelor cu acțiune digitală pentru comunicare, colectare, stocare și prelucrare a informațiilor. Crearea de documente electronice. Utilizarea serviciilor electronice de bază (e-guvernare, e-business, e-educație, e-sănătate, e-cultură).

Dezvoltare tangențială

Competențe interpersonale, civice, morale – lucrul în echipă.

Competențe de învățare / de a învăța să înveți – planificarea și organizarea propriei învățări atât individual cât și în grup.

Competențe de comunicare într-o limbă străină – comunicarea într-o limbă străină în situații noi.

*Competențele specifice dezvoltate în cadrul cursului **Introducere în programarea algoritmilor***

1. Utilizarea mijloacelor TIC pentru prelucrarea datelor.
2. Perceperea științifică a rolului și impactului fenomenelor informatice din societatea contemporană, cunoașterea principiilor și legităților de funcționare a echipamentelor digitale.
3. Elaborarea modelelor informatice ale obiectelor, sistemelor și proceselor frecvent întâlnite în activitatea cotidiană.
4. Aplicarea mijloacelor de algoritmizare, formalizare, analiză, sinteză și implementare pe calculator a instrumentarului necesar pentru soluționarea problemelor legate de prelucrarea digitală a informației în activitatea cotidiană.
5. Efectuarea experimentelor virtuale, rezolvarea problemelor de activitate cotidiană și elaborarea de modele ale fenomenelor studiate folosind aplicații, laboratoare și medii digitale educaționale; interpretarea rezultatelor obținute.
6. Respectarea cadrului normativ-juridic, a regulilor de securitate, ergonomice și etice în utilizarea mijloacelor tehnologiei informației și comunicațiilor.

Administrarea cursului

| Statut | Arie curriculară | Treaptă educațională | Număr module | Număr de ore |
|-----------------|-----------------------|----------------------|--------------|--------------|
| Opțional | Matematică și Științe | Gimnaziu, Liceu | 2 | 34 |

Repartizare ore pe module

| Modulul I: Elemente de programare | | | |
|---|----------|-------|------|
| Unități de învățare | Ore | | Note |
| | Gimnaziu | Liceu | |
| Ce este programarea | 1 | 1 | |
| Medii de programare | 1 | 1 | |
| Date și variabile | 1 | 1 | |
| Structura programului. Afișare și citire | 2 | 1 | |
| Programe cu structură liniară. Operații și expresii aritmetice | 2 | 1 | |
| Selecții. Operații relaționale și logice. | 2 | 1 | |
| Selecții multiple. | 1 | 1 | |
| Repetarea acțiunilor | 1 | 1 | |
| Bucle dirijate de contor | 1 | 1 | |
| Bucle dirijate de condiție | 1 | 1 | |
| Tablouri liniare | 1 | 1 | |
| Tablouri bidimensionale | 1 | 1 | |
| Șiruri de caractere. Operații asupra șirurilor de caractere | 2 | 1 | |
| Utilizarea subprogramelor | - | 1 | |
| Funcții | - | 2 | |
| Variabile globale și locale | - | 1 | |
| Total ore Modul 1 | 17 | 17 | |

Modulul II: Introducere în Algoritmi

| Unități de învățare | Ore | | Note |
|---|-----------|-----------|------|
| | Gimnaziu | Liceu | |
| Algoritmi. Istorie | 1 | 1 | |
| Metode de reprezentare a algoritmilor | 1 | 1 | |
| Algoritmi elementari | 2 | 1 | |
| Căutarea în tablouri | 1 | 1 | |
| Ordonarea obiectelor. Algoritmul BubbleSort | 2 | 1 | |
| Alt algoritm de sortare: Sortarea prin selecție | 1 | 1 | |
| Ce înseamnă Forța Brută? Probleme de forță brută | 2 | 1 | |
| Recursia | - | 1 | |
| Algoritmul de propagare a undei | - | 1 | |
| Tehnici recursive: Divide et Impera | 1 | 1 | |
| Tehnici recursive: Reluare | 1 | 1 | |
| Grafuri. Noțiuni. Reprezentări | 1 | 1 | |
| Grafuri. Parcurgeri | 1 | 1 | |
| Arbori de cost minim | 1 | 1 | |
| Drumuri minime în graf | - | 1 | |
| Algoritmi euristici. Greedy | 2 | 1 | |
| Algoritmi euristici. Jocul "Viața" | - | 1 | |
| Total ore Modul 2 | 17 | 17 | |
| Total ore curs | 34 | 34 | |

Note:

- Repartizarea orelor pe teme este orientativă.
- Ordinea temelor poate fi schimbată dacă nu este afectată logica științifică sau didactică.

