

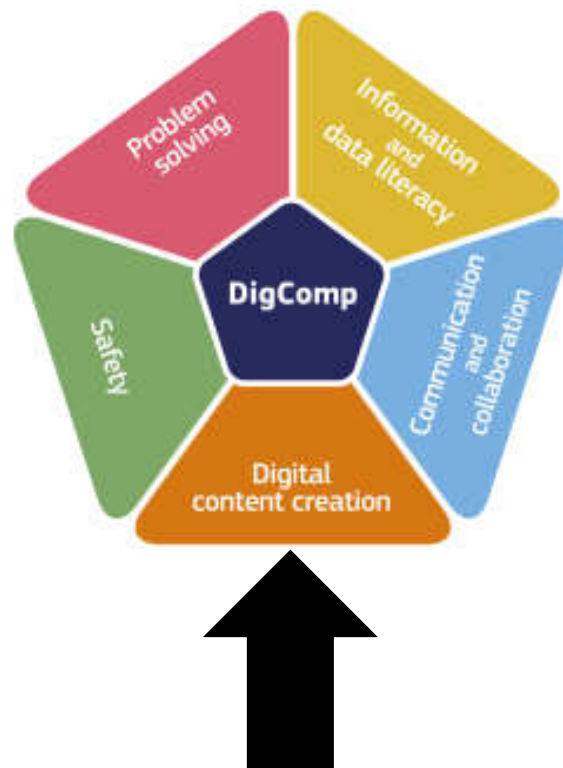


**Microcredite pentru domeniul de competențe:**

### **3. CREARE DE CONȚINUT DIGITAL**

**Competența:**

#### **3.4 PROGRAMARE**



## Cuprins

INTRODUCERE .....	8
PRECONDIȚII .....	9
DICȚIONAR DE TERMENI ȘI ABREVIERI .....	10
NIVEL DEBUTANT (Nivel 1 ȘI NIVEL 2) .....	11
Limbaje de programare – tipuri și clasificare (MC 3.4.A.1) .....	11
Limbaje de programare – tipuri și clasificare (MC 3.4.A.1) .....	12
Specificații.....	12
Rezultatele învățării (MC 3.4.A.1).....	13
Descriere (MC 3.4.A.1).....	13
Întrebări (MC 3.4.A.1).....	13
IDE - Mediu de dezvoltare integrat (MC 3.4.A.2) .....	14
Specificații.....	14
Rezultatele învățării (MC 3.4.A.2).....	15
Descriere (MC 3.4.A.2).....	15
Întrebări (MC 3.4.A.2).....	15
Alegeți și instalați un IDE (MC 3.4.A.3) .....	16
Specificații.....	16
Rezultatele învățării (MC 3.4.A.3).....	17
Descriere (MC 3.4.A.3).....	17
Întrebări (MC 3.4.A.3).....	17
Documentație de programare (MC 3.4.A.4).....	18
Specificații.....	18
Rezultatele învățării (MC 3.4.A.4).....	19
Descriere (MC 3.4.A.4).....	19
Întrebări (MC 3.4.A.4).....	19
Domenii de aplicație (MC 3.4.A.5).....	20
Specificații.....	20
Rezultatele învățării (MC 3.4.A.5).....	21
Descriere (MC 3.4.A.5).....	21
Întrebări (MC 3.4.A.5).....	21

NIVEL INTERMEDIAR (Nivelul 3 și Nivelul 4).....	1
Specificațiile/funcționalitățile unei aplicații (MC 3.4.B.1).....	22
Specificațiile/funcționalitățile unei aplicații (MC 3.4.B.1).....	23
Specificații.....	23
Rezultatele învățării (MC 3.4.B.1).....	24
Descriere (MC 3.4.B.1).....	24
Întrebări (MC 3.4.B.1).....	24
Alegerea unui limbaj de programare corespunzător platformei țintă (MC 3.4.B.2).....	25
Specificații.....	25
Rezultatele învățării (MC 3.4.B.2).....	26
Descriere (MC 3.4.B.2).....	26
Întrebări (MC 3.4.B.2).....	26
Alegerea unui IDE corespunzător limbajului de programare (MC 3.4.B.3).....	27
Specificații.....	27
Rezultatele învățării (MC 3.4.B.3).....	28
Descriere (MC 3.4.B.3).....	28
Întrebări (MC 3.4.B.3).....	28
Sintaxa limbajului de programare (MC 3.4.B.4).....	29
Specificații.....	29
Rezultatele învățării (MC 3.4.B.4).....	30
Descriere (MC 3.4.B.4).....	30
Întrebări (MC 3.4.B.4).....	30
Tipuri de date (MC 3.4.B.5).....	31
Specificații.....	31
Rezultatele învățării (MC 3.4.B.5).....	32
Descriere (MC 3.4.B.5).....	32
Întrebări (MC 3.4.B.5).....	32
Variabile și operatori (MC 3.4.B.6).....	33
Specificații.....	33
Rezultatele învățării (MC 3.4.B.6).....	34
Descriere (MC 3.4.B.6).....	34
Întrebări (MC 3.4.B.6).....	34
Instrucțiuni de control flux (MC 3.4.B.7).....	35

Specificații.....	35
Rezultatele învățării (MC 3.4.B.7).....	36
Descriere (MC 3.4.B.7).....	36
Întrebări (MC 3.4.B.7).....	36
Rularea programelor (MC 3.4.B.8) .....	37
Specificații.....	37
Rezultatele învățării (MC 3.4.B.8).....	38
Descriere (MC 3.4.B.8).....	38
Întrebări (MC 3.4.B.8).....	38
NIVEL AVANSAT (Nivelul 5 Și Nivelul 6) .....	1
Metode/funcții (MC 3.4.C.1) .....	40
Specificații.....	40
Rezultatele învățării (MC 3.4.C.1).....	41
Descriere (MC 3.4.C.1).....	41
Întrebări (MC 3.4.C.1).....	41
Operații de intrare/ieșire (I/O) (MC 3.4.C.2) .....	42
Specificații.....	42
Rezultatele învățării (MC 3.4.C.2).....	43
Descriere (MC 3.4.C.2).....	43
Întrebări (MC 3.4.C.2).....	43
Biblioteci și module (MC 3.4.C.3) .....	44
Specificații.....	44
Rezultatele învățării (MC 3.4.C.3).....	45
Descriere (MC 3.4.C.3).....	45
Întrebări (MC 3.4.C.3).....	45
Paradigme de programare (MC 3.4.C.4).....	46
Specificații.....	46
Rezultatele învățării (MC 3.4.C.4).....	47
Descriere (MC 3.4.C.4).....	47
Întrebări (MC 3.4.C.4).....	47
Programare imperativă și orientată pe obiecte (MC 3.4.C.5) .....	48
Specificații.....	48
Rezultatele învățării (MC 3.4.C.5).....	49

Descriere (MC 3.4.C.5).....	49
Întrebări (MC 3.4.C.5).....	49
Programare logică și funcțională (MC 3.4.C.6) .....	50
Specificații.....	50
Rezultatele învățării (MC 3.4.C.6).....	51
Descriere (MC 3.4.C.6).....	51
Întrebări (MC 3.4.C.6).....	51
Programare concurentă (MC 3.4.C.7).....	52
Specificații.....	52
Rezultatele învățării (MC 3.4.C.7).....	53
Descriere (MC 3.4.C.7).....	53
Întrebări (MC 3.4.C.7).....	53
Testare și depanare (MC 3.4.C.8) .....	54
Specificații.....	54
Rezultatele învățării (MC 3.4.C.8).....	55
Descriere (MC 3.4.C.8).....	55
Întrebări (MC 3.4.C.8).....	55
Limbaje de programare compilate vs. interpretate (MC 3.4.C.9) .....	56
Specificații.....	56
Rezultatele învățării (MC 3.4.C.9).....	57
Descriere (MC 3.4.C.9).....	57
Întrebări (MC 3.4.C.9).....	57
Limbaje de marcare (MC 3.4.C.10) .....	58
Specificații.....	58
Rezultatele învățării (MC 3.4.C.10).....	59
Descriere (MC 3.4.C.10).....	59
Întrebări (MC 3.4.C.10).....	59
NIVEL EXPERT (Nivelul 7 și Nivelul 8).....	1
Soluții de proiectare pentru probleme complexe (MC 3.4.D.1).....	60
Soluții de proiectare pentru probleme complexe (MC 3.4.D.1).....	61
Specificații.....	61
Rezultatele învățării (MC 3.4.D.1).....	62
Descriere (MC 3.4.D.1) .....	62

Întrebări (MC 3.4.D.1).....	62
Management de proiect (MC 3.4.D.2).....	63
Specificații.....	63
Rezultatele învățării (MC 3.4.D.2).....	64
Descriere (MC 3.4.D.2) .....	64
Întrebări (MC 3.4.D.2).....	64
Conducere. Creativitate (MC 3.4.D.3) .....	65
Specificații.....	65
Rezultatele învățării (MC 3.4.D.3).....	66
Descriere (MC 3.4.D.3) .....	66
Întrebări (MC 3.4.D.3).....	66
ANEXA 1: REZULTATELE ÎNVĂȚĂRII .....	67
DOMENIUL DE COMPETENȚĂ: CREARE DE CONȚINUT DIGITAL.....	67
COMPETENȚA: PROGRAMARE .....	67
NIVEL DEBUTANT (NIVELUL 1 și NIVELUL 2).....	69
NIVEL INTERMEDIAR (NIVELUL 3 și NIVELUL 4).....	71
NIVEL AVANSAT (NIVELUL 5 și NIVELUL 6).....	74
EXPERT (NIVELUL 7 și NIVELUL 8).....	78

## INTRODUCERE

Limbașele de programare cuprind o gamă largă de subiecte și caracteristici care permit dezvoltatorilor să comunice instrucțiuni către computere.

Diferite limbașe de programare pot accentua anumite subiecte mai mult decât altele. Învățarea mai multor limbi vă poate oferi o perspectivă mai largă asupra programării și vă poate ajuta să alegeți cel mai bun instrument pentru diferite proiecte.

Aplicațiile software sunt dezvoltate pentru un domeniu de aplicație specific (de exemplu, finanțe, medicină, auto, Internet, securitate). Cunoașterea aceluși domeniu este importantă pentru a înțelege și formaliza și dezvolta funcționalitățile viitoarei aplicații.

Dezvoltarea ariilor de cunoștințe și a atitudinilor pentru dezvoltarea aplicațiilor într-un limbaj de programare prin educație formală, programe de formare și experiență practică poate îmbunătăți abilitățile de programare ale unui individ, dar pentru a performa în orice limbaj de programare este nevoie de multă practică și experiență.



