

Unitatea de învățământ:

Profesor:

Disciplina: **Informatică (CS)**

Clasa a IX-a, **2h/săpt.**

Specializarea **Matematică-Informatică**

Proiectarea unităților de învățare

Propunere

Anul școlar 2026-2027

UNITATEA DE ÎNVĂȚARE *Principii de elaborare a unui program*

NR. DE ORE ALOCATE – 8

(2 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)

CONȚINUTURI	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Caracteristici ale gândirii computaționale, etapele elaborării unui program, reprezentarea algoritmilor	1.3, 2.3, 3.3, 4.3, 5.3, 6.3.	- Reamintire - noțiunea de algoritm și proprietățile acestuia. - Identificarea datelor cu care lucrează algoritmi și aplicarea modelului Intrare-Prelucrare-Ieșire (modelul IPI) pe exemple concrete și învățare prin descoperire. - Enumerarea etapelor de elaborare a unui program și explicarea rolului fiecărei. - Enumerarea modurilor de reprezentare ale unui algoritm. - Aplicare – învățare prin descoperire – (pentru exersare model IPI), studiu de caz (pentru moduri de reprezentare a algoritmilor), activitate în echipă (pentru reprezentare în limbaj pseudocod).	- Manual - Probleme rezolvate, exemple - Aplicații practice	- Aplicații practice - Observare sistematică

CONȚINUTURI	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVAȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Introducere în lb Python. Comunicarea cu programul, urmărirea execuției algoritmilor;		- Introducere în limbajul Python – vocabular, tipuri de date, citiri/scrieri, operatori – analiză exemple și probleme rezolvate - Comunicare cu programul – aplicare într-un mediu de dezvoltare integrat (ex PyCharm) - Prezentarea structurilor de control: if, for, while – analiză probleme rezolvate - Implementare în limbajul Python a unor algoritmi simpli, urmărirea execuției algoritmilor și învățare prin descoperire.	- Manual - Probleme rezolvate, exemple - Aplicații practice - Calculator	
Eficiența algoritmilor. Proiectarea modulară a algoritmilor.		- Analiza eficienței algoritmilor (timp și memorie) și introducerea notației O, pe o problemă rezolvată. - Analiza proiectării modulare a algoritmilor pe un exemplu concret. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Aplicații practice - Fișă de lucru - Calculator	
Recapitulare/evaluare				Aplicație practică

UNITATEA DE ÎNVĂȚARE Prelucrări ale datelor numerice.**NR. DE ORE ALOCATE – 7****(2 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)**

CONȚINUTURI	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Operații cu cifrele unui număr natural. Divizori, multipli, cmmdc, descompunerea în factori primi. Transformarea unui număr natural dintr-o bază de numerație în altă bază de numerație.	1.2 2.2 3.2 4.2 5.2 6.2.	- Analiza prelucrării cifrelor unui număr natural – probleme rezolvate: determinarea ultimei cifre, numărarea cifrelor, adăugarea unei cifre la dreapta/stanga unui număr natural. - Analiza determinării divizorilor/multiplilor unui număr natural, determinarea cmmdc, descompunerea unui număr în factori primi – probleme rezolvate. - Analiza transformării unui număr natural dintr-o bază de numerație într-o altă bază de numerație – probleme rezolvate. - Aplicații – algoritmi elementari.	- Manual - Probleme rezolvate, exemple - Aplicații practice - Fișă de lucru - Calculator	- Aplicații practice - Observare sistematică - Miniproiect de grup - Portofoliu digital – include rezolvarea fișelor de lucru - Evaluarea/Autoevaluarea comportamentului și a activității elevului
Recapitulare/evaluare				Test de evaluare

UNITATEA DE ÎNVĂȚARE *Tehnici de organizare a codului în limbaj de programare.***NR. DE ORE ALOCATE – 12***(3 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)*