Modulul I: Elemente de programare

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|--|--|
| Unitatea 1: Elemente de programare | | |
| <ul style="list-style-type: none"> cunoașterea noțiunilor de bază ale asociației procesului de programare identificarea produselor program și a elementelor acestora descrierea etapelor evoluției limbajelor de programare explicarea proceselor aferente creării unui program | <p>Programul</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Instrucțiuni Algoritmi Programe Limbaje de programare <ul style="list-style-type: none"> Generații Limbajul C Dezvoltarea C++ | <p>Exerciții de:</p> <ul style="list-style-type: none"> Descriere a domeniilor de utilizare a programelor Argumentare a necesității utilizării programelor scriere a programului "Hello, World!" și a programelor similare compilare a programelor scrise corectare a erorilor de program lansare în execuție a programelor |
| <ul style="list-style-type: none"> descrierea etapelor elaborării unui program în limbajele de programare C/C++ Utilizarea mediilor de dezvoltare pentru crearea programelor | <p>Medii de dezvoltare a programelor</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Etapile elaborării unui program. <ul style="list-style-type: none"> Scrierea codului Depanarea Compilarea Executarea programului Medii de dezvoltare <ul style="list-style-type: none"> Proprietăți Exemple | <p>Exerciții de:</p> <ul style="list-style-type: none"> Identificare a mediilor de dezvoltare a programelor pentru limbajele C/C++ prezente pe calculatorul local Accesare, înregistrare și lansare a mediilor de dezvoltare web Identificare a componentelor de control esențiale ale mediilor de dezvoltare Personalizare a interfeței mediilor de dezvoltare Scriere și depanare a programelor în medii de dezvoltare a programelor |
| Unitatea 2: Date și Acțiuni | | |
| <ul style="list-style-type: none"> recunoașterea noțiunilor de | <p>Date și Variabile</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Date. | <p>Exerciții de:</p> <ul style="list-style-type: none"> Selecție a tipului adecvat al datelor. |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|--|--|
| <ul style="list-style-type: none"> bază asociate datelor asocierea tipurilor adecvate pentru valori numerice și caracteriale declarare a variabilelor definire a constantelor | <ul style="list-style-type: none"> Tipuri de date simple: <ul style="list-style-type: none"> int float char Identificatori Variabile Constante | <ul style="list-style-type: none"> Declarare a variabilelor Definire a constantelor Atribuire directă a valorilor variabilelor declarate |
| <ul style="list-style-type: none"> Afișarea mesajelor și valorilor Atribuirea valorilor variabilelor simple în corespondere cu tipul acestora | <p>Afișare și citire date</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Introducerea și afișarea datelor în programele C: <ul style="list-style-type: none"> printf scanf specificatori de format Introducerea și afișarea datelor în programele C++: <ul style="list-style-type: none"> cin >> cout << | <p>Exerciții de:</p> <ul style="list-style-type: none"> Atribuire a valorilor variabilelor cu ajutorul instrucțiunilor de citire (scanf (C), cin (C++)) Afișare valori ale variabilelor (printf (C), cout (C++)) Afișare formatată a valorilor variabilelor |
| <ul style="list-style-type: none"> Enumerarea operațiilor aplicabile variabilelor numerice Descrierea expresiilor aritmetice Atribuirea rezultatelor către o variabilă | <p>Operații aritmetice și Expresii</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Operații <ul style="list-style-type: none"> atribuire + - * / % Expresii <ul style="list-style-type: none"> Prioritatea operațiilor Paranteze | <p>Exerciții de:</p> <ul style="list-style-type: none"> Atribuire a valorilor variabilelor prin calcularea expresiilor simple Afișare a valorilor expresiilor calculate Afișare formatată a valorilor expresiilor calculate Formare a expresiilor aritmetice complexe <p>Probleme de:</p> <ul style="list-style-type: none"> Realizare a calculelor elementare Calculare a ariei, volumului, figurilor geometrice, intervalelor de timp. |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|--|--|
| Unitatea 3: Selecții | | |
| <ul style="list-style-type: none"> descrierea noțiunii de ramificare în program formarea expresiilor relaționale integrarea expresiilor relaționale în instrucțiunile de selecție elaborarea programelor care vor conține instrucțiuni de selecție | <p>Operații relaționale și Selecții Subiecte:</p> <ul style="list-style-type: none"> Operații relaționale <ul style="list-style-type: none"> <, >, <=, >=, ==, != Instrucțiunea de selecție if <ul style="list-style-type: none"> Schema logică de execuție a instrucțiunii Selectarea ramurii de execuție | <p>Exerciții de:</p> <ul style="list-style-type: none"> Formare și evaluare a expresiilor relaționale simple Formare și evaluare a expresiilor relaționale complexe <p>Probleme de:</p> <ul style="list-style-type: none"> Determinare a valorii maxime / minime Verificare a parității / divizibilității Modelare a calculatoarelor aritmetice Transformări multiple |
| <ul style="list-style-type: none"> Identificare a operațiilor logice descrierea expresiilor relaționale complexe folosind instrumentarul limbajului de programare utilizarea instrucțiunilor de selecție multiplă pentru verificarea condițiilor complexe | <p>Operații logice și Selecții complexe Subiecte:</p> <ul style="list-style-type: none"> Operații logice <ul style="list-style-type: none"> &&, , ! Expresii logice Tabele de adevăr Selecții imbricate | <p>Exerciții de:</p> <ul style="list-style-type: none"> Formare și evaluare a expresiilor logice simple Formare și evaluare a expresiilor logice complexe Descriere a selecțiilor multiple Descriere a selecțiilor imbricate <p>Probleme de:</p> <ul style="list-style-type: none"> Identificare a tipului de triunghi cunoscând măsurile unghiurilor Modelare a situațiilor decizionale complexe din activități cotidiene |