## PRECONDIȚII

Pentru a dezvolta cunoștințe, abilități și atitudini legate de competența COPYRIGHT ȘI LICENȚE, mai multe domenii servesc ca premise. Acestea includ:

1. **Cunoștințe de bază de calculator** : ar trebui să fiți familiarizați cu cum să utilizați un computer, să navigați în sistemul de fișiere, să creați și să gestionați fișiere și foldere și să efectuați sarcini de bază precum copierea, lipirea și instalarea de software.
2. **Matematică** : Deși nu toate limbajele de programare necesită matematică avansată, a avea o înțelegere de bază a aritmeticii, algebrei și logicii poate fi benefică pentru rezolvarea problemelor și înțelegerea algoritmilor.
3. **Gândire logică** : Programarea implică împărțirea problemelor în pași logici mai mici și conceperea de soluții. Dezvoltarea unor abilități puternice de gândire logică vă va ajuta să scrieți cod mai eficient și mai organizat.
4. **Abilități de rezolvare a problemelor** : Programarea se referă la rezolvarea problemelor. A fi capabil să abordeze problemele metodic și să gândească critic va fi esențial în scrierea unui cod eficient.
5. **Cunoașterea limbii engleze** : Cele mai multe resurse de programare, tutoriale și documentație sunt disponibile în limba engleză. O bună înțelegere a limbii engleze va facilita accesul la materiale de învățare și comunicarea cu comunitatea de programare.
6. **Înțelegerea de bază a calculatoarelor și a algoritmilor** : Familiarizarea cu modul în care computerele procesează datele și execută instrucțiuni va oferi o bază pentru înțelegerea modului în care limbajele de programare interacționează cu hardware-ul. Înțelegerea algoritmilor de bază va fi de asemenea utilă în programare.
7. **Sisteme de operare** : cum ar fi Windows, macOS sau Linux. Un sistem de operare (OS) este o componentă software fundamentală care gestionează hardware-ul computerului și oferă servicii și interfețe pentru programele utilizatorului. Acționează ca un intermediar între aplicații și hardware, permițând utilizatorilor să interacționeze cu resursele computerului într-un mod mai ușor de utilizat și mai eficient.

## DICȚIONAR DE TERMENI ȘI ABREVIERI

**Microcredit (microcredențial, precutat MC)** = curs de scurtă durată privitor la un anumit subiect sau arie de cunoaștere restrânsă și finalizat cu o formă de certificare.

**Rezultatele Învățării (engl. : Learning Outcome, prescurtat LO)** = Rezultatele unui proces sau experiențe de învățare, ce pot fi validate prin diverse forme de evaluare.

**K - S - A (engl. : Knowledge, Skills, Attitudes)** = Cunoștințe, Abilități, Atitudini care sunt rezultatul unui proces de învățare.

**NIVEL DEBUTANT  
(NIVEL 1 ȘI NIVEL 2)**



## Limbaje de programare – tipuri și clasificare (MC 3.4.A.1)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Limbaje de programare – tipuri și clasificare</b> <b>Cod: MC 3.4.A.1</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	DEBUTANT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.A.1)

Rezultate ale învățării (ref. Nivelul 1-2 LO 3.4.1 - 3.2.4):

- Înțelegerea tipurilor de aplicații care pot fi dezvoltate într-un limbaj de programare.
- Explicați caracteristicile diferitelor tipuri de aplicații privind platforma de lucru și mediul.
- Este conștient de modul în care limbajul de programare poate fi utilizat pentru a dezvolta aplicații pentru diverse sisteme și medii.
- Enumerați o listă cu programare în limbaj comun.

### Descriere (MC 3.4.A.1)

La acest nivel de bază de expertiză în programare, dezvoltatorii ar trebui să înțeleagă diferitele tipuri de aplicații dezvoltate prin limbaje de programare. Ei ar trebui să aibă o înțelegere fundamentală că aplicațiile desktop rulează pe sisteme de operare precum Windows, macOS sau Linux, oferind funcționalitate de sine stătătoare. De asemenea, ar trebui să înțeleagă că aplicațiile mobile vizează platforme precum Android și iOS, adaptate dispozitivelor portabile. Ei ar trebui să realizeze că aplicațiile web sunt folosite pentru crearea de site-uri web și servicii, accesibile prin diferite browsere. Ei ar trebui să știe că aplicațiile încorporate implică programare specializată pentru funcții dedicate în dispozitive precum sistemele IoT, asigurând compatibilitatea cu medii de calcul specifice.

Dezvoltatorii ar trebui să enumere caracteristicile diferitelor tipuri de aplicații, recunoscând trăsăturile cheie ale tipurilor de aplicații pe baza platformei lor de lucru (desktop, mobil, web, încorporat) și a mediului (software autonom, optimizat pentru interfețe tactile, accesat prin internet, vizat pentru anumite funcții în dispozitive care necesită o utilizare eficientă a resurselor și procesare în timp real), recunoscând diferența dintre limbajele de programare specifice platformei și multiplatforme.

La nivel de bază, oamenii ar trebui să fie familiarizați cu exemple de limbaje de programare, inclusiv C, C++, C#, Java, Kotlin, JavaScript, HTML, XML, React, Python, Ruby, Prolog, Lisp și Scala.

### Întrebări (MC 3.4.A.1)

1. Puteți enumera principalele tipuri de aplicații pe baza platformei lor de implementare (desktop, mobil, web, încorporat)?
2. Descrieți pe scurt diferențele în modul în care sunt dezvoltate aplicațiile pentru diferite sisteme și medii.
3. Care sunt caracteristicile cheie ale aplicațiilor desktop, în special ale celor care rulează pe sisteme de operare precum Windows, macOS sau Linux?
4. Explicați caracteristicile aplicațiilor mobile concepute pentru platforme precum Android și iOS.
5. Descrieți modul în care limbajele de programare pot fi utilizate pentru a crea aplicații web, site-uri web și servicii web.
6. Cum poate fi utilizat eficient un limbaj de programare în programarea sistemelor încorporate?
7. Dați exemple de limbaje de programare pentru desktop/mobile/web/aplicații încorporate.

## IDE - Mediu de dezvoltare integrat (MC 3.4.A.2)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>IDE - Mediul de dezvoltare integrat:</b> <b>Cod MC 3.4.A.2</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	DEBUTANT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.A.2)

Rezultate ale învățării (ref. Nivelul 1-2 LO 3.4.5 - 3.2.6):

- Identificați ce este un IDE - Integrated Development Environment.
- Este conștient de principalele caracteristici și funcționalități pentru diferite IDE-uri.

### Descriere (MC 3.4.A.2)

La nivelul de bază al expertizei în programare, indivizii ar trebui să recunoască IDE ca un acronim pentru „Mediul de dezvoltare integrat” și să înțeleagă rolul său în sarcinile de programare legate de crearea de conținut digital.

Dezvoltatorii ar trebui să arate conștientizarea principalelor caracteristici și funcționalități ale diferitelor IDE-uri. Ei înțeleg diferitele instrumente și capabilități oferite de diferite IDE-uri, demonstrând capacitatea de a naviga și de a utiliza aceste instrumente în mod eficient. Această conștientizare include familiaritatea cu funcțiile adaptate pentru crearea de conținut digital, asigurând eficiența activităților de codare, depanare și testare. Aceste caracteristici principale întâlnite în mod obișnuit într-un IDE sunt Editor de cod, compiler/interpret, depanare, management de proiect, control versiuni, automatizare build, șabloane și fragmente de cod, refactorizarea codului, analiza codului pentru erori potențiale, completarea automată, instrumentele de evidențiere a erorilor și a avertismentelor, găsirea și înlocuirea instrumentelor, Navigarea codului, Documentația sau integrarea referințelor API externe, Integrarea testării unitare, Personalizarea și Extensiile.

### Întrebări (MC 3.4.A.2)

1. Ce înseamnă IDE și cum l-ați defini?
2. Explicați rolul IDE în facilitarea dezvoltării software.
3. Puteți identifica și descrie pe scurt trei caracteristici principale întâlnite în mod obișnuit într-un mediu de dezvoltare integrat (IDE) care contribuie la îmbunătățirea productivității și a experienței de codare a dezvoltatorului?
4. Ce instrumente și funcționalități specifice oferă dezvoltatorilor pentru a identifica și rezolva erori în timpul procesului de execuție a codului?
5. Cum beneficiază suportul încorporat pentru sistemele de control al versiunilor, cum ar fi Git, dezvoltatorii în gestionarea modificărilor codului și colaborarea cu alții?
6. Cum contribuie funcționalitatea de completare automată într-un IDE la codificarea eficientă și reducerea erorilor?

## Alegeți și instalați un IDE (MC 3.4.A.3)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Alegeți și instalați un IDE.</b> <b>Cod: MC 3.4.A.3</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	DEBUTANT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review



### Rezultatele învățării (MC 3.4.A.3)

Rezultate ale învățării (ref. Nivelul 1-2 LO 3.4.7 - 3.2.8):

- Găsiți și alegeți un IDE corespunzător unui limbaj de programare.
- Instalați un IDE.

### Descriere (MC 3.4.A.3)

La nivelul de bază al expertizei în programarea pentru crearea de conținut digital, programatorii ar trebui să posede abilități practice legate de selectarea și utilizarea mediilor de dezvoltare integrate (IDE).

În primul rând, ar trebui să demonstreze capacitatea de a găsi și alege un IDE potrivit pentru un anumit limbaj de programare, luând în considerare factori precum suportul pentru limbaj, caracteristicile și compatibilitatea. Aceasta reflectă o înțelegere fundamentală a importanței IDE-urilor în procesul de dezvoltare.

În al doilea rând, ei ar trebui să dea dovadă de competență în instalarea unui IDE, arătând abilitățile practice de a configura mediul de dezvoltare ales. Aceasta include navigarea proceselor de instalare, configurarea setărilor, asigurarea unei configurări continue și gestionarea actualizărilor. Acest lucru ar putea implica configurarea instrumentelor de compilare, setarea opțiunilor de compilare și înțelegerea procesului de compilare.

### Întrebări (MC 3.4.A.3)

1. Cum ați găsi o listă de medii de dezvoltare integrate (IDE) disponibile potrivite pentru un anumit limbaj de programare pe Internet?
2. Având în vedere un set de caracteristici și caracteristici asociate cu diferite IDE-uri, puteți explica procesul de luare a deciziilor în alegerea unui IDE pentru un anumit limbaj de programare?
3. Evidențiați cel puțin trei caracteristici cheie pe care le-ați acorda prioritate la alegerea unui IDE.
4. Descrieți pașii implicați în instalarea unui IDE cu setări predefinite sau personalizate. Include considerații precum descărcarea și rularea unui program de instalare, configurarea setărilor și gestionarea actualizărilor.
5. Cum ați aborda configurarea instrumentelor de compilare, setarea opțiunilor de compilare și înțelegerea procesului de construire într-un IDE?
6. Când instalați un IDE cu setări personalizate, cum ați adapta configurația pentru a îndeplini cerințele specifice?

## Documentație de programare (MC 3.4.A.4)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Documentatie de programare</b> <b>Cod: MC 3.2.A.4</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	DEBUTANT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.A.4)

Rezultate ale învățării (ref. Nivelul 1-2 LO 3.4.9 - 3.2.11):

- Explicați și găsiți documentația de programare a limbajului.
- Este capabil să ruleze un program simplu.

### Descriere (MC 3.4.A.4)

La nivelul de bază al expertizei în programarea pentru crearea de conținut digital, dezvoltatorii ar trebui să demonstreze competență în înțelegerea și localizarea documentației limbajului de programare. Ei ar trebui să recunoască patru tipuri cheie: documentație în limba oficială, tutoriale, postări/discuții și materiale de referință. Aceste cunoștințe fundamentale subliniază importanța diferitelor resurse pentru învățarea eficientă și rezolvarea problemelor.

