

3. PLANUL MODULULUI ȘI CONȚINUTUL

Introducere în codare și programare



Project No. 2023-2-IT03-KA220-YOU-000179130

Version 1.0

1. PLANUL MODULULUI

Planul modulului	
Titlul modulului	Introducere în programare
Durata	3 ore (total)
Obiective	Până la sfârșitul acestui modul, participanții vor putea: • Să înțeleagă conceptele de bază ale programării și codării. • Să recunoască diferite limbaje de programare și utilizările acestora. • Să scrie programe simple folosind structuri fundamentale de programare (de exemplu: variabile, bucle, condiții). • Să dezvolte abilități de rezolvare a problemelor prin exerciții de programare. • Să se familiarizeze cu structura generală a unui program. • Să înțeleagă importanța algoritmilor și a gândirii logice în programare. • Să învețe cum să depaneze erori simple de cod. • Să exploreze aplicațiile reale ale programării în diverse industrii. • Să promoveze gândirea computațională și creativitatea prin activități de programare.
Planul lecției	Bazele Designului și Dezvoltării Digitale Modul de formare pentru tineri cu dizabilități de auz

Planul modului	
	Durata: ~60 minute (3 topicuri × 20 minute) 1. Ce sunt codarea și programarea? (20 minute) Obiectiv: • Să înțeleagă ce sunt codarea și programarea și importanța acestora în lumea digitală de astăzi. Puncte principale de acoperit: • Definiția codării și a programării • Diferența dintre limbajele de programare și exemple (de ex., Python, JavaScript) • Exemple reale ale aplicațiilor programării (aplicații, site-uri web, jocuri) Activitate: • Discuții și brainstorming despre locurile în care elevii întâlnesc programarea în viața de zi cu zi • Vizionarea unui scurt videoclip care explică importanța programării în tehnologie Metodologie de predare: • Prelegere cu materiale vizuale (prezentări, videoclipuri) • Discuții interactive 2. Concepte de bază și elementele fundamentale ale codării (20 minute) Obiectiv: • Introducerea conceptelor fundamentale de programare, precum variabilele, tipurile de date și structurile de control de bază. Puncte principale de acoperit: • Variabile și tipuri de date (numere, text) • Operatori de bază și expresii

Planul modului	
	• Flux de control simplu: instrucțiuni "if-else" și bucle ("loops") • Conceptul de algoritmi și pași pentru rezolvarea problemelor Activitate: • Exercițiu practic: realizarea de pseudocod sau diagrame de flux pentru sarcini de zi cu zi (de exemplu: pregătirea unui sandviș, pregătirea pentru școală) Metodologie de predare: • Demonstrație cu exemple live de codare • Exerciții ghidate și lucru în grup 3. Scrierea primului program (20 de minute) Obiectiv: • Să permită elevilor să scrie și să ruleze primul lor program simplu. Puncte principale de acoperit: • Introducere într-un mediu de programare prietenos pentru începători (de ex. Scratch, Code.org, Python IDLE) • Structura unui program simplu (instrucțiunea de afișare „print”, input, logică de bază) • Depanarea (debugging) erorilor comune Activitate: • Programare pas cu pas: elevii scriu un program care cere numele și afișează un mesaj de salut • Evaluare între colegi și exerciții de depanare

Planul modului	
	Metodologie de predare: • Demonstrație live de programare • Practică ghidată cu suport și feedback • Colaborare între colegi
Materiale necesare	1. Materiale vizuale și prezentări: • Prezentări tip slide-uri sau PowerPoint cu conceptele cheie • Videoclipuri scurte care explică programarea și codarea (opțional) 2. Calculatoare și software: • Calculatoare sau laptopuri pentru fiecare elev sau în perechi • Un mediu de programare prietenos pentru începători (de ex. Scratch, Python IDLE, Code.org, Blockly) 3. Fișe de lucru și materiale tipărite: • Șabloane de pseudocod sau diagrame de flux pentru activități • Fișe de lucru cu exerciții despre concepte de bază și rezolvarea problemelor • Fișe „cheat sheet” cu sintaxa și comenzile de bază în programare 4. Instrumente interactive: • Platforme online de programare sau site-uri pentru exersare (de ex. Code.org, Khan Academy, Trinket) • Exerciții de depanare sau provocări de tip joc 5. Materiale suplimentare: • Tablă albă și markere pentru explicații și discuții • Exemple tipărite de programe simple și algoritmi • Caiete sau foi pentru notițe și brainstorming