| <ul style="list-style-type: none"> Motivare a necesității utilizării instrucțiunilor de selecție multiplă Descriere a setului de ramuri ale | <p>Selecții multiple Subiecte:</p> <ul style="list-style-type: none"> Instrucțiunea de selecție multiplă (switch) <ul style="list-style-type: none"> Schema logică a instrucțiunii Selectorul <ul style="list-style-type: none"> Variabilă Expresie Ramurile de selecție | <p>Exerciții de:</p> <ul style="list-style-type: none"> Formare și evaluare a expresiilor de selecție multiplă Identificare a setului de valori și acțiuni pentru selector <p>Probleme de:</p> <ul style="list-style-type: none"> Transformări multiple |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|--|--|
| selecției multiple <ul style="list-style-type: none"> Integrare a instrucțiunilor de selecție multiplă în programe | | <ul style="list-style-type: none"> Modelare a situațiilor decizionale complexe din activități cotidiene |
| Unitatea 4: Bucle | | |
| <ul style="list-style-type: none"> descrierea noțiunii de repetare în programare identificarea situațiilor care implică utilizarea repetărilor explicarea modelului de funcționare a repetărilor contorizate / condiționate | Repetarea acțiunilor Subiecte: <ul style="list-style-type: none"> Repetarea acțiunilor <ul style="list-style-type: none"> Repetarea contorizată Schema logică a repetării contorizate Repetarea condiționată Schema logică a repetării condiționate | Exerciții de: <ul style="list-style-type: none"> Modelare a repetărilor contorizate în calcule aritmetice Modelare a repetărilor condiționate în calcule aritmetice Descriere a situațiilor cotidiene care implică repetarea contorizată / condiționată |
| <ul style="list-style-type: none"> Explicarea noțiunii de buclă cu contor Descrierea structurii buclelor cu contor Identificarea algoritmilor / acțiunilor care pot fi realizate cu ajutorul buclelor cu contor Elaborarea programelor cu repetări contorizate | Bucle cu contor Subiecte: <ul style="list-style-type: none"> Bucle dirijate de contor Structura instrucțiunii for <ul style="list-style-type: none"> Contorul Valoarea inițială Valoarea / expresia finală Pasul de modificare Bucle imbricate. | Exerciții de: <ul style="list-style-type: none"> Verificare a corectitudinii descrierii instrucțiunii – buclă Descriere a instrucțiunilor – buclă cu ajutorul instrumentarului limbajului de programare Evaluare a rezultatelor afișate de programele cu instrucțiuni – buclă Transformare a programelor prin utilizarea instrucțiunilor buclă Probleme de: <ul style="list-style-type: none"> Calculare a expresiilor aritmetice: sume, progresii, factorial, etc. Generare a șirurilor celebre de numere: Fibonacci, pătrate, prime, etc. Desenare în consolă |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|--|---|
| <ul style="list-style-type: none"> Explicarea noțiunii de buclă dirijată de condiție Descrierea structurii instrucțiunilor buclă cu pre / post condiție Identificarea algoritmilor / acțiunilor care pot fi realizate cu ajutorul buclelor cu pre / post condiție Elaborarea programelor cu repetări condiționate | <p>Bucle dirijate de condiție Subiecte:</p> <ul style="list-style-type: none"> Bucle dirijate de condiție While (precondiție) Do – while (postcondiție) Structura instrucțiunii while <ul style="list-style-type: none"> Expresia evaluată Condiția de repetare Corpul buclei Structura instrucțiunii do - while Bucle imbricate. | <ul style="list-style-type: none"> Efectuare a calculelor pe tabla de șah, etc. <p>Exerciții de:</p> <ul style="list-style-type: none"> Verificare a corectitudinii buclelor condiționate Descriere a buclelor condiționate cu ajutorul instrumentarului limbajului de programare Evaluare a rezultatelor afișate de programele cu bucle condiționate Transformare a programelor prin utilizarea instrucțiunilor buclă <p>Probleme de:</p> <ul style="list-style-type: none"> Calculare a sumei cifrelor în numere întregi. Repetare condiționată a calculelor. Verificare a corectitudinii datelor introduse Desenare controlată în consolă |
| Unitatea 5: Structuri de date | | |
| <ul style="list-style-type: none"> Explicare a noțiunii de structură de date Descrierea structurii unui tablou liniar Identificarea elementelor individuale ale unui tablou liniar Declarare a variabilelor de tip tablou liniar Utilizare a tablourilor liniare pentru stocarea și operarea cu date numerice | <p>Tablouri liniare Subiecte:</p> <ul style="list-style-type: none"> Tablouri liniare <ul style="list-style-type: none"> Declararea Atribuirea valorilor Parcurgerea Afișarea Operații cu tablouri liniare <ul style="list-style-type: none"> Accesarea valorilor elementelor după indice Căutarea elementelor cu valoare dată | <p>Exerciții de:</p> <ul style="list-style-type: none"> Verificare a corectitudinii declarării tablourilor Verificare a corectitudinii introducerii datelor în tablouri Evaluare a rezultatelor afișate de programele cu tablouri liniare Transformare a programelor prin utilizarea tablourilor <p>Probleme de:</p> <ul style="list-style-type: none"> Calculare a rezultatelor operațiilor aritmetice pe submulțimi din elementele tabloului. Căutare a elementului cu proprietăți date în tablou Adăugare / înlocuire a elementelor în tablou |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|--|---|
| | | <ul style="list-style-type: none"> • Deplasare a elementelor tabloului • Ordonare a valorilor elementelor tabloului după un criteriu dat • Căutare a secvențelor de valori cu proprietăți date, etc. |