În al doilea rând, oamenii ar trebui să demonstreze abilități practice în rularea unui program simplu inspirat din documentație. Ei ar trebui să creeze, să editeze și să salveze fișiere de cod sursă într-un mediu de dezvoltare integrat (IDE), înțelegând extensiile de fișiere specifice. În plus, ei sunt capabili să construiască și să compile cod, făcând adesea referire la documentația oficială pentru îndrumare. Aceste abilități reflectă o abordare practică a codificării într-un context de creare de conținut digital, punând bazele pentru sarcini de programare mai avansate.

### Întrebări (MC 3.4.A.4)

1. Explicați cele patru tipuri de documentație în limbajul de programare.
2. Descrieți procesul de creare, editare și salvare a fișierelor de cod sursă într-un mediu de dezvoltare integrat (IDE).
3. Cum vă asigurați că fișierele au extensiile corecte pentru limbajul de programare cu care lucrați?
4. În ce moduri folosiți documentația limbajului de programare, în special documentația oficială, atunci când lucrați la un proiect de codare?
5. Dacă întâmpinați o eroare în timpul procesului de construire și compilare a codului, cum ați aborda depanarea și rezolvarea problemei?

## Domenii de aplicație (MC 3.4.A.5)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Domenii de aplicație.</b> <b>Cod: MC 3.2.A.5</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	DEBUTANT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.A.5)

Rezultate ale învățării (ref. Nivelul 1-2 LO 3.4.12 - 3.2.13):

- Clasificați domeniile de aplicație.
- Identificați cerințele și constrângerile specifice pentru un domeniu de aplicație specific.

### Descriere (MC 3.4.A.5)

La nivelul de bază al expertizei în programarea pentru crearea de conținut digital, dezvoltatorii ar trebui să fie capabili să clasifice domeniile de aplicații, înțelegându-le ca zone din viața reală în care aplicațiile sunt destinate să fie aplicate. Aceste cunoștințe fundamentale asigură o conștientizare a diverselor contexte în care funcționează aplicațiile de creare de conținut digital, cum ar fi finanțe, sănătate, educație, comerț electronic, transport și logistică, divertisment și media, producție și automatizare industrială, guvern, servicii publice, IoT ( Internetul lucrurilor), științe.

În al doilea rând, programatorii prezintă capacitatea de a identifica cerințe și constrângeri specifice într-un domeniu de aplicație dat. Aceștia pot pune întrebări serioase despre considerente critice, cum ar fi cerințele utilizatorilor, structurile de date, algoritmi, măsurile de securitate și standardele de conformitate. Această abilitate reflectă capacitatea de a analiza și înțelege provocările unice asociate cu programarea într-un context specific, punând bazele pentru crearea de conținut digital eficient și direcționat.

### Întrebări (MC 3.4.A.5)

1. Definiți termenul „domeniu de aplicație” (sau domeniu cu probleme) în contextul dezvoltării software.
2. Furnizați cinci exemple de domenii de aplicații diferite în dezvoltarea de software.
3. Cum influențează recunoașterea domeniului și contextului unui domeniu de aplicație proiectarea și implementarea software-ului?
4. Explicați rolul domeniului aplicației în ghidarea dezvoltatorilor în timpul procesului de dezvoltare a software-ului.
5. Enumerați și descrieți pe scurt considerentele cheie pe care un dezvoltator ar trebui să le abordeze atunci când lucrează într-un anumit domeniu de aplicație.
6. Furnizați exemple de aplicații software în cel puțin trei domenii de aplicații diferite.

## NIVEL INTERMEDIAR (NIVELUL 3 ȘI NIVELUL 4)



## Specificațiile/funcționalitățile unei aplicații (MC 3.4.B.1)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Specificațiile/funcționalitățile unei aplicații</b> <b>Cod: MC 3.2.B.1</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.1)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.14 - 3.2.15):

- Explicați principalele specificații/funcționalități pentru o aplicație într-un anumit domeniu.
- Stabiliți principalele specificații/funcționalități pentru o aplicație într-un anumit domeniu.

### Descriere (MC 3.4.B.1)

La nivelul intermediar de expertiză în programarea pentru crearea de conținut digital, dezvoltatorii ar trebui să fie capabili să explice principalele specificații și funcționalități necesare pentru o aplicație dintr-un anumit domeniu. Aceasta implică înțelegerea faptului că specificațiile specifice domeniului sunt cerințele și capacitățile unice esențiale pentru abordarea eficientă a nevoilor aceluși domeniu. De exemplu, o aplicație de comerț electronic necesită înregistrarea utilizatorului, un catalog de produse și procesarea securizată a plăților.

În al doilea rând, oamenii ar trebui să demonstreze capacitatea de a stabili aceste specificații punând întrebări critice despre funcțiile și constrângerile viitoare. Aceștia pot solicita detalii despre înregistrarea utilizatorilor, opțiunile de reluare a testului și mecanismele de feedback, asigurând o înțelegere cuprinzătoare a funcționalităților aplicației. Acest set de abilități de bază îi echipează pe dezvoltatori să alinieze crearea de conținut digital cu caracteristicile și cerințele distinctive ale diverselor domenii de aplicație, cum ar fi finanțe, sănătate, educație, comerț electronic, transport și logistică, divertisment și media, producție și automatizare industrială, guvern, public. servicii, IoT (Internet of Things), științe.

### Întrebări (MC 3.4.B.1)

1. Cum diferă specificațiile în diferite domenii de aplicație și de ce este esențial ca funcționalitățile să se alinieze cu caracteristicile domeniului?
2. Furnizați exemple de specificații și funcționalități specifice domeniului pentru o aplicație educațională.
3. Explicați semnificația punerii întrebărilor critice despre funcțiile și constrângerile viitoare atunci când stabiliți specificațiile pentru o aplicație.
4. Când se stabilesc funcționalități pentru înregistrarea utilizatorilor într-o aplicație de comerț electronic, ce detalii critice ar trebui luate în considerare? Discutați dacă este suficient să cereți numele complet, vârsta, telefonul sau doar e-mailul și de ce.
5. Când proiectați funcționalități pentru o funcție de test, ce considerații ar trebui luate în ceea ce privește reluarea chestionarelor? Discutați procesul de luare a deciziilor pentru amestecarea întrebărilor și/sau răspunsurilor și furnizarea de feedback sau răspunsuri corecte pe baza posibilității de reluare.



## Alegerea unui limbaj de programare corespunzător platformei țintă (MC 3.4.B.2)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Alegerea unui limbaj de programare corespunzător platformei țintă.</b> <b>Cod: MC 3.2.B.2</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.2)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.16 - 3.2.17):

- Este conștient de criteriile de alegere a unui limbaj de programare corespunzător platformei țintă.
- Alegeți un limbaj de programare specific corespunzător platformei țintă.

### Descriere (MC 3.4.B.2)

La nivelul intermediar de expertiză în programare, programatorii ar trebui să demonstreze că sunt conștienți de criteriile de alegere a unui limbaj de programare aliniat cu platforma țintă. Recunoscând că diferitele limbi excelează în diferite domenii de aplicație, dezvoltatorii iau în considerare factori precum dacă aplicația este o aplicație web sau mobilă, software desktop, joc sau instrument de analiză a datelor. În plus, sunt conștienți de punctele forte și punctele slabe ale limbajelor de programare în raport cu platforma aleasă, cum ar fi Java/Kotlin pentru Android sau Swift pentru iOS.

În al doilea rând, dezvoltatorii aleg în mod activ un limbaj de programare specific, pe baza cercetării amănunțite asupra constrângerilor aplicației, a publicului țintă și a platformelor utilizate. Ei iau în considerare aspectele financiare, făcând distincție între opțiunile gratuite și comerciale și evaluează frecvența actualizărilor atât pentru limbaje de programare, cât și pentru mediile de dezvoltare integrate.

### Întrebări (MC 3.4.B.2)

1. Cum contribuie factorii precum punctele forte, punctele slabe și domeniile de expertiză ale limbajelor de programare la luarea unei decizii informate pe baza platformei țintă?
2. Furnizați exemple de limbaje de programare care sunt potrivite pentru anumite platforme.
3. Atunci când alegeți un limbaj de programare corespunzător platformei țintă, ce factori ar trebui luați în considerare cu privire la constrângerile aplicației, publicul țintă și platformele utilizate?
4. Descrieți pașii pe care i-ați urma pentru a căuta informații despre constrângerile aplicației legate de grupul țintă și platformele utilizate.
5. Când alegeți un anumit limbaj de programare, cum evaluați aspectele financiare, inclusiv dacă este gratuit sau comercial, și frecvența actualizărilor?

## Alegerea unui IDE corespunzător limbajului de programare (MC 3.4.B.3)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Alegerea unui IDE corespunzător limbajului de programare.</b> Cod: MC 3.2.B.3
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.3)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.18 - 3.2.19):

- Este conștient de criteriile de alegere a unui IDE.
- Alegeți IDE-ul potrivit pentru un anumit limbaj de programare.

### Descriere (MC 3.4.B.3)

La nivel intermediar, dezvoltatorii ar trebui să cunoască diferitele criterii de selectare a unui IDE, luând în considerare factori precum asocierea cu un limbaj de programare, compatibilitatea cu platforma, suportul de programare, instrumentele integrate și suportul comunității. Conștientizarea lor se extinde la importanța integrării cu instrumente externe și luarea în considerare a ecosistemelor IDE. Indivizii înțeleg că preferințele personale joacă un rol crucial în procesul de selecție, pledând pentru testarea diferitelor IDE-uri pentru a determina confortul și productivitatea.

Mai mult, la acest nivel, dezvoltatorii pot alege IDE-ul potrivit pentru un limbaj de programare specific, aliniind alegerea cu nevoile proiectului și preferințele personale/de echipă. Ei demonstrează familiaritatea cu IDE-urile populare pentru diferite limbi, prezentând o înțelegere nuanțată a selectării celor mai potrivite instrumente pentru diferite proiecte de aplicații.

### Întrebări (MC 3.4.B.3)

1. Enumerați cel puțin trei criterii pentru alegerea unui mediu de dezvoltare integrat (IDE) atunci când vă lansați într-un nou proiect de programare.
2. Explicați semnificația verificării compatibilității unui IDE cu instrumente sau cadre externe. Furnizați un exemplu de scenariu în care integrarea perfectă cu instrumente externe este crucială pentru îmbunătățirea fluxului de lucru de dezvoltare.
3. Descrieți conceptul de ecosistem IDE și modul în care poate beneficia dezvoltatorii.
4. Cum ar putea influența preferințele individuale productivitatea și nivelul de confort al unui dezvoltator și de ce este recomandat să testați diferite IDE-uri înainte de a face o selecție finală?
5. Selectați un limbaj de programare din listă (Java, Python, JavaScript, C++, C#, Swift) și recomandați cele mai potrivite IDE-uri pe baza înțelegerii dvs. a caracteristicilor și capacităților lor. Explicați rațiunea din spatele alegerilor dvs., ținând cont de cerințele limbii și de preferințele echipei.