CONȚINUTURI (detalii)	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Subprograme: Rol, caracteristici	1.5, 2.3, 2.5, 3.3, 3.5, 4.3, 4.5, 5.3, 5.5, 6.3, 6.5	- Analiza descompunerii unei probleme în subprobleme (subprograme), reprezentare în limbaj pseudocod – studii de caz. - Analiza fluxului de execuție. - Aplicații practice. - Identificare de concluzii și caracteristici.	- Manual - Aplicații practice	- Aplicații practice - Observare sistematică - Miniproiect de grup - Portofoliu digital – include rezolvarea fișelor de lucru - Evaluarea/Autoevaluarea comportamentului și a activității elevului
Subprograme: Concepte de bază – sintaxă definiție, apel, returnarea rezultatelor, mecanism de execuție, parametri.		- Analiza sintaxei: definiție și apel. - Probleme rezolvate – analiza conceptelor – definiție, apel, returnarea rezultatelor, mecanism de transmitere al parametrilor, mecanism de executare. - Tipuri de parametri efectivi – exemple. - Implementarea aceleiași probleme în două variante – una cu subprograme și alta fără subprograme. - Activitate în echipă – completarea unui program rezolvat cu alte subprograme. - Identificare avantajele/dezavantajele utilizării subprogramelor. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Aplicații practice - Calculator	
Subprograme: Variabile locale, variabile globale.		- Identificarea variabilelor locale/variabilelor globale pe exemple date. - Identificarea regulilor de acces pe exemple date. - Rescrierea unui program rezolvat, dar cu variabile globale. - Activitate în echipă – rescrierea unui program rezolvat, dar cu variabile globale. - Analiza comparativă: utilizarea parametrilor vs variabile globale.	- Manual - Probleme rezolvate, exemple - Aplicații practice - Calculator	

CONȚINUTURI (detalii)	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVAȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Subprograme: Funcții predefinite în Python.		- Identificarea funcțiilor predefinite matematice – aplicații pe exemple. - Identificare tipuri de conversii: implicite, explicite - Identificare funcții de conversie – aplicații pe exemple. - Analiza problemă rezolvată. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Fișă de lucru - Calculator	
Introducere în POO: Noțiuni de bază necesare utilizării unor clase predefinite: clasă, membri ai clasei, obiecte.		- Analiza conceptelor de bază POO pe exemple concrete – clasă, atribut, metode, obiect. - Paralelă programare procedurală vs POO – exemplu din viața reală, implementare în limbajul Python. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Fișă de lucru - Calculator	
Introducere în POO: Instanțierea unei clase predefinite, acces la membrii unui obiect. Bibliotecă.		- Analiza instanțierii claselor predefinite: int, float, str, list, bool.	- Manual - Exemple	
Recapitulare/evaluare				Test de evaluare

UNITATEA DE ÎNVĂȚARE Fișiere text.**NR. DE ORE ALOCATE – 6****(2 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)**

CONȚINUTURI (detalii)	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Caracteristici, principii de lucru; Implementarea fișierelor text în limbajul Python, clasa predefinită TextIOWrapper; Alte metode pentru prelucrarea fișierelor text;	1.4, 2.4, 3.4, 4.4, 5.4, 6.4	- Analiza caracteristicilor unui fișier text. - Enumerarea etapelor de lucru: deschidere, prelucrare (citire, scriere), închidere. - Explicarea noțiunii de sfârșit de fișier. - Utilizarea instrucțiunii “with open”. - Completarea unui program lacunar într-un program care determină divizorii unui număr natural. - Compararea timpului de executare a unui program care citește datele dintr-n fișier în raport cu citirea de la tastatură. - Analiza cazurilor limită în lucru cu fișiere text: fișier inexistent, fișier fără conținut. - Alte metode utile în lucrul cu fișierele text: tell() și seek(). - Proiectarea unei aplicații cu valențe practice (generarea de numere aleatoare). - Aplicații practice.	- Manual - Exemple de cod cu rezultate de execuție - Investigație pe două programe identice cu citiri din surse diferite - Probleme rezolvate - Fișă de lucru	- Aplicații practice - Observare sistematică - Investigație de grup - Autoevaluare pe un cod lacunar - Miniproiect de grup - Portofoliu digital – include rezolvarea fișei de lucru
Recapitulare/evaluare				Test de evaluare.

UNITATEA DE ÎNVĂȚARE Biblioteca Tkinter pentru interfețe grafice.

NR. DE ORE ALOCATE – 12

(3 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)

CONȚINUTURI (detalii)	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Introducere – clase, caracteristici, avantaje/dezavantaje; Funcții și metode de bază pentru interfețe grafice simple;	1.4, 2.4, 3.4, 4.4, 5.4, 6.4	- Analiza avantajelor/dezavantajelor utilizării interfețelor grafice. - Prezentarea primului program cu o fereastră. - Identificarea, pe câte un exemplu de program, a metodelor/funțiilor asociate componentelor de bază într-o interfață grafică simplă. - Aplicații practice.	- Manual - Exemple de cod cu rezultate de execuție - Probleme rezolvate - Calculatorul	- Aplicații practice - Observare sistematică
Organizarea elementelor în fereastră: pack, grid, place; Variabile Tkinter: StringVar, IntVar; Alte clase și metode pentru realizarea interfețelor grafice;		- Prezentarea, pe câte un exemple de program, a diferitelor sisteme de poziționare a obiectelor grafice într-o fereastră. - Prezentarea, pe câte un exemplu de program, a avariabilelor StringVar, IntVar. - Prezentarea, pe câte un exemplu de program, a evenimentelor: reacții la apăsarea de taste/click mouse. - Compararea unei aplicații cu interfață grafică în raport cu aceeași aplicație cu interfață text – minicalculator. - Analiza de probleme rezolvate cu interfețe grafice. - Aplicații practice.	- Manual - Exemple de cod cu rezultate de execuție - Probleme rezolvate - Fișă de lucru - Calculatorul	- Miniproiect de grup - Portofoliu digital – include rezolvarea fișei de lucru - Evaluarea/Autoevaluarea comportamentului și a activității elevului
Recapitulare/evaluare				Aplicație practică.