| <ul style="list-style-type: none"> • Descrierea structurii unui tablou bidimensional • Identificarea elementelor individuale ale unui tablou bidimensional • Declarare a variabilelor de tip tablou bidimensional • Utilizare a tablourilor bidimensionale pentru analiza datelor numerice | <p>Tablouri bidimensionale Subiecte:</p> <ul style="list-style-type: none"> • Tablouri bidimensionale <ul style="list-style-type: none"> ○ Declararea ○ Atribuirea valorilor ○ Afișarea • Operații cu tablouri bidimensionale <ul style="list-style-type: none"> ○ Accesarea valorilor elementelor după indici ○ Căutarea elementelor cu valoare dată • Tablouri bidimensionale pătrate <ul style="list-style-type: none"> ○ Diagonala principală ○ Diagonala secundară | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Verificare a corectitudinii declarării tablourilor bidimensionale • Verificare a corectitudinii introducerii datelor în tablouri • Evaluare a rezultatelor afișate de programele cu tablouri bidimensionale • Transformare a programelor prin utilizarea tablourilor bidimensionale <p>Probleme de:</p> <ul style="list-style-type: none"> • Căutare a elementului cu proprietăți date în tablou • Completare a elementelor tabloului bidimensional cu valori conform unor reguli prestabilite • Adăugare / înlocuire a elementelor în tablou • Excludere / adăugare a liniilor și coloanelor de valori • Transformări simetrice ale tablourilor pătrate |
| <ul style="list-style-type: none"> • Descrierea noțiunii de șir de caractere în calitate de tip de date structurat • Identificarea elementelor individuale ale unui șir de caractere | <p>Șiruri de caractere Subiecte</p> <ul style="list-style-type: none"> • Șiruri de caractere: <ul style="list-style-type: none"> ○ Declararea ○ Atribuirea valorilor (citirea) ○ Afișarea • Operații cu șiruri de caractere: <ul style="list-style-type: none"> ○ Determinare lungime, | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Verificare a corectitudinii declarării șirurilor de caractere • Verificare a corectitudinii introducerii șirurilor de caractere • Evaluare a rezultatelor afișate |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|---|--|
| <ul style="list-style-type: none"> • Selectarea instrumentarului de operare cu șiruri de caractere în funcție de proprietățile acestora • Utilizarea șirurilor de caractere pentru stocarea datelor în program | <ul style="list-style-type: none"> ○ Comparare, ○ Căutare, ○ Concatenare, ○ Copiere, ○ Radiere. | <p>Probleme de:</p> <ul style="list-style-type: none"> • Transformare a șirurilor de caractere • Modelare a operațiilor pe șiruri de caractere • Verificare a proprietăților șirurilor de caractere • Căutare a secvențelor de caractere cu proprietăți date, etc. |
| Unitatea 6: Funcții | | |
| <ul style="list-style-type: none"> • Descrierea noțiunii de subprogram • Motivarea utilizării subprogramelor • Analiza modele-lor de probleme, evenimente și fenomene pentru divizare în subprobleme corelate | <p>Subprograme Subiecte</p> <ul style="list-style-type: none"> • Rezolvarea informatică a problemelor complexe <ul style="list-style-type: none"> ○ Divizarea în subprobleme ○ Corelarea subproblemelor ○ Asamblarea soluțiilor • Noțiunea de subprogram • Tipuri de subprograme | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Divizare a problemelor cu structură complexă în probleme elementare • Determinare a dependenței între subprobleme • Modelare a transferului de date între subprobleme • Clasificare a subprogramelor |
| <ul style="list-style-type: none"> • Motivarea utilizării funcțiilor în program • Analiza problemelor, evenimentelor, fenomenelor și situațiilor pentru modelarea informatică a acestora cu ajutorul funcțiilor • Descrierea structurii unei funcții C / C++ | <p>Funcții Subiecte</p> <ul style="list-style-type: none"> • Declararea funcțiilor <ul style="list-style-type: none"> ○ Antet ○ Descriere ○ Variabile locale ○ Returnare valoare • Apelarea funcțiilor <ul style="list-style-type: none"> ○ Parametri ○ Returnarea rezultatului • Tipul void <ul style="list-style-type: none"> ○ Referințe ○ Parametri cu structură complexă | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Verificare a corectitudinii antetului / apelului funcției • Evaluare a rezultatelor afișate de program • Controlul valorilor variabilelor în procesul de realizare a instrucțiunilor programului (funcția main – funcția utilizatorului) • Transformare a programelor prin utilizarea funcțiilor <p>Probleme de:</p> <ul style="list-style-type: none"> • Calculare a valorilor numerice simple cu ajutorul funcțiilor. |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|---|--|
| <ul style="list-style-type: none"> Elaborarea propriilor funcții | | <ul style="list-style-type: none"> Identificare a eficienței programelor cu funcții Modificare a valorilor variabilelor în funcții |
| <ul style="list-style-type: none"> Descrierea proceselor recursive Identificarea în modelele informatice a modelelor recursive și a componentelor acestora Descrierea structurii unei funcții recursive Elaborarea propriilor funcții recursive | <p>Funcții recursive Subiecte</p> <ul style="list-style-type: none"> Recursia <ul style="list-style-type: none"> Regula de consistență Cazul elementar Formula recurentă Funcții recursive. <ul style="list-style-type: none"> Antet Structură Controlul execuției funcțiilor recursive. Transformarea funcțiilor | <p>Exerciții de:</p> <ul style="list-style-type: none"> Verificare a corectitudinii antetului / apelului funcției recursive Evaluare a rezultatelor afișate de programul cu funcții recursive Controlul valorilor variabilelor în procesele recursive Transformare a programelor prin utilizarea funcțiilor recursive <p>Probleme de:</p> <ul style="list-style-type: none"> Calcul aritmetic recursiv Combinatorică Logică Misiuni complexe, reducibile la recursie |