## Sintaxa limbajului de programare (MC 3.4.B.4)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Sintaxa limbajului de programare.</b> <b>Cod: MC 3.2.B.4</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 6 – Maxim 8 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.4)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.20 - 3.2.22):

- Este conștient de definiția sintaxei pentru un limbaj de programare.
- Este conștient de rolul sintaxei pentru un limbaj de programare.
- Este conștient de principalele reguli ale sintaxei unei limbi.

### Descriere (MC 3.4.B.4)

La nivelul intermediar de expertiză în programare în crearea de conținut digital, dezvoltatorii posedă cunoștințe avansate cu privire la sintaxa limbajelor de programare. Ei sunt conștienți de faptul că sintaxa implică un set de reguli care guvernează structura codului și gramatica, cruciale pentru înțelegerea compilatorului sau interpretului. Recunoscând cele șapte roluri ale sintaxei, dezvoltatorii înțeleg importanța acesteia în analizarea, detectarea erorilor, lizibilitatea codului, consistența limbajului, definirea comportamentului, reflectarea evoluției limbajului și asigurarea compatibilității cu instrumentele și cadrele.

În plus, programatorii sunt conștienți de principalele reguli care guvernează sintaxa unui limbaj. Aceasta include înțelegerea unor aspecte precum comentarii pentru note explicative, terminatori și separatori, cuvinte cheie cu semnificații specifice, reguli de identificare pentru denumirea variabilelor și funcțiilor, regulile operatorilor pentru efectuarea operațiilor, instrucțiunile și expresiile pentru instrucțiuni, blocurile de cod definite prin indentare sau acolade și semnificația sensibilității majusculilor. Această conștientizare sporită le permite dezvoltatorilor să scrie cod coerent și fără erori, promovând lizibilitatea, consecvența și aderarea la cele mai bune practici în aplicațiile de programare.

### Întrebări (MC 3.4.B.4)

1. Ce înseamnă termenul „sintaxă” în contextul unui limbaj de programare?
2. Cum ați descrie rolul sintaxei în asigurarea validității codului și a înțelegerii de către un compilator sau un interpret?
3. Explicați rolul sintaxei în analiza codului sursă și semnificația acestuia în procesul de compilare.
4. Cum contribuie sintaxa la detectarea erorilor în programare?
5. În ce moduri o sintaxă bine definită asigură consistența limbajului?
6. Cum influențează regulile limbajului, cuprinse în sintaxă, comportamentul și semantica unei aplicații dezvoltate?
7. Furnizați exemple despre cum regulile de sintaxă se aliniază cu cele mai bune practici și ajută la evitarea ambiguității în cod.
8. Descrieți scopul comentariilor într-un limbaj de programare și rolul lor în interpretarea codului.
9. Cum contribuie terminatorii și separatorii la claritatea și organizarea codului, în special în instrucțiuni sau liste complexe?

## Tipuri de date (MC 3.4.B.5)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Tipuri de date.</b> <b>Cod: MC 3.2.B.5</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 4 – Maxim 6 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.5)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.23 - 3.2.24):

- Este conștient de clasificarea tipurilor de date și de caracteristicile acestora.
- Enumerați principalele tipuri de date și caracteristicile acestora într-un program de limbă.

### Descriere (MC 3.4.B.5)

La nivelul intermediar de expertiză în programare, dezvoltatorii ar trebui să recunoască faptul că tipurile de date clasifică variabilele sau expresiile pe baza valorilor lor, determinând operațiunile permise, intervalele de valori și cerințele de memorie. Ei înțeleg rolul esențial al tipurilor de date în utilizarea eficientă a memoriei, asigurând acuratețea datelor și influențând funcționalitatea generală a programului. Programatorii apreciază că tipurile de date disponibile într-un limbaj de programare influențează direct operațiunile aplicabile datelor și modelează comportamentul programului.

În plus, dezvoltatorii demonstrează capacitatea de a enumera principalele tipuri de date, explicând reprezentările lor și distingând formele interne și externe. Acestea oferă exemple, cum ar fi Integer pentru numere întregi, Floating-Point pentru numere reale, Character pentru caractere individuale, String pentru secvențe de text, Boolean pentru decizii binare, Array pentru colecții organizate, Pointer pentru adrese de memorie și Struct/Record pentru utilizator compus - tipuri definite. De asemenea, ele descriu reprezentări interne, cum ar fi stocarea în memorie în reprezentare binară.

### Întrebări (MC 3.4.B.5)

1. Care este rolul fundamental al tipurilor de date într-un limbaj de programare și de ce sunt considerate elemente de bază pentru programare?
2. Furnizați o listă cuprinzătoare a principalelor tipuri de date utilizate în programare și subliniați caracteristicile acestora.
3. Alegeți un scenariu de programare specific și explicați ce tipuri de date ar fi cele mai potrivite, justificând alegerile dvs. în funcție de caracteristicile acestora.
4. Faceți diferența între reprezentările interne și externe ale tipurilor de date, folosind exemple pentru a ilustra înțelegerea dvs.
5. În contextul dezvoltării unei aplicații web care gestionează datele utilizatorului, propuneți un tip de date adecvat pentru stocarea vârstelor utilizatorilor, asigurând atât eficiența, cât și acuratețea.
6. Detaliați rolurile tipurilor de date avansate precum Pointers și Struct/Record în programare.
7. Furnizați un exemplu de scenariu în care utilizarea unei matrice este mai avantajoasă decât alte tipuri de date și justificați alegerea dvs.



## Variabile și operatori (MC 3.4.B.6)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Variabile și operatori .</b> Cod: MC 3.2.B.6
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 4 – Maxim 6 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.6)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.23 - 3.2.24):

- Este conștient de cum să declare o variabilă.
- Este conștient de modul de utilizare a operatorilor.

### Descriere (MC 3.4.B.6)

La nivel intermediar, dezvoltatorii ar trebui să aibă înțelegerea că declararea variabilelor este un pas fundamental în programare, asigurându-se că acestea sunt definite înainte de utilizare. Ei recunosc un identificator ca mecanism de denumire pentru diferite entități de programare, respectând reguli definite, cum ar fi permisiunile pentru setul de caractere (litere, cifre, caractere de subliniere) și considerații privind lungimea. De asemenea, ei demonstrează conștientizarea convențiilor de denumire a variabilelor, esențiale pentru menținerea clarității și coerenței codului în limbajele de programare.

Programatorii ar trebui să știe că operatorii au reprezentări simbolice sau cuvinte cheie care facilitează operațiuni specifice asupra operanzilor, care pot fi valori sau variabile. Ei ar trebui să fie familiarizați cu categoriile comune de operatori, cuprinzând aritmetica, comparația, logica, atribuirea, pe biți, condițional, cast, instanceof, accesul membrilor și adresa-de. Ei înțeleg tipurile de operatori, inclusiv unari, binari și ternari, recunoscându-și rolurile distincte în logica de programare. De asemenea, ei înțeleg conceptul de notație operator și variabilitatea acestuia (infix, prefix, post-fix), recunoscând că stilul de notație este dependent de limbaj.

### Întrebări (MC 3.4.B.6)

1. Explicați semnificația declarării variabilelor în programare.
2. Subliniați regulile pentru definirea unui nume de variabilă și furnizați un exemplu de nume de variabilă valid, luând în considerare caracterele permise.
3. Definiți ce este un identificator în contextul programării.
4. Descrieți rolul operatorilor într-un limbaj de programare.
5. Enumerați categoriile comune de operatori și oferiți exemple pentru fiecare categorie, ilustrând funcționalitățile lor distincte.
6. Faceți diferența între operatorii unari, binari și ternari.
7. Explicați conceptul de notație operator și variațiile acestuia (infix, prefix, post-fix).

## Instrucțiuni de control flux (MC 3.4.B.7)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Instrucțiuni de control flux</b> <b>Cod: MC 3.2.B.7</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 4 – Maxim 6 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.7)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.27):

- Enumerați și explicați instrucțiunile fluxului de control.

### Descriere (MC 3.4.B.7)

La nivelul intermediar de expertiză în programare, oamenii ar trebui să înțeleagă că instrucțiunile fluxului de control sunt cruciale pentru proiectarea programelor care execută diferite acțiuni bazate pe condiții specifice. Un programator de nivel intermediar posedă cunoștințele necesare pentru a folosi diverse instrucțiuni de flux de control care modelează secvența de execuție a unui program. Aceasta include înțelegerea semnificației instrucțiunilor fluxului de control, care sunt construite integrale care dictează ordinea de execuție a programului. Aceste instrucțiuni oferă mijloacele de implementare a comportamentului de ramificare și buclă condiționată. Dezvoltatorii ar trebui să fie competenți în diferitele tipuri de instrucțiuni de flux de control, inclusiv instrucțiuni condiționale (if, if...else), instrucțiuni bucle (do...while, while, for), instrucțiuni de control bucle (break, continue), instrucțiuni de control (goto) și instrucțiuni de gestionare a excepțiilor (try...catch).

Ei ar trebui să fie capabili să înțeleagă și să aplice logica din spatele fiecărei instrucțiuni de flux de control. Aceasta implică urmărirea condițiilor în care are loc ramificarea, a criteriilor de iterație pentru bucle și a rolului instrucțiunilor de control în alterarea fluxului programului.

Programatorii intermediari ar trebui să demonstreze capacitatea de a lua decizii informate cu privire la momentul în care să utilizeze instrucțiuni specifice fluxului de control. Ei înțeleg implicațiile alegerilor lor asupra comportamentului programului, lizibilității și întreținerii.

### Întrebări (MC 3.4.B.7)

1. Explicați semnificația instrucțiunilor fluxului de control în programare și modul în care acestea contribuie la structura și logica generală a unui program.
2. Furnizați un exemplu de scenariu în care utilizarea eficientă a fluxului de control îmbunătățește funcționalitatea programului.
3. Enumerați și explicați pe scurt fiecare tip de instrucțiune de flux de control, inclusiv condițional, buclă, control al buclei, control și gestionarea excepțiilor.
4. Situații ilustrate în care fiecare tip este utilizat în mod corespunzător.
5. Furnizați o analogie din lumea reală sau un exemplu pentru a ilustra utilizarea unei declarații de flux de control.
6. Pentru instrucțiunile în buclă (do...while, while, for), descrieți diferențele fundamentale și scenariile în care fiecare este cel mai potrivit.
7. Explicați scopul și funcționarea instrucțiunilor de tratare a excepțiilor (try...catch) în controlul fluxului de program în timpul situațiilor de eroare.

## Rularea programelor (MC 3.4.B.8)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Rularea programelor</b> <b>Cod: MC 3.2.B.8</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 2 – Maxim 4 ore
Nivelul învățării	INTERMEDIAR
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.B.8)

Rezultate ale învățării (ref. Nivelul 3-4 LO 3.4.28 - 3.2.30):

- Este conștient de cum să scrie un fișier sursă simplu.
- Este conștient de cum să ruleze un fișier sursă.