2. CONȚINUTURI

Etapale lecției	Descriere	Conținut
Termeni și definiții	Digital Literacy	Alfabetizarea digitală înseamnă a avea cunoștințele și abilitățile necesare pentru a folosi instrumente digitale, cum ar fi software-ul de design și limbajele de programare, pentru a crea aplicații, site-uri web și conținut online. Aceasta include înțelegerea modului în care funcționează sistemele digitale și capacitatea de a construi și proiecta produse digitale simple. Termeni cheie: • Limbaj de programare (Programming Language) – Un limbaj folosit pentru a scrie instrucțiuni pe care un computer le poate înțelege. • Cod (Code) – Setul de instrucțiuni scrise într-un limbaj de programare. • Algoritm (Algorithm) – O serie de pași clari pentru rezolvarea unei probleme. • Variabilă (Variable) – O locație de stocare într-un program care păstrează date. • Tip de date (Data Type) – Tipul de informație (de ex. numere întregi, șiruri de text, numere zecimale) stocată într-o variabilă. • Funcție / Metodă (Function / Method) – Un bloc de cod care îndeplinește o sarcină specifică și poate fi reutilizat. • Bucă (Loop) – O secvență de instrucțiuni care se repetă până când este îndeplinită o condiție. • Condiție / If-Else (Condition / If-Else) – O instrucțiune care execută acțiuni diferite în funcție de faptul dacă o condiție este adevărată sau falsă. • Depanare (Debugging) – Procesul de identificare și corectare a erorilor din cod.

Etaplele lecției	Descriere	Conținut
		• Sintaxă (Syntax) – Regulile care definesc structura corectă a codului într-un limbaj de programare. • Intrare (Input) – Datele pe care un program le primește de la utilizator sau din altă sursă. • Ieșire (Output) – Datele produse de un program pentru utilizator. • IDE (Mediu de Dezvoltare Integrat) – O aplicație software care oferă instrumente pentru scrierea, testarea și depanarea codului. • Bytecode – Un cod intermediar care poate fi interpretat sau compilat. • Compilare (Compilation) – Procesul de traducere a codului într-un limbaj de mașină pe care computerul îl poate executa. • Limbaj de mașină (Machine Language) – Codul binar pe care procesorul unui computer îl înțelege direct. • Comentarii (Comments) – Note scrise în cod pentru a explica ce face programul; sunt ignorate de computer. • Eroare de sintaxă (Syntax Error) – O eroare cauzată de cod incorect care încalcă regulile limbajului. • Eroare logică (Logic Error) – O eroare în care programul rulează, dar produce rezultate incorecte..
Video	Youtube Links	1. " Ce este codarea?" – CrashCourse Computer Science https://www.youtube.com/watch?v=fikvFQoYZ3Q O explicație ușor de înțeles despre ce este programarea și de ce este importantă. 2. "Introducere în programare" – freeCodeCamp https://www.youtube.com/watch?v=8oRjP8yj_w8

Etapale lecției	Descriere	Conținut
		O prezentare generală, potrivită pentru începători, a fundamentelor programării. . 3. "Învăță programare într-o oră" – Tech with Tim https://www.youtube.com/watch?v=BoardJz_VRc O prezentare generală, potrivită pentru începători, a fundamentelor programării. . 4. "Ce este limbajul de programare?" – Computer Science Gaming https://www.youtube.com/watch?v=7sA8vfQWWvg O explicație scurtă despre limbajele de programare și modul în care funcționează.
Desfășurarea activității		1. Activitate de grup: Obiective: - Să încurajeze elevii să aplice concepte de bază ale programării, precum secvențe, condiții și bucle, prin proiectarea unui algoritm simplu, pas cu pas, pentru o sarcină din viața reală. Descrierea activității: - Împărțiți elevii în grupuri mici (3–4 elevi pe grup). - Fiecare grup alege sau primește o sarcină simplă din viața de zi cu zi (de exemplu: prepararea unui sandviș, spălatul pe dinți, pregătirea pentru școală). - Elevii lucrează împreună pentru a crea un algoritm detaliat, pas cu pas, sau o diagramă de tip flowchart care descrie sarcina, incluzând puncte de decizie (condiții de tip „dacă-atunci”) și pași repetitivi (bucle). - Fiecare grup își prezintă algoritmul în fața clasei, explicând cum funcționează, cu accent pe pașii logici și punctele de decizie.