Modulul II: Introducere în Algoritmi

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|---|--|
| Unitatea 1: Algoritmica | | |
| <ul style="list-style-type: none"> descrierea noțiunii de algoritm enumerarea proprietăților unui algoritm identificarea algoritmilor în activitatea umană | <p>Algoritmi. Istorie Subiecte:</p> <ul style="list-style-type: none"> Noțiune de algoritm. Istorie. Proprietăți Algoritmi celebri | <p>Exerciții de:</p> <ul style="list-style-type: none"> Identificare a algoritmilor Descriere a proprietăților algoritmilor Explicare a proprietăților algoritmilor Enumerare a domeniilor în care se folosesc algoritmi Identificare a algoritmilor în activități cotidiene |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|---|--|
| <ul style="list-style-type: none"> identificarea elementelor grafice utilizate pentru descrierea algoritmilor prin scheme logice interpretarea acțiunilor și operațiilor algoritmice descrise în limbaj uman prin scheme logice | <p>Metode de reprezentare a algoritmilor</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Metode de reprezentare ale algoritmilor: <ul style="list-style-type: none"> Limbajul uman, Schemele logice, Pseudocodul. Capcanele descrierii algoritmilor Descrierea algoritmilor cu ajutorul limbajelor de programare | <p>Exerciții de:</p> <ul style="list-style-type: none"> Interpretare prin scheme logice a algoritmilor descriși în limbaj uman Analiză și interpretare a algoritmilor descriși cu ajutorul schemelor logice Elaborare a programelor într-un limbaj de programare, în baza algoritmilor descriși prin scheme logice Testare a rezultatelor generate de programele elaborate |
| <ul style="list-style-type: none"> Descrierea algoritmilor de rezolvare a problemelor în modurile acceptate Elaborarea programelor care vor implementa algoritmi elementari | <p>Algoritmi elementari</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Cel mai mare divizor comun (Algoritmul Euclid), Ciurul lui Eratostene (numere prime), Algoritmul Babilonean pentru extragerea rădăcinii pătrate, Numere Fibonacci | <p>Exerciții de:</p> <ul style="list-style-type: none"> Interpretare a algoritmilor descriși în limbaj uman Implementare a algoritmilor într-un limbaj de programare Testare a corectitudinii implementării algoritmilor <p>Probleme:</p> <ul style="list-style-type: none"> Rezolvare de probleme în care este necesar de utilizat algoritmi în studiu <p>Studiu de caz:</p> <ul style="list-style-type: none"> Algoritmi Greciei Antice |
| Unitatea 2: Algoritmi pe tablouri | | |
| <ul style="list-style-type: none"> recunoașterea principalelor tipuri de structuri de date selectarea structurilor de date adecvate problemelor rezolvate efectuarea operațiilor definite pe structurile de date | <p>Căutare și numărare în tablouri</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Parcurgerea tablourilor liniare și bidimensionale Căutarea elementelor cu proprietățile date Căutarea secvențelor cu proprietățile date Numărarea elementelor cu proprietățile date la parcurgere Verificarea proprietăților elementelor | <p>Exerciții de:</p> <ul style="list-style-type: none"> Definire a structurilor de date cu proprietăți prestabilite Operare cu structurile de date Modelare a parcurgerii structurilor de date liniare Modelare a parcurgerii structurilor de date 2D <p>Probleme:</p> <ul style="list-style-type: none"> pentru determinarea minimului / maximului |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|--|---|
| | <ul style="list-style-type: none"> Incrementarea și resetarea contorului | <ul style="list-style-type: none"> pentru interschimbări a elementelor structurilor pentru verificarea "simetriei" structurilor de numărare a elementelor prime / pozitive / divizibile cu X, etc. |
| <ul style="list-style-type: none"> descrierea procesului de ordonare motivarea necesității de utilizare a sortării în viața cotidiană descrierea algoritmilor de sortare implementarea algoritmului de ordonare în problemele informatice | <p>Ordonarea elementelor în tablou. Bubblesort</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Rolul sortării în optimizarea algoritmilor Sortarea prin metoda bulelor <ul style="list-style-type: none"> Verificarea ordinii elementelor vecine Interschimbarea elementelor Repetarea Inversarea ordinii de sortare | <p>Exerciții de:</p> <ul style="list-style-type: none"> Verificare a ordonării elementelor unei structuri de date (tablou, șir de caractere) Interschimbare a elementelor dintr-o structură de date Ordonare a elementelor unei structuri de date. <p>Probleme:</p> <ul style="list-style-type: none"> determinarea ordinii elementelor accesul rapid la elementele structurilor cu valori extreme (primele K cele mai mari / mici elemente) |
| <ul style="list-style-type: none"> descrierea algoritmilor alternativi de ordonare motivarea modului specific de ordonare a șirurilor de caractere descrierea algoritmului de sortare prin selecție descrierea algoritmului de sortare prin inserție implementarea algoritmilor de ordonare în | <p>Alți algoritmi de ordonare</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Sortarea prin metoda selecției <ul style="list-style-type: none"> Căutări de maximum / minimum Memorarea indicelui Interschimbarea elementelor Sortarea prin inserție <ul style="list-style-type: none"> Inserarea unui element arbitrar într-un subșir sortat Deplasarea elementelor Cum sortăm șiruri de caractere? <ul style="list-style-type: none"> Codurile caracterelor | <p>Exerciții de:</p> <ul style="list-style-type: none"> modelare a ordonării elementelor unei structuri de date (tablou, șir de caractere) prin metoda selecției Interschimbare a perechilor de elemente poziționate arbitrar dintr-o structură de date Extindere a secvențelor sortate prin inserarea în acestea a elementelor noi. <p>Probleme:</p> <ul style="list-style-type: none"> Ordonare prin selecție a elementelor unei structuri Ordonare lexicografică Comparare a eficienței algoritmilor |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|---|---|--|
| problemele informatice | <ul style="list-style-type: none"> ○ Ordonarea lexicografică | |
| Unitatea 3: Metode de programare | | |
| <ul style="list-style-type: none"> • descrierea principiului de rezolvare a problemelor informatice prin metoda trierii • identificarea mulțimii tuturor soluțiilor posibile a problemei informatice • implementarea algoritmilor de forță brută pentru rezolvarea problemelor informatice | <p>Ce înseamnă "forța brută"</p> <p>Subiecte:</p> <ul style="list-style-type: none"> • Spațiul soluțiilor posibile pentru algoritmi 1D • Spațiul soluțiilor posibile pentru algoritmi 2D • Selectarea soluției optimale • Limite de aplicare ale metodei | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Căutare și numărare a elementelor de valoare dată în tablouri 1D • Numărare a punctelor planului cu proprietăți date • Generare a submulțimilor unei mulțimi <p>Probleme de:</p> <ul style="list-style-type: none"> • Identificare a soluției optime pentru probleme discrete: <ul style="list-style-type: none"> ○ Sumelor echilibrate într-un tablou liniar ○ Problema rucsacului ○ Problema intersecției cercurilor |