### Descriere (MC 3.4.B.8)

La nivelul intermediar de expertiză, dezvoltatorii ar trebui să fie conștienți de crearea unui nou fișier sursă cu extensia de fișier corectă pentru limbajul de programare ales. Ei pot aplica reguli de sintaxă pentru a crea o aplicație simplă care poate implica variabile, funcții, instrucțiuni de flux de control. Competenți în editori de text sau medii de dezvoltare integrate (IDE-uri), ei urmează un proces sistematic: crearea unui fișier nou, scrierea codului, salvarea acestuia cu un nume semnificativ și extensia corespunzătoare și selectarea unei locații pentru recuperarea ușoară. Dezvoltatorul este capabil să organizeze codul în proiecte și să gestioneze fișierele de proiect. Aceasta include crearea structurilor de proiect, adăugarea de dependențe și configurarea căilor de construcție.

Abilitățile se extind până la înțelegerea pașilor nuanțați necesari pe baza limbajului de programare, a instrumentelor și a mediului de dezvoltare utilizat. Cunoștință în compilarea/interpretarea, dacă este necesar, pentru a genera un fișier executabil. Când rulează programul, persoanele pot naviga atât pe interfețele de linie de comandă, cât și pe IDE-uri. În interfața de linie de comandă, ei folosesc comanda corespunzătoare, în timp ce în interfața IDE-ului, ei folosesc opțiunea potrivită pentru a executa aplicația.

### Întrebări (MC 3.4.B.8)

1. Puteți descrie pașii esențiali implicați în crearea unui nou fișier sursă pentru un limbaj de programare?
2. Cum vă decideți asupra unui nume semnificativ pentru fișierul sursă și care sunt considerentele dvs. atunci când alegeți extensia corectă de fișier?
3. Cum vă decideți asupra unei locații potrivite pe computer pentru a salva fișierul pentru a-l recupera mai târziu?
4. Puteți descrie pașii generali pentru a rula un fișier sursă, luând în considerare limbajul de programare, instrumentele instalate și mediul de dezvoltare pe care îl utilizați?
5. Explicați conceptul de compilare în programare și în ce circumstanțe ar trebui să vă compilați codul sursă înainte de a-l rula?
6. Descrieți procesul de rulare a unui program într-o interfață de linie de comandă (terminal)/Mediu de dezvoltare integrat (IDE).
7. Descrieți pașii pe care i-ați urma pentru a organiza codul în proiecte, inclusiv crearea unei structuri de proiect, adăugarea de dependențe și configurarea căilor de construcție.

**NIVEL AVANSAT  
(NIVELUL 5 ȘI NIVELUL 6)**



## Metode/funcții (MC 3.4.C.1)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Metode/funcții.</b> <b>Cod: MC 3.2.C.1</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 4 – Maxim 6 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review



### Rezultatele învățării (MC 3.4.C.1)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.31 - 3.2.33):

- Declarați metode/funcții.
- Definiți metode/funcții.
- Aplica metode/funcții.

### Descriere (MC 3.4.C.1)

La nivelul avansat de expertiză în programare, dezvoltatorii ar trebui să înțeleagă că declararea metodei implică recunoașterea elementelor cheie, cum ar fi numele metodei, tipul de returnare, parametrii și corpul metodei. Aceste cunoștințe acoperă diverse limbaje de programare, asigurând consistența în practicile de declarare a metodelor.

Competența se extinde la definirea metodei, unde expertiza constă în elaborarea comportamentului, logicii și funcționalității metodei pe baza declarației anterioare. Aceasta include crearea unui corp de metodă, formularea algoritmilor și a logicii pentru executarea sarcinii și, dacă este cazul, includerea unei instrucțiuni return pentru a specifica valoarea trimisă înapoi apelantului.

Competența aplicației este demonstrată prin abilitatea de a invoca metode în diferite părți ale aplicației, executând acțiunile desemnate. Aceasta implică o înțelegere cuprinzătoare a modului în care metodele interacționează în baza de cod, contribuind la funcționalitatea și eficiența generală a programului.

### Întrebări (MC 3.4.C.1)

1. Care sunt elementele esențiale ale unei declarații de metodă în programare?
2. Explicați componentele definiției unei metode în programare, inclusiv semnificația corpului metodei, a algoritmului și a instrucțiunii return.
3. Furnizați un exemplu de scenariu în care apelarea unei metode este crucială pentru executarea unor acțiuni specifice într-un program. Descrieți pașii implicați în apelarea unei metode din diferite părți ale bazei de cod.
4. În diferite limbaje de programare, care sunt elementele cheie comune găsite în mod constant în declarațiile de metodă?
5. Demonstrați înțelegerea interacțiunii dintre declararea și definirea metodei.

## Operații de intrare/ieșire (I/O) (MC 3.4.C.2)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Operații de intrare/ieșire (I/O).</b> <b>Cod: MC 3.2.C.2</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 4 – Maxim 6 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.2)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.34 - 3.2.35):

- Este conștient de operațiunile de intrare/ieșire (I/O).
- Efectuați operațiuni de intrare/ieșire (I/O).

### Descriere (MC 3.4.C.2)

La nivelul avansat de expertiză în programare, dezvoltatorii trebuie să știe că operațiunile de intrare/ieșire (I/O) în programare sunt mijloacele de citire a datelor din surse externe sau de trimitere a datelor către destinații externe, cum ar fi interacțiunea cu utilizatorii, fișierele, rețeaua, resurse, baze de date și alte sisteme. Recunoaște rolul fundamental pe care îl joacă operațiunile I/O în procesarea datelor. Competențele avansate în operațiuni I/O demonstrează capacitatea de a efectua diverse operațiuni I/O, inclusiv citirea intrărilor din diverse surse (tastatură, fișiere și conexiuni la rețea), scrierea ieșirii (în consolă, fișiere, rețea și baze de date), implementarea formatei ieșire pentru afișaje care pot fi citite de om, aplicarea tehnicilor de gestionare a erorilor pentru gestionarea excepțiilor în timpul operațiunilor I/O, efectuarea proceselor de serializare și deserializare pentru stocarea sau transmiterea datelor, înțelegerea și utilizarea fluxurilor standard (input - stdin, output - stdout și error - stderr) .

### Întrebări (MC 3.4.C.2)

1. Care este scopul operațiunilor de intrare/ieșire (I/O) în programare și de ce sunt esențiale pentru aplicațiile software care interacționează cu sistemele externe?
2. Demonstrați pașii implicați în efectuarea operațiunilor I/O. Furnizați exemple de citire a intrărilor din diferite surse (tastatură, fișiere, rețea) și de scriere a rezultatelor către diferite destinații (consolă, fișiere, rețea, baze de date).
3. Cum ați afișa datele într-un format care poate fi citit de om, aplicând formatarea numerelor și datelor în timpul operațiunilor de ieșire?
4. Definiți serializarea și deserializarea în contextul programării.
5. Cum sunt utilizate fluxurile de intrare (stdin), de ieșire (stdout) și de eroare (stderr) în programare?

## Biblioteci și module (MC 3.4.C.3)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Biblioteci și module.</b> <b>Cod: MC 3.2.C.3</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.3)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.36 - 3.2.37):

- Este la curent cu bibliotecile și modulele.
- Operați cu biblioteci și module.

### Descriere (MC 3.4.C.3)

La nivelul avansat de expertiză în programare, dezvoltatorii ar trebui să fie capabili să definească conceptul de biblioteci și module în programare. Explicați rolul bibliotecilor ca colecții de cod pre-compilat cu funcționalități specifice și module ca unități de cod autonome. De asemenea, ei ar trebui să poată discuta despre importanța unor factori precum compatibilitatea, acordarea de licențe, documentația și sprijinul comunității atunci când iau în considerare utilizarea bibliotecilor și modulelor.

În al doilea rând, ar trebui să explice impactul operațiunii cu biblioteci și module asupra vitezei și eficienței dezvoltării software. Discutați cum utilizarea codului pre-scris permite dezvoltatorilor să se concentreze pe logica de bază și să implementeze mai rapid aspecte unice ale unei aplicații.

### Întrebări (MC 3.4.C.3)

1. Explicați conceptul de biblioteci și module în programare. Ce diferențiază o bibliotecă de un modul și cum sunt acestea utilizate pentru a îmbunătăți aplicațiile software?
2. Discutați importanța luării în considerare a unor factori precum compatibilitatea, acordarea de licențe, documentația și suportul comunității atunci când lucrați cu biblioteci și module.
3. Descrieți beneficiile reutilizării codului prin biblioteci și module.
4. Cum poate funcționa operarea cu biblioteci și module să contribuie la o dezvoltare mai rapidă?
5. Cum poate un programator să participe activ și să beneficieze de pe urma eforturilor comunității de a dezvolta și îmbunătăți bibliotecile?

## Paradigme de programare (MC 3.4.C.4)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Paradigme de programare.</b> <b>Cod: MC 3.2.C.4</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 50%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.4)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.38):

- Enumerați și explicați paradigmele de programare.

### Descriere (MC 3.4.C.4)

La nivelul avansat de expertiză în programare, dezvoltatorii trebuie să știe că paradigmele de programare au abordări fundamentale pentru structurarea și organizarea codului, reprezentând diferite filosofii și metodologii de proiectare și implementare a software-ului. Fiecare paradigmă oferă un mod distinct de a gândi despre proiectarea programului, subliniind anumite principii și practici.

Programatorii ar trebui să enumere și să explice caracteristicile principalelor paradigme de programare, inclusiv programarea imperativă, orientată pe obiecte, funcțională, logică, concurentă, paralelă, procedurală, bazată pe evenimente și structurată. Evidențiați conceptele cheie și exemplele asociate fiecărei paradigme. De asemenea, ar trebui să poată evidenția faptul că multe limbaje și cadre de programare moderne suportă o combinație de paradigme de programare.

### Întrebări (MC 3.4.C.4)

1. Enumerați principalele paradigme de programare și explicați pe scurt caracteristicile fiecăreia. Furnizați un exemplu de limbaj pentru fiecare paradigmă.
2. Descrieți caracteristicile cheie ale programării imperative. Furnizați exemple de limbaje de programare care urmează această paradigmă.
3. Explicați principiile de bază ale programării funcționale. Furnizați exemple de limbaje de programare funcționale.
4. Discutați despre caracteristicile programării orientate pe obiecte (OOP). Furnizați exemple de limbaje OOP.
5. Exemplificați limbaje de programare moderne sau cadre care acceptă o combinație de paradigme de programare.
6. Furnizați un exemplu de scenariu în care o combinație de paradigme ar putea fi avantajoasă.

## Programare imperativă și orientată pe obiecte (MC 3.4.C.5)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Programare imperativă și orientată pe obiecte.</b> <b>Cod: MC 3.2.C.5</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review



### Rezultatele învățării (MC 3.4.C.5)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.39 - 3.2.40):

- Explicați conceptele pentru programarea procedurală imperativă.
- Explicați conceptele de programare orientată pe obiecte (OOP).