UNITATEA DE ÎNVĂȚARE Liste în limbajul Python.**NR. DE ORE ALOCATE – 8****(2 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)**

CONȚINUTURI (detalii)	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Modelul conceptual liniar - listă Caracteristicile unei liste; cazuri particulare după tipul de acces; Parcurgerea elementelor, prelucrarea datelor dintr-o listă, cu și fără memorare.	1.1, 1.4, 1.5, 2.1, 2.4, 2.5,	- Recunoșterea modelului conceptual de tip listă și caracteristicilor unei liste, exemple din viața reală. - Studii de caz din viața reală – liste de frecvență, liste cu sau fără memorarea datelor. - Exemplificarea noțiunilor de tip stivă/coada – prelucrare, operații. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Fișă de lucru	- Aplicații practice
Clasa list din Python: Caracteristici. Operatori specifici. Metode pentru operații de bază. Alte metode adecvate pentru prelucrarea listelor.	3.1, 3.4, 3.5, 4.1, 4.4, 4.5, 5.1, 5.4, 5.5, 6.1, 6.4, 6.5	- Prezentarea caracteristicilor unei liste din lb Python: sintaxă, tipul elementelor, indexare, parcurgere – exemple. - Prezentarea operatorilor specifici unei liste – exemple. - Analiza metodelor din clasa list – exemple, probleme rezolvate. - Analiza funcțiilor predefinite specifice listelor – exemple, probleme rezolvate. - Analiza unei aplicații practice în care sunt utilizate liste, stive, cozi. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Fișă de lucru - Calculator	- Observare sistematică - Miniproiect de grup - Portofoliu digital – include rezolvarea fișei de lucru
Recapitulare/evaluare				Test de evaluare

UNITATEA DE ÎNVĂȚARE Prelucrări cu liste.**NR. DE ORE ALOCATE – 10****(2 ore la dispoziția profesorului – utilizate ca o completare și o adaptare a parcurgerii programei școlare la contextul clasei)**

CONȚINUTURI (detalii)	COMPETENȚE SPECIFICE	ACTIVITĂȚI DE ÎNVĂȚARE	RESURSE	EVALUARE
<i>Ce?</i>	<i>De ce?</i>	<i>Cum?</i>	<i>Cu ce?</i>	<i>Cerințe</i>
Metode de generare sistematică a elementelor unei liste. Generarea unor secvențe de valori și aplicarea algoritmilor de bază.	1.2 2.2 3.2 4.2 5.2 6.2	- Explicarea modului de formare a unei liste de valori ai cărei termeni respectă o anumită regulă – exemple. - Aplicarea generării elementelor unei astfel de liste – probleme rezolvate. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Fișă de lucru - Calculator	- Aplicații practice
Metode de sortare: Sortarea prin selecție; Sortarea prin metoda bulelor; Sortarea cu liste de frecvență;		- Explicarea necesității operației de sortare în programare – exemple. - Prezentarea caracteristicilor metodei de sortare prin selecția minimului – exemplu de execuție pas cu pas, cod Python, problemă rezolvată. - Prezentarea caracteristicilor metodei de sortare prin metoda bulelor – exemplu de execuție pas cu pas, cod Python, problemă rezolvată. - Analiza comparativă a celor două metode de sortare (selecția minimului și metoda bulelor) dpdv: complexității, număr de interschimbări, stabilitate. - Prezentarea caracteristicilor metodei de sortare cu liste de frecvență – exemplu de execuție pas cu pas, cod Python, studii de caz. - Aplicații practice.	- Manual - Probleme rezolvate, exemple - Fișă de lucru - Calculator	- Observare sistematică - Miniproiect de grup - Portofoliu digital – include rezolvarea fișei de lucru - Evaluarea/Autoevaluarea comportamentului și a activității elevului
Recapitulare/evaluare				Test de evaluare