| <ul style="list-style-type: none"> • identificarea formulelor de reducere a dimensiunii pentru probleme elementare • determinarea cazurilor elementare pentru calculul recursiv • elaborarea programelor care vor conține subprograme recursive | <p>Recursia</p> <p>Subiecte:</p> <ul style="list-style-type: none"> • Recursia. Algoritmi elementari și nu numai. • Ce este recursia <ul style="list-style-type: none"> ○ Reducerea dimensiunii problemelor ○ Cazuri elementare ○ Regula de consistență. • Algoritmi recursivi elementari <ul style="list-style-type: none"> ○ Calcularea puterii ○ Factorialul ○ Combinări ○ Evaluarea expresiilor | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Modelare a formulelor recurente și cazurilor elementare • Verificare a regulii de consistență • Simulare a calculului recursiv • Elaborare a funcțiilor recursive <p>Probleme de:</p> <ul style="list-style-type: none"> • Calcul recursiv al sumelor și produselor • Calcul recursiv a elementelor șirurilor de numere celebre • Calcul recursiv combinatoric • Evaluare recursivă a secvențelor de caractere |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|--|--|
| <ul style="list-style-type: none"> descrierea algoritmului de propagare a undei implementarea algoritmului într-un limbaj de programare procedural identificarea situațiilor reale, care pot fi modelate folosind algoritmul Lee | <p>Algoritmul de propagare a undei</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Algoritmul Lee <ul style="list-style-type: none"> Descriere Modelare Structuri de date | <p>Exerciții de:</p> <ul style="list-style-type: none"> Modelare a algoritmului Lee Analiză a principiului de extindere a undei <p>Probleme:</p> <ul style="list-style-type: none"> De rezolvare a proceselor cotidiene cu ajutorul algoritmului Lee <ul style="list-style-type: none"> Problema celui mai scurt drum în labirint Problema fotografiei |
| <ul style="list-style-type: none"> descrierea tehnicii Divide et Impera elaborarea funcțiilor pentru sortările rapide în baza strategiei Divide et Impera cunoașterea problemei Turnurilor din Hanoi identificarea problemelor rezolvabile prin tehnica Divide et Impera | <p>Tehnici recursive. Divide et impera</p> <ul style="list-style-type: none"> Tehnica ”Împarte și Stăpânește” <ul style="list-style-type: none"> De unde provine denumirea? Modelarea recursivă Etapa de divizare Etapa de asamblare Sortări recursive <ul style="list-style-type: none"> QuickSort MergeSort Probleme celebre: Turnurile din Hanoi <ul style="list-style-type: none"> Istoria problemei Cum rezolvăm? | <p>Exerciții de:</p> <ul style="list-style-type: none"> Modelare a ordonării prin divizare (mai mic – mai mare) Modelare a ordonării prin divizare (în părți egale) Formulare a condițiilor de continuare și oprire a proceselor de divizare <p>Probleme:</p> <ul style="list-style-type: none"> Implementarea algoritmului de sortare rapidă QuickSort Implementarea algoritmului de sortare rapidă MergeSort Probleme de ordonare a volumelor mari de date |
| <ul style="list-style-type: none"> descrierea componentelor metodei reluării identificarea spațiului de soluții a problemei | <p>Reluarea. Labirinturi și probleme pe tabla de șah</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Tehnica reluării: <ul style="list-style-type: none"> Componentele metodei “mișcări de extindere” Mișcări de revenire Reluarea în labirint | <p>Exerciții de:</p> <ul style="list-style-type: none"> Modelare interactivă a căutării ieșirii din labirint Formulare a condițiilor de continuare și revenire pentru labirinturi 2 și 3 dimensionale |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|---|---|
| <ul style="list-style-type: none"> formularea condițiilor de continuare (mișcări înainte) și revenire (mișcări înapoi) pentru soluțiile informatice ale problemelor | <ul style="list-style-type: none"> Determinarea tuturor traseelor posibile Determinarea traseelor care evită zone predefinite ale labirintului Reluarea pe tabla de șah: <ul style="list-style-type: none"> Parcurea tablei de șah cu mișcări ale calului Problema celor N dame Attila și regele | <p>Problema:</p> <ul style="list-style-type: none"> Implementarea tehnicii reluării într-un algoritm recursiv Rezolvarea problemei labirintului în condiții standard Rezolvarea problemei labirintului cu restricții adiționale Rezolvarea informatică a problemei parcurgerii tablei de șah cu mișcări ale calului Rezolvarea informatică a problemei amplasării pe tabla de șah a N dame Rezolvarea informatică a problemei Attila și Regele |
| Unitatea 4: Algoritmi pe grafuri | | |
| <ul style="list-style-type: none"> identificarea elementelor unui graf definirea noțiunilor aferente grafurilor determinarea tipului unui graf descrierea grafului folosind diverse metode de reprezentare elaborarea programelor pentru tranziția între formele de reprezentare a grafului | <p>Grafuri. Noțiuni și reprezentări</p> <p>Subiecte:</p> <ul style="list-style-type: none"> Noțiuni: <ul style="list-style-type: none"> Vârf Muchie Drum Ciclu Tipuri <ul style="list-style-type: none"> Graf orientat Graf neorientat Graf planar Graf complet Metode de reprezentare în calculator: <ul style="list-style-type: none"> Matrice de adiacență Listă de muchii | <p>Exerciții de:</p> <ul style="list-style-type: none"> Identificare a elementelor unui graf Determinare a tipului unui graf Formare a structurilor de date pentru reprezentarea grafului <p>Probleme de:</p> <ul style="list-style-type: none"> Identificare a nodurilor "centrale" (conectate direct la cele mai multe noduri) Identificare a nodurilor izolate Transformare a modului de reprezentare a grafului |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|---|--|
| <ul style="list-style-type: none"> • explicarea noțiunii de parcurgere a nodurilor unui graf • definirea noțiunilor de componentă conexă, tare conexă • identificarea situațiilor și problemelor care impun utilizarea algoritmilor de parcurgere pentru rezolvarea lor în informatică. | <p>Parcurgeri ale grafului</p> <p>Subiecte:</p> <ul style="list-style-type: none"> • Algoritmi de parcurgere <ul style="list-style-type: none"> ○ Parcurgerea în adâncime – DFS ○ Parcurgerea în lățime - BFS. • Conexitate <ul style="list-style-type: none"> ○ Componente conexe ○ Componente tare conexe | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Parcurgere a grafului în lățime • Parcurgere în adâncime a grafului • Identificare a componentelor conexe în grafuri neorientate • Identificare a componentelor tare conexe în grafuri orientate • <p>Probleme de:</p> <ul style="list-style-type: none"> • Determinare a existenței conexiunii între noduri date • Căutare a componentelor conexe (problema grupărilor politice) |
| <ul style="list-style-type: none"> • identificarea vizuală a grafurilor de tip arbore • descrierea noțiunii arbore de cost minim • construirea arborilor de cost minim | <p>Arbori de cost minim</p> <p>Subiecte:</p> <ul style="list-style-type: none"> • Arbori • Arbori de cost minim. <ul style="list-style-type: none"> ○ Algoritmul Prim, ○ Algoritmul Kruskal | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Construire interactivă a arborilor de cost minim (Kruskal) • Construire interactivă a arborilor de cost minim (Prim) <p>Probleme de:</p> <ul style="list-style-type: none"> • Calcul a greutatei minime a unui arbore parțial (Kruskal) • Reconstrucție a căii ferate (Prim) |
| <ul style="list-style-type: none"> • explicarea noțiunii de drum minim • descrierea algoritmilor de determinare a distanței minime • implementarea algoritmilor de determinare a | <p>Drumuri minime în graf</p> <p>Subiecte:</p> <ul style="list-style-type: none"> • Noțiuni • Drum minim <ul style="list-style-type: none"> ○ Între 2 vârfuri ○ De la un vârf la toate celelalte ○ Între toate vârfurile • Algoritmul <ul style="list-style-type: none"> ○ Dijkstra, | <p>Exerciții de:</p> <ul style="list-style-type: none"> • Determinare a drumului minimal (Dijkstra) interactiv • Determinare a drumului minimal (Floyd) <p>Probleme de:</p> <ul style="list-style-type: none"> • Generare a tabelului de distanțe |