### Descriere (MC 3.4.C.5)

La nivelul avansat de expertiză în programare, dezvoltatorii ar trebui să știe că programarea procedurală imperativă, o paradigmă fundamentală, se concentrează pe secvențierea pașilor și a procedurilor pentru îndeplinirea sarcinilor. Conceptele cheie cuprind secvențierea, variabilele, structurile de control, programarea structurată, modularitatea și procedurile/funcțiile. O înțelegere solidă a acestor concepte permite programatorilor avansați să creeze cod bine organizat, modular și extensibil, care abordează eficient problemele complexe.

De asemenea, programatorii ar trebui să știe că Programarea Orientată pe Obiect (OOP) este o paradigmă care pune accent pe obiecte, clase și metode pentru organizarea codului și rezolvarea problemelor. Conceptele cheie cuprind obiecte și clase, încapsulare, moștenire, polimorfism, suprasolicitare a metodei, supraîncărcare a metodei, constructori și destructori, transmitere de mesaje, modificatori de vizibilitate.

### Întrebări (MC 3.4.C.5)

1. Explicați conceptul de „secvențiere” în programarea procedurală imperativă.
2. Cum contribuie „modularitatea” la eficacitatea codului procedural imperativ?
3. Discutați semnificația „programarii structurate” în îmbunătățirea mentenanței și a lizibilității codului.
4. Detaliați conceptul de „încapsulare” în OOP și oferiți un exemplu care ilustrează modul în care acesta îmbunătățește securitatea și organizarea codului.
5. Discutați despre principiile „moștenirii” în POO și explicați cum facilitează reutilizarea codului și promovează o structură ierarhică.
6. Descrieți rolul „constructorilor și destructorilor” în POO, subliniind când și de ce sunt esențiali în proiectarea clasei.

## Programare logică și funcțională (MC 3.4.C.6)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Programare logica si functionala.</b> <b>Cod: MC 3.2.C.6</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.6)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.41 - 3.2.42):

- Explicați conceptele pentru logica declarativă.
- Explicați conceptele pentru funcțional declarativ.

### Descriere (MC 3.4.C.6)

La nivelul avansat de expertiză în aplicații de programare, dezvoltatorii ar trebui să aibă o înțelegere profundă a programării logice declarative, punând accent pe exprimarea relațiilor și a faptelor fără a specifica în mod explicit fluxul de control. Această paradigmă, adesea exemplificată de limbaje precum Prolog, permite dezvoltatorilor să declare logica și să lase sistemul să determine secvența de execuție. Programatorul este priceput la construirea de sisteme bazate pe logică, folosind fapte și definierea relațiilor fără a fi nevoie de instrucțiuni procedurale explicite. Aceste cunoștințe sunt cruciale pentru sarcinile care implică sisteme bazate pe reguli, unde exprimarea logicii relațiilor are prioritate față de specificarea execuției pas cu pas.

La programarea funcțională declarativă, programatorul avansat ar trebui să înțeleagă că se concentrează pe exprimarea calculului ca funcții matematice fără a se baza pe starea mutabilă sau efectele secundare. Paradigma funcțională, exemplificată de limbaje precum Haskell și Lisp, promovează o reprezentare mai concisă și mai abstractă a algoritmilor. Programatorul este priceput la crearea de sisteme robuste, modulare și scalabile prin valorificarea puterii funcțiilor de ordin superior, a imuabilității și a concentrându-se pe calcule pure.

### Întrebări (MC 3.4.C.6)

1. Puteți articula principiul cheie al programării logice declarative și cum diferă de paradigmele imperative, subliniind expresia relațiilor și faptelor?
2. Furnizați un exemplu de scenariu în care programarea logică declarativă, cum ar fi Prolog, ar fi mai avantajoasă decât o abordare imperativă.
3. Descrieți principiile fundamentale ale programării funcționale declarative.
4. Cum îmbunătățește paradigma de programare funcțională claritatea și mentenabilitatea codului în comparație cu abordările imperative sau orientate pe obiecte?

## Programare concurentă (MC 3.4.C.7)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Programare concurentă.</b> <b>Cod: MC 3.2.C.7</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 3 – Maxim 5 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.7)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.43 - 3.2.44):

- Enumerați tehnicile de concurență.
- Efectuați concurență.

### Descriere (MC 3.4.C.7)

Expertiza avansată în aplicații de programare implică o înțelegere cuprinzătoare a tehnicilor de concurență și capacitatea de a le implementa eficient. Dezvoltatorii ar trebui să înțeleagă tehnici de multi-threading (thread-uri operate într-un singur proces, partajarea spațiului de memorie) și multi-procesare (utilizarea de procese separate cu spații de memorie distincte pentru concurență).

Ei ar trebui să identifice sarcini independente și să evalueze ce părți ale programului pot rula concomitent fără conflicte. Apoi ar trebui să determine resursele partajate și să aplice mecanisme de sincronizare (blocare, semafore) pentru a gestiona accesul și pentru a preveni cursele de date și pentru a asigura accesul la date în siguranță, folosind măsuri de siguranță și sincronizare.

### Întrebări (MC 3.4.C.7)

1. Descrieți conceptul fundamental de concurență și rolul acestuia în eficiența programului.
2. Faceți diferența între multi-threading și multi-procesare, explicând modul în care fiecare tehnică atinge concurența.
3. Alegeți între multi-threading și multi-procesare în funcție de cerințele și constrângerile unei aplicații date.
4. Având un fragment de cod, identificați sarcinile independente potențiale potrivite pentru execuția concomitentă.
5. Evaluează o bucată de cod și identifică resursele partajate care pot duce la curse de date. Propuneți mecanisme de sincronizare pentru a atenua aceste probleme.
6. Analizați un fragment de cod și identificați zonele potențiale în care pot apărea blocaje sau condiții de cursă. Propuneți soluții pentru a preveni sau rezolva aceste probleme.

## Testare și depanare (MC 3.4.C.8)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Testare și depanare.</b> <b>Cod: MC 3.2.C.8</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 8 – Maxim 10 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.8)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.45 - 3.2.48):

- Testare și depanare.
- Depanați o aplicație.
- Testați o aplicație.

### Descriere (MC 3.4.C.8)

Programatorii de nivel avansat posedă o înțelegere și aplicare cuprinzătoare a fazelor de testare și depanare. Aceștia recunosc testarea ca un proces sistematic de identificare a defectelor, asigurându-se că software-ul respectă specificațiile și se comportă conform intenției. De asemenea, înțelegeți depanarea ca fiind soluția vizată a defectelor, care implică examinarea codului, urmărirea execuției și identificarea cauzei principale a problemei. Cunoașterea și aplicarea tehnicilor de depanare cum ar fi, utilizarea declarațiilor de tipărire în mod strategic pentru a scoate valori variabile, depanarea interactivă în IDE-uri sau instrumente, utilizarea punctelor de întrerupere și inspecția variabilelor, implementează înregistrarea cu biblioteci pentru a înregistra detaliile de execuție a programului pentru o analiză aprofundată, colaborare pentru revizuirea codului.

În al doilea rând, la nivel avansat, dezvoltatorii demonstrează expertiză în metodologii de testare precum testarea unitară (automatizată prin cadre), testarea integrării (pentru a valida interacțiunile dintre diverse unități și module), testarea funcțională (scenarii), testarea regresiei (pentru a preveni introducerea defecte cu modificări noi), testarea performanței (pentru a evalua capacitatea de răspuns a software-ului în diferite condiții), testarea securității (identificarea vulnerabilităților și atenuarea riscurilor potențiale).

### Întrebări (MC 3.4.C.8)

1. Cum ați articula rolul testării și depanării în ciclul de viață al dezvoltării software?
2. Descrieți tehnicile de depanare pe care le-ați folosi pentru a identifica și rezolva probleme într-o bază de cod complexă.
3. Când vă confrunțați cu o eroare într-o aplicație, cum ați utiliza declarațiile de tipărire, depanarea interactivă în IDE-uri, bibliotecile de jurnalizare și revizuirile de cod pentru a identifica și aborda eficient cauza principală a problemei?
4. Explicați importanța diferitelor tipuri de testare (testare unitară, testare de integrare, testare funcțională, testare de regresie, testare de performanță și testare de securitate) în asigurarea unei aplicații software robuste și fiabile. Cum se completează aceste tipuri?
5. Cum abordați procesele de testare și depanare, având în vedere natura lor iterativă?

## Limbaje de programare compilate vs. interpretate (MC 3.4.C.9)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Limbaje de programare compilate vs. interpretate.</b> Cod: MC 3.2.C.9
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 6 – Maxim 8 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review



### Rezultatele învățării (MC 3.4.C.9)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.49 - 3.2.52):

- Explicați diferențele dintre compilat și interpretat (scripting).
- Caracteristicile limbajului compilat.
- Caracteristicile limbajului interpretat.
- Este conștient de tastarea limbajului de programare.

### Descriere (MC 3.4.C.9)

Expertiza avansată în aplicații de programare implică distincția între limbajele compilate și cele interpretate. Limbile compilate sunt supuse unui pas de compilare pre-execuție, traducând codul sursă în cod de mașină eficient sau bytecode. Limbile interpretate, dimpotrivă, execută cod în timp real, linie cu linie, oferind cicluri rapide de dezvoltare și ușurință de depanare.

Profesioniștii avansați recunosc caracteristicile limbajelor compilate, subliniind performanța superioară, permit execuții multiple fără recompilare, dar portabilitate redusă datorită compilării specifice platformei. Exemplele includ C, C++, Rust și Go. Limbile interpretate acordă prioritate ușurinței dezvoltării. Ele prezintă în general performanțe mai lente, dar interpreții moderni încorporează optimizări. În special, se laudă cu portabilitate îmbunătățită, deoarece același cod poate rula pe diferite platforme. Exemplele includ Python, Ruby, JavaScript și PHP.

Mai mult decât atât, practicienii avansați înțeleg distincția dintre limbajele tipizate static și dinamic. Limbajele tipizate static impun declarații de tip în timpul compilării, asigurând corectitudinea tipului, exemplificată de Java sau C++. Limbile tipizate dinamic, tipizate de Python sau JavaScript, determină tipurile în timpul execuției, oferind flexibilitate pentru ca tipurile de variabile să se schimbe dinamic în timpul execuției.

### Întrebări (MC 3.4.C.9)

1. Descrieți diferențele cheie dintre limbajele compilate și cele interpretate în contextul etapei de compilare pre-execuție.
2. Luând în considerare caracteristicile limbajelor compilate, discutați despre compromisul dintre performanță superioară și portabilitate redusă.
3. Explicați avantajele limbajelor interpretate, subliniind execuția lor în timp real și ciclurile rapide de dezvoltare.
4. Discutați implicațiile limbajelor tipizate static, subliniind rolul verificării tipului la compilare.
5. Cum se adaptează tastarea dinamică modificările tipurilor de variabile în timpul execuției programului și ce avantaje aduce această flexibilitate anumitor scenarii de dezvoltare?

## Limbaje de marcare (MC 3.4.C.10)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Limbaje de marcare.</b> <b>Cod: MC 3.2.C.10</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 6 – Maxim 8 ore
Nivelul învățării	AVANSAT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.C.10)

Rezultate ale învățării (ref. Nivelul 5-6 LO 3.4.53 - 3.2.54):

- Explicați conceptele pentru limbaje de marcare.
- Transformarea și stilizarea limbajelor de marcare.