Etapale lecției	Descriere	Conținut
		Materiale necesare: - Foi mari de hârtie sau table albe, markere sau șabloane tipărite pentru diagrame tip flowchart. - Fișe de lucru cu simboluri de bază pentru algoritmi și ghiduri pentru diagrame flowchart (opțional). Rezultate ale învățării: - Elevii vor demonstra înțelegerea logicii de programare și modul în care sarcinile din viața reală pot fi transformate în instrucțiuni asemănătoare codului. - Își vor dezvolta abilitățile de rezolvare a problemelor și de colaborare.
		2. Joc de rol: Obiective: - Să ajute elevii să înțeleagă cum să explice clar concepte de programare și să simuleze comunicarea din lumea reală în domeniul programării. De asemenea, consolidează înțelegerea pașilor de programare și a procesului de luare a deciziilor. Descrierea activității: - Elevii lucrează în perechi; un elev joacă rolul „Mentorului de programare”, iar celălalt rolul „Începătorului”, care vrea să învețe cum să realizeze o sarcină simplă (de exemplu: pregătirea unui sandviș, spălatul pe dinți). „Începătorul” îi cere „Mentorului de programare” să explice cum se realizează sarcina printr-un proces „de codare”, pas cu pas.

Etapale lecției	Descriere	Conținut
		• „Mentorul de programare” trebuie să descrie procesul folosind instrucțiuni simple și clare, incluzând puncte de decizie și repetiții, ca și cum ar învăța pe cineva să gândească precum un programator. • Se schimbă rolurile după finalizarea explicației. • După ce ambii au jucat ambele roluri, elevii discută despre importanța instrucțiunilor clare și a gândirii logice în programare. Materiale necesare: • Nu sunt necesare materiale, doar spațiu pentru jocul de rol. • Opțional: exemple de sarcini sau pași pentru a-i ajuta pe elevi să înceapă. Rezultate ale învățării: • Elevii vor exersa transformarea sarcinilor din viața reală în instrucțiuni logice, pas cu pas. • Își vor dezvolta abilitatea de a explica concepte tehnice clar și de a asculta activ.
		3. Scenarii din viața reală: Obiectiv: • Să îi ajute pe elevi să înțeleagă cum o rutină zilnică poate fi împărțită în pași clari și logici, asemănători programării. Activitatea dezvoltă abilități de rezolvare a problemelor, secvențiere și luare a deciziilor. Descrierea activității: 1. Elevii lucrează în perechi; un elev joacă rolul „Programatorului” (persoana care creează instrucțiunile), iar celălalt rolul „Utilizatorului” (persoana care urmează instrucțiunile).

Etapale lecției	Descriere	Conținut
		2. „Programatorul” descrie o rutină obișnuită de dimineață (de exemplu: „pregătirea pentru școală”) ca și cum ar crea un „program” sau un set de instrucțiuni. ● Trebuie incluse etape precum trezirea, spălatul pe dinți, alegerea hainelor, micul dejun etc. ● Trebuie incluse puncte de decizie (de exemplu: „dacă este frig, poartă o jachetă”) și repetări (de exemplu: „spală-te pe dinți de două ori”). 3. „Utilizatorul” ascultă și apoi execută rutina exact așa cum a fost descrisă. 4. Se schimbă rolurile, astfel încât ambii elevi să exerseze atât oferirea, cât și urmarea instrucțiunilor. 5. La final, se discută ce a fost ușor sau dificil în înțelegerea și urmarea instrucțiunilor și cum se leagă acest lucru de împărțirea problemelor în pași în programare. Materiale necesare: ● Nu sunt necesare materiale specifice, doar spațiu pentru jocul de rol. ● Opțional: se poate oferi o listă de verificare sau un exemplu de rutină pentru a-i ajuta pe elevi să se pregătească. Rezultate ale învățării: ● Elevii vor învăța să transforme rutinele zilnice într-o succesiune de instrucțiuni, similar programării. ● Își vor îmbunătăți abilitățile de comunicare și înțelegerea logicii condiționale și a secvențierii.

Etapile lecției	Descriere	Conținut
Recapitulare		1. Ce este un program? A) Un tip de hardware folosit în calculatoare. B) Un joc jucat pe calculator. C) Un set de instrucțiuni pe care un calculator le urmează pentru a realiza o sarcină. ☒ D) Un dispozitiv care stochează date.
	Întrebări	2.Care dintre următoarele este un exemplu de variabilă în programare? A) O comandă pentru afișarea textului pe ecran. B) O locație de stocare care reține date, cum ar fi un număr sau un text. ☒ C) Un mesaj care explică ce face codul. D) Un tip de limbaj de programare.
		3.Ce face o buclă (loop) într-un program? A) Verifică dacă o condiție este adevărată sau falsă. B) Repetă un set de instrucțiuni de mai multe ori. ☒ C) Stochează date pentru utilizare ulterioară.

Etapale lecției	Descriere	Conținut
		D) Trimite date către un alt calculator.
		4. De ce este important debugging-ul în programare? A) Pentru a adăuga noi funcționalități programului. B) Pentru a identifica și corecta erorile (bug-urile) din cod. ☒ C) Pentru a face programul să ruleze mai rapid. D) Pentru a traduce codul în alte limbaje.