| Unități de competență | Module / Conținuturi / Noțiuni-cheie | Activități de învățare și produse școlare recomandate |
|--|---|---|
| distanței minime | <ul style="list-style-type: none"> ○ Floyd | <ul style="list-style-type: none"> ● Restabilire a traiectoriei distanței minime ● Determinare a tuturor distanțelor minime ● Amplasare optimală |
| Unitatea 5: Algoritmi euristici | | |
| <ul style="list-style-type: none"> ● identificarea spațiului de soluții a problemei ● formularea criteriului de acceptanță a elementelor ● stabilirea restricțiilor ● formarea soluției ● identificarea problemelor rezolvabile prin algoritmi Greedy | <p>Tehnica Greedy</p> <p>Subiecte:</p> <ul style="list-style-type: none"> ● Soluții rapide versus soluții exacte ● Etapele tehnicii Greedy <ul style="list-style-type: none"> ○ Spațiul de soluții ○ Criteriul de acceptanță ○ Coeficientul de acceptanță ○ Restricții | <p>Exerciții de:</p> <ul style="list-style-type: none"> ● Identificare a spațiului de soluții ● Determinare a restricțiilor problemei ● Îmbunătățire a criteriului de acceptanță <p>Probleme de:</p> <ul style="list-style-type: none"> ● Maximizare / minimizare ● Modelare ● Grafuri |
| <ul style="list-style-type: none"> ● Descrierea spațiului de joc ● Selectarea structurilor inițiale ● Modelarea soluțiilor cu proprietăți specifice | <p>Jocul "Viața"</p> <p>Subiecte:</p> <ul style="list-style-type: none"> ● Descriere reguli ● Modelare stări ● Formule de recurență ● Analiza stare curentă | <p>Exerciții de:</p> <ul style="list-style-type: none"> ● Modelare a etapelor de joc ● Selectare a configurației inițiale ● Implementare a algoritmului <p>Probleme de:</p> <ul style="list-style-type: none"> ● Generare a figurilor <ul style="list-style-type: none"> ○ Stabile ○ Periodice ○ Dinamice ○ Tren ○ Răpitori |

Sugestii pentru evaluare

Axarea procesului de învățare-predare-evaluare pe competențe generează o structură continuă a evaluării.