### Descriere (MC 3.4.C.10)

Competențele avansate în aprogramare implică o înțelegere nuanțată a limbajelor de marcare. Aceste limbaje, cum ar fi HTML, XML, Markdown și LaTeX, servesc la adnotarea textului, definindu-i structura, prezentarea și comportamentul în cadrul unui document. Stăpânirea în markup include înțelegerea conceptelor esențiale, inclusiv etichete de marcare, atribute, imbricare, validare și asigurarea interoperabilității pe diverse platforme și dispozitive.

În plus, un profesionist demonstrează capacitatea de a transforma și stiliza conținutul folosind tehnologii precum Cascading Style Sheets (CSS) pentru documente web și Extensible Stylesheet Language Transformations (XSLT) pentru documente XML. Această abilitate implică utilizarea tehnologiilor de transformare pentru a îmbunătăți prezentarea vizuală și reprezentarea structurală a conținutului. Aplicarea competență a acestor tehnici asigură crearea de conținut digital accesibil, bine organizat și plăcut din punct de vedere estetic în diverse contexte și platforme.

### Întrebări (MC 3.4.C.10)

1. Puteți articula conceptele fundamentale ale limbajelor de marcare?
2. Demonstrează înțelegerea rolului limbajelor de marcare în definirea structurii, prezentării și comportamentului textului dintr-un document.
3. Pot explica modul în care limbajele de marcare contribuie la interoperabilitatea pe diverse platforme și dispozitive.
4. Demonstrează competență în utilizarea foilor de stil în cascadă (CSS) pentru stilarea documentelor web.
5. Competență demonstrată în utilizarea Extensible Stylesheet Language Transformations (XSLT) pentru transformarea și modelarea documentelor XML.
6. Demonstrează capacitatea de a aplica în mod eficient tehnologiile de transformare pentru a îmbunătăți prezentarea vizuală și reprezentarea structurală a conținutului.

**NIVEL EXPERT  
(NIVELUL 7 ȘI NIVELUL 8)**



## Soluții de proiectare pentru probleme complexe (MC 3.4.D.1)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Soluții de proiectare pentru probleme complexe.</b> <b>Cod: MC 3.2.D.1</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 8 – Maxim 10 ore
Nivelul învățării	EXPERT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.D.1)

Rezultate ale învățării (ref. Nivelul 7-8 LO 3.4.5% - 3.2.57):

- Stabiliți declarații și cerințe pentru probleme complexe cu definiție limitată.
- Proiectați soluții pentru a rezolva probleme complexe cu mulți factori care interacționează.
- Proiectează implementarea sistemelor informatice complexe.

### Descriere (MC 3.4.D.1)

Profesioniștii înalt specializați sunt capabili să articuleze întrebări critice pentru a dezvălui complexități în enunțuri de probleme prost definite sau în evoluție în domenii specializate. Demonstrează capacitatea de a rafina în mod iterativ declarațiile problemei prin colaborarea cu experți din domeniu și analizând nevoile utilizatorilor. Integrează abordări sistematice și strategice pentru a proiecta soluții pentru probleme complexe cu numeroși factori de interacțiune. Demonstrează competență în descompunerea problemelor, analiza datelor, modelarea matematică și colaborarea între discipline. Aplica tehnici de învățare automată și inteligența artificială pentru analiza și prezicerea interacțiunilor complexe.

Proiectează și integrează componente hardware, software și de rețea pentru a crea sisteme informatice funcționale și sigure. Responsabilitate demonstrată în îndeplinirea cerințelor de dimensiune, performanță și complexitate, asigurând scalabilitatea și ușurința întreținerii. Evaluează și recomandă noi tehnologii, luând în considerare factori precum caracteristicile de securitate, suportul pe termen lung, integrarea și constrângerile bugetare. Ia decizii informate cu privire la selectarea platformelor, limbilor și cadrelor țintă pe baza considerațiilor ecosistemului și a cerințelor proiectului.

Gestionează resursele personale în mod eficient, luând în considerare disponibilitatea bibliotecilor, caracteristicile de securitate, suportul pe termen lung, integrarea și constrângerile bugetare. Menține o comunicare deschisă și transparentă cu părțile interesate, asigurând practici responsabile de dezvoltare.

### Întrebări (MC 3.4.D.1)

1. Cum abordați rafinarea iterativă a declarațiilor problemei, în special atunci când colaborați cu experți în domeniu și luați în considerare nevoile utilizatorilor?
2. Descrieți o situație în care ați aplicat cu succes o abordare sistematică și strategică pentru a proiecta soluții pentru o problemă complexă cu factori multipli care interacționează.
3. Cum vă asigurați competența în descompunerea problemelor, analiza datelor, modelarea matematică și colaborarea între diferite discipline în procesul dvs. de proiectare a soluției?
4. Furnizați exemple despre cum ați aplicat tehnici de învățare automată și inteligență artificială pentru a analiza și a prezice interacțiuni complexe în proiectele dvs.
5. Cum vă asigurați că sistemul computerizat pe care îl proiectați îndeplinește cerințele de dimensiune, performanță și complexitate, asigurând scalabilitate și ușurință de întreținere?
6. Cum luați decizii în cunoștință de cauză atunci când selectați platformele, limbile și cadrele țintă, luând în considerare considerentele ecosistemului și cerințele proiectului?

## Management de proiect (MC 3.4.D.2)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	Management de proiect. Cod: MC <b>3.2.D.2</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 8 – Maxim 10 ore
Nivelul învățării	EXPERT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.D.2)

Rezultate ale învățării (ref. Nivelul 7-8 LO 3.4.58 - 3.2.60):

- Supervizează implementarea sistemelor informatice complexe.
- Supraveghează conformitatea cu specificațiile.
- Implementarea metodelor de management de proiect.

### Descriere (MC 3.4.D.2)

Expertiza foarte specializată în aplicații de programare implică supravegherea implementării sistemelor informatice complexe cu o conștientizare aprofundată a diferitelor limbaje și tehnologii. Managerul pune accent pe conformitatea cu specificațiile, asigurând aderarea precisă la cele mai bune practici de codificare, structurile de proiect, controlul versiunilor și alte concepte vitale pentru crearea de aplicații sofisticate, care pot fi întreținute.

Implementarea metodelor de management de proiect cuprind sarcini precum planificarea cuprinzătoare a proiectului, definirea obiectivelor și dezvoltarea termenelor. Managementul echipei implică reunirea de echipe calificate, atribuirea de roluri și promovarea colaborării. Managerul de proiect se angajează în managementul riscurilor, asigurarea calității și monitorizarea și controlul continuu, asigurându-se că proiectul rămâne pe drumul cel bun. Gestionarea resurselor și documentarea meticuloasă contribuie la execuția de succes a proiectelor software, demonstrând o abordare holistică a ciclului de viață al dezvoltării.

### Întrebări (MC 3.4.D.2)

1. Cum ați asigurat conformitatea cu specificațiile, cele mai bune practici de codificare și structurile de proiect în proiectele dumneavoastră anterioare.
2. Cum a contribuit acest angajament la crearea de aplicații sofisticate și care pot fi întreținute?
3. Explicați-vă abordarea cu privire la planificarea cuprinzătoare a proiectului, inclusiv modul în care definiți obiectivele proiectului, dezvoltați termenele și alocați resursele.
4. Cum a contribuit competența dumneavoastră în managementul proiectelor la execuția de succes a proiectelor software?
5. Descrieți experiența dvs. în formarea de echipe calificate, atribuirea de roluri și promovarea colaborării în cadrul unui proiect de dezvoltare software.
6. Cum gestionați gestionarea riscurilor și asigurarea calității în proiectele de dezvoltare software?



## Conducere. Creativitate (MC 3.4.D.3)

### Specificații

Categorie cursanți	Orice persoană
Titlul și codul microcreditului	<b>Conducere. Creativitate.</b> <b>Cod: MC 3.2.D.3</b>
Țara (Țările)/Regiunea (regiunile) emitentului	IRLANDA, ITALIA, CIPRU, GRECIA, ROMANIA <a href="http://dsw.projectsgallery.eu">http://dsw.projectsgallery.eu</a>
Organisme de acordare	Consortiul DSW Număr proiect: 101087628
Data emiterii	noiembrie 2023
Volumul de lucru necesar pentru învățare	Minim 8 – Maxim 10 ore
Nivelul învățării	EXPERT
Tipul de evaluare	Întrebări notate automat. Număr de întrebări: 10 Scor de promovare: 75%
Forma de participare la activitatea de învățare	Asincronă online
Forma de asigurare a calității	Peer Review

### Rezultatele învățării (MC 3.4.D.3)

Rezultate ale învățării (ref. Nivelul 7-8 LO 3.4.61 - 3.2.62):

- Îndrumați-i pe alții în analiza și dezvoltarea aplicațiilor.
- Propune noi idei și procese în domeniu.

### Descriere (MC 3.4.D.3)

La nivelul de înaltă specializare de expertiză în programarea aplicațiilor, profesionistul își asumă un rol de lider îndrumându-i pe alții în analiza și dezvoltarea aplicațiilor. Ei contribuie activ la practicile profesionale, favorizând învățarea continuă și actualizările cunoștințelor în cadrul echipei. Prin mentorat, proiecte de colaborare și crearea de documentație și cele mai bune practici, ei împuternicesc organizația lor să exceleze. Aceasta implică dezvoltarea de modele reutilizabile, contribuția la proiecte open-source și susținerea practicilor etice și conforme, automatizării și integrării continue.

În plus, expertul încurajează inovația propunând noi idei și procese în domeniu. Aceștia subliniază importanța creativității, a rezolvării problemelor, a cercetării și a colaborării interdisciplinare. Accentul lor se extinde la îmbunătățirea continuă, eficiența, reducerea costurilor, scalabilitate, design centrat pe utilizator, sustenabilitate și managementul riscurilor. În general, acest profesionist se străduiește să aducă schimbări pozitive, progrese și eficiență, asigurându-se că organizația lor rămâne competitivă și adaptabilă într-un mediu în schimbare rapidă .

### Întrebări (MC 3.4.D.3)

1. Cum contribuie activ profesionistul la învățarea continuă și la actualizarea cunoștințelor în cadrul echipei și ce strategii folosesc pentru a promova o cultură a învățării și dezvoltării?
2. Puteți oferi exemple de proiecte de colaborare și impactul contribuțiilor dumneavoastră asupra succesului echipei și a rezultatelor proiectului?
3. În ce moduri își împuternicește individul organizația să exceleze prin dezvoltarea de modele reutilizabile și contribuții la proiecte open-source?
4. Cum pledează expertul pentru practici etice și conforme, automatizare și integrare continuă în procesul de dezvoltare?
5. Puteți împărtăși cazuri în care ați propus noi idei și procese în domeniu, punând accent pe creativitate, rezolvarea problemelor și colaborarea interdisciplinară?