Evaluarea:

- este un proces permanent, dimensiunea esențială a procesului educațional și o practică efectivă;
- stimulează învățarea, formarea și dezvoltarea competențelor;
- se axează pe necesitatea de a compara pregătirea elevilor cu competențele specifice, cu unitățile de competențe (sub-competențele) și cu obiectivele (operaționale) fiecărei lecții;
- se fundamentează pe standarde educaționale de stat – standarde de competență – orientate spre ceea ce va ști, ce va ști să facă și cum va fi elevul la finalizarea cursului;
- implică utilizarea unei varietăți de metode (tradiționale și moderne);
- este un proces reglator, care determină calitatea activităților școlare;
- trebuie să-i ghideze pe elevi spre o autoapreciere corectă și spre o îmbunătățire continuă a performanțelor școlare.

Scopurile de bază a evaluării:

- motivarea elevilor și realizarea feed-back-ului;
- stimularea la elevi a efortului de autoevaluare formativă (autoevaluarea formativă reprezintă procesul prin care elevul însuși este pus să judece calitatea lucrului său în raport cu obiectivele definite și cu criteriile de apreciere propuse);
- formarea deprinderilor de evaluare reciprocă (evaluarea reciprocă constituie un proces de interacțiune evaluativă orientat spre emiterea unor judecăți de valoare în baza unor criterii prestabilite);
- evidențierea succesului, realizând astfel principiul centrării pe personalitatea celui evaluat.

În procesul educațional, profesorul va pune în aplicare:

- a) evaluarea inițială, realizând funcția prognostică;
- b) evaluarea curentă, realizând funcția formativă;
- c) evaluarea finală (sumativă), realizând funcția diagnostică.

Valoarea **evaluării formative** constă în formarea permanentă și continuă a competențelor la elevi reflectate în standardele educaționale. Sarcinile de evaluare formativă urmează să fie separate pe grade de dificultate, pentru a permite o individualizare a evaluării și o motivare suplimentară a elevilor evaluați, fapt ce va permite valorificarea la maximum a potențialului intelectual al fiecărui elev.

Elaborarea itemilor pentru evaluare va fi realizată în contextul taxonomiilor corespunzătoare.

Evaluarea sumativă se va realiza la sfârșitul fiecărui modul, semestru și an școlar. În calitate de elemente componente ale instrumentelor de evaluare se recomandă utilizarea testelor asistate de calculator și a itemilor de tip problemă pentru rezolvare la calculator. În ansamblu, evaluările sumative realizate vor demonstra dacă la finele modulului sunt dobândite achizițiile determinate de unitățile de competență preconizate în curriculum pentru compartimentul respectiv.

Metodele folosite pentru evaluarea continuă presupun chestionarea orală sau scrisă, metode interactive: studii de caz, lucrări practice, proiecte, testări interactive asistate de calculator.

Pentru realizarea cu succes a unei activități practice, elevii trebuie să fie avizați de către profesor asupra:

- tematicii lucrărilor;
- modului în care ele vor fi evaluate (baremele/grilele/criteriile de notare);
- condițiilor care le sunt oferite pentru a realiza aceste activități.

Realizarea evaluării continue permite o apreciere obiectivă a cunoștințelor și competențelor elevilor, precum și a progreselor înregistrate de aceștia.

Probele de evaluare utilizate la clasă vor conține itemi și sarcini prin intermediul cărora se vor evalua, prioritar, nu doar cunoștințe și capacități separate, ci formarea de competențe.

Prin sarcini didactice de divers nivel de complexitate, profesorul orientează și dirijează activitatea de studiere a elevilor, evidențiază ce și cum trebuie să învețe, formându-le un stil de muncă intelectual. Evaluarea realizată astfel evită caracterul de “surpriză” al rezultatelor. Ea nu se efectuează în scop de “sanționare”, ci permite autoevaluarea rezultatelor obținute, transformând elevul în subiect al propriei formări.

Toate aceste metode permit profesorului să dispună de puncte de reper și să adune informații asupra desfășurării activității elevului, oferindu-i astfel elevului posibilitatea de a demonstra ceea ce știe într-o varietate de contexte și situații.

Pe baza acestor informații profesorul își fundamentează judecata de valoare într-o apreciere obiectivă a achizițiilor elevilor și a progreselor înregistrate. Evaluările realizate la finele anului de învățământ vor demonstra achiziția sub competențelor indicate în curriculumul respectiv.

În contextul principiilor evaluării prioritare și dominante în procesul educațional, este important raportul evaluarea curentă – evaluarea formativă. Succesul lecției rezultă din atingerea obiectivelor preconizate. În acest sens, secvența **Evaluare** este obligatorie pentru fiecare lecție și în cadrul acestei secvențe se va evalua nivelul de realizare a obiectivelor lecției.

Evaluările realizate vor include în mod obligatoriu și itemi rezolvarea cărora necesită conexiuni interdisciplinare. Este important ca fiecare elev, profesor și părinte/tutore să conștientizeze că evaluarea în orice circumstanțe trebuie să fie obiectivă.

Bibliografie

Resurse bibliografice

1. Cerchez Emanuela, Șerban Marinel. Limbalul C / C++ pentru liceu (vol I), Polirom, Iași, 2007
2. S. Corlat, A. Gremalschi Grafuri. metodologia predării în cadrul instruirii de performanță la disciplinele Matematică & Informatică Universitatea de Stat din Tiraspol, 2013.
3. Sedgewick, Robert. Algorithms in C++. Parts (I - IV), Fundamentals, Data structures, sorting, searching. Addison – Wesley, 1999
4. Sedgewick, Robert. Algorithms in C++. Part V, Graph algorithms. Addison – Wesley, 1999
5. Ryuhei Uehara First Course in Algorithms Through Puzzles, Springer, 2019

Documente regulatorii

1. Cadrul de referință al Curriculumului Național, Chișinău, 2017
2. Repere conceptuale privind elaborarea curriculumului la disciplinele opționale, Chișinău, 2017