**ANEXA 1: REZULTATELE ÎNVĂȚĂRII**  
**DOMENIUL DE COMPETENȚĂ: CREARE DE CONȚINUT DIGITAL**  
**COMPETENȚA: PROGRAMARE**

DOMENIUL DE COMPETENȚĂ 2 : CREARE DE CONȚINUT DIGITAL		
COMPETENȚA 3.4: PROGRAMARE		
1	La nivel de bază și cu îndrumare, pot:	<ul style="list-style-type: none"> <li>identifica principalele tipuri de aplicații care pot fi dezvoltate într-un limbaj de programare,</li> <li>enumera principalele limbaje de programare, identifica ce este un IDE.</li> </ul>
2	La nivel de bază și cu autonomie și îndrumări adecvate acolo unde este necesar, pot:	<ul style="list-style-type: none"> <li>identifica caracteristicile principale ale diferitelor tipuri de aplicații,</li> <li>detecta caracteristicile și funcționalitățile de bază pentru diferite IDE-uri,</li> <li>să identific domeniul aplicației, explica ce este documentația de programare a limbajului.</li> </ul>
3	Pe cont propriu și rezolvând probleme simple, pot:	<ul style="list-style-type: none"> <li>explica principalele specificații/funcționalități pentru o aplicație într-un anumit domeniu,</li> <li>alege un limbaj de programare specific corespunzător platformei,</li> <li>alege IDE-ul potrivit pentru un anumit limbaj de programare,</li> <li>explica sintaxa unui anumit limbaj de programare.</li> </ul>
4	În mod independent, în funcție de nevoi și rezolvând probleme bine definite și nede rutină, pot:	<ul style="list-style-type: none"> <li>explica tipurile de date,</li> <li>explica și declara variabilele, utiliza operatorii,</li> <li>aplica instrucțiuni de control flux.</li> </ul>
5	Pe lângă îndrumarea altora, pot:	<ul style="list-style-type: none"> <li>defini, declara și aplica metode/funcții, efectua operațiuni de intrare/ieșire (I/O),</li> <li>realiza biblioteci și module, enumera și explica paradigmele de programare.</li> </ul>
6	La nivel avansat, în funcție de nevoile mele și ale altora, și în contexte complexe, pot:	<ul style="list-style-type: none"> <li>explica conceptele pentru programarea orientată pe obiecte (OOP) imperativă sau logica procedurală sau declarativă imperativă sau funcțională declarativă, efectua testare și depanare,</li> <li>explica diferențele dintre programe compilate și interpretat (scripting),</li> <li>explicați conceptele limbajelor de marcare.</li> </ul>
7	La nivel de înaltă specializare, pot:	<ul style="list-style-type: none"> <li>proiecta și supraveghea implementarea sistemelor informatice complexe,</li> <li>crea aplicații pentru a rezolva probleme complexe cu definiție limitată să-mi integrez cunoștințele pentru a contribui la practicile și cunoștințele profesionale și pentru a-i ghida pe alții în analiza și dezvoltarea aplicațiilor.</li> </ul>
8	La cel mai avansat și specializat nivel, pot:	<ul style="list-style-type: none"> <li>crea soluții pentru a rezolva probleme complexe cu mulți factori care interacționează</li> <li>propune noi idei și procese în domeniu.</li> </ul>

## NIVEL DEBUTANT (NIVELUL 1 ȘI NIVELUL 2)

### DOMENIUL DE COMPETENȚĂ 2 : CREARE DE CONȚINUT DIGITAL

#### COMPETENȚA 3.4: PROGRAMARE

##### NIVEL DEBUTANT

Rezultatele învățării	Nivel	K-S-A	Descriere
1. Înțelegerea tipurilor de aplicații care pot fi dezvoltate într-un limbaj de programare .	L1 – L2	K	Enumerați principalele tipuri de aplicații privind platforma de implementare care pot fi dezvoltate pentru desktop, mobil, web, embedded. Descrieți care sunt principalele diferențe dintre ele (cum pot fi utilizate pentru a dezvolta aplicații pentru diverse sisteme și medii).
2. Explicați caracteristicile diferitelor tipuri de aplicații privind platforma de lucru și mediul.	L2	K	Enumerați caracteristicile aplicațiilor pe baza platformei lor de lucru (desktop, mobil, web, încorporat) și a mediului (software autonom, optimizat pentru interfețe tactile, accesat prin internet, vizat pentru funcții specifice în dispozitive care necesită utilizarea eficientă a resurselor și procesare în timp real) .
3. Este conștient de modul în care limbajul de programare poate fi utilizat pentru a dezvolta aplicații pentru diverse sisteme și medii.	L1 – L2	K	Recunoașteți diferența dintre limbajele de programare specifice platformei și cele multiplatforme. Unele limbaje de programare sunt concepute special pentru anumite platforme.
4. Enumerați o listă cu programare în limbaj comun.	L1	K	Cunoaște diferite exemple de limbaje de programare: C, C++, C#, Java, Kotlin, JavaScript, HTML, XML, React, Python, Ruby, Prolog, Lisp, Scala.

5. Identificați ce este un IDE - Integrated Development Environment	L1	K	Știe că IDE înseamnă „Integrated Development Environment”. Este o aplicație software care oferă instrumente și caracteristici complete pentru a facilita dezvoltarea aplicațiilor software.
6. Este conștient de principalele caracteristici și funcționalități pentru diferite IDE-uri	L1 – L2	K	Un mediu de dezvoltare integrat (IDE) oferă de obicei un set cuprinzător de caracteristici și funcționalități pentru a facilita dezvoltarea de software.
7. Găsiți și alegeți un IDE corespunzător unui limbaj de programare	L2	S	Știe cum să găsească pe Internet o listă cu IDE-uri disponibile pentru un anumit limbaj de programare. Pe baza caracteristicilor IDE-urilor se poate alege un IDE.
8. Instalați un IDE	L2	S	Știe cum să instaleze un IDE folosind setările predefinite sau personalizate.
9. Explicați și găsiți documentația de programare a limbajului	L1 – L2	K	Știe că există 4 tipuri de documentație de programare a limbajului: documentație în limba oficială, tutoriale, postări/discuții, materiale de referință.
10. Este capabil să ruleze un program simplu	L2	S	Poate crea, edita și salva fișiere de cod sursă cu extensia specifică din IDE. Este capabil să construiască și să compileze codul (de obicei luat din documentația oficială) în cadrul IDE.
11. Explicați controlul versiunii	L2	S	Controlul versiunilor este folosit pentru a urmări modificările codului . Știe cum să revină la o stare de lucru cunoscută dacă este necesar.
12. Clasificați domeniile de aplicație	L2	K	Domeniile de aplicație sunt zone din viața reală în care se intenționează să se aplice aplicația.
13. Identificați cerințele și constrângerile specifice pentru un domeniu de aplicație specific	L2	S	Poate pune întrebări despre considerente precum cerințele utilizatorilor, structurile de date, algoritmi, măsurile de securitate și standardele de conformitate relevante pentru domeniul aplicației.

## NIVEL INTERMEDIAR (NIVELUL 3 ȘI NIVELUL 4)

DOMENIUL DE COMPETENȚĂ 2 : CREARE DE CONȚINUT DIGITAL			
COMPETENȚA 3.4: PROGRAMARE			
INTERMEDIARI			
Rezultatele învățării	Nivel	K-S-A	Descriere
14. Explicați principalele specificații/funcționalități pentru o aplicație într-un anumit domeniu.	L3	K	Știe că specificațiile sau funcționalitățile pentru o aplicație dintr-un anumit domeniu se referă la cerințele și capacitățile specifice pe care ar trebui să le aibă aplicația pentru a răspunde în mod eficient nevoilor și provocărilor aceluși domeniu.
15. Stabiliți principalele specificații/funcționalități pentru o aplicație într-un anumit domeniu.	L3	A	Adresează întrebări critice despre funcțiile viitoare și despre constrângerile aplicării.
16. Este conștient de criteriile de alegere a unui limbaj de programare corespunzător platformei țintă	L3	K	Conștientizarea alegerii limbajului de programare potrivit pentru aplicația dvs. depinde de mai mulți factori (single/multiplatforme, web/mobil/desktop/).
17. Alegeți un limbaj de programare specific corespunzător platformei țintă	L3	A	Caută informații despre constrângerile aplicației privind grupul țintă, platformele utilizate. Luați în considerare aspectul financiar ca gratuit/comercial, sursă deschisă, frecvența actualizărilor.
18. Este conștient de criteriile de alegere a unui IDE	L3-L4	K	Cunoaște diferite criterii pentru alegerea unui IDE, cum ar fi asociere cu un limbaj de programare, compatibilitate cu platforma (unică sau multiplatformă), suport de programare, instrumente și plugin-uri integrate, interfață UX, documentație și comunitate, performanță, licență și costuri.
19. Alegeți IDE-ul potrivit pentru un anumit limbaj de programare	L3-L4	S	Poate alege IDE-ul în funcție de proiect și de preferințele personale/de echipă. Alegerea ar trebui să fie aliniată cu fluxul de lucru și cerințele, având în vedere că nu există o soluție unică.

20. Este conștient de definiția sintaxei pentru un limbaj de programare	L3	K	Știe că sintaxa în contextul unui limbaj de programare se referă la setul de reguli și convenții care dictează modul în care trebuie scris codul pentru a fi considerat valid și de înțeles de către compilator sau interpret.
21. Este conștient de rolul sintaxei pentru un limbaj de programare	L3-L4	K	Știe că sintaxa are 7 roluri. 1. Analizarea codului sursă. 2. Detectarea erorilor. 3. Lizibilitatea codului. 4. Consecvența limbajului. 5. Reguli de limbaj. 6. Evoluția limbajului. 7. Compatibilitatea diferitelor instrumente, biblioteci și cadru.
22. Este conștient de principalele reguli ale sintaxei unei limbi	L3-L4	K	Știe că principalele reguli ale sintaxei unui limbaj de programare se referă la: 1. Comentarii. 2. Terminatoare și separatoare. 3. Cuvinte cheie. 4. Reguli de identificare. 5. Regulile operatorului. 6. Enunțuri și expresii. 7. Blocuri de cod. 8. Sensibilitatea majusculilor. 9. Reguli de spațiu alb.
23. Este conștient de clasificarea tipurilor de date și de caracteristicile acestora	L4	K	Știe că un tip de date într-un limbaj de programare este o clasificare care specifică tipul de valoare pe care o poate deține o variabilă sau o expresie. Acesta definește setul de operații care pot fi efectuate asupra datelor, intervalul de valori pe care le poate reprezenta și spațiul de memorie necesar pentru stocarea acestora.
24. Enumerați principalele tipuri de date și caracteristicile acestora într-un program de limbă	L4	S	Este capabil să enumere principalele tipuri de date, ce reprezintă acestea și care sunt reprezentarea internă și externă.



25. Este conștient de cum să declare o variabilă	L4	K	<p>Știe că, în funcție de limbaj, variabila de programare trebuie declarată înainte de a fi utilizată.</p> <p>Știe că un identificator este un nume dat unui pachet, clasă, interfață, metodă sau variabilă.</p> <p>Cunoaște regulile definirii unui nume de variabilă.</p>
26. Este conștient de modul de utilizare a operatorilor	L4	K	<p>Știe că operatorii dintr-un limbaj de programare sunt simboluri sau cuvinte cheie care reprezintă operații specifice care trebuie efectuate pe unul sau mai mulți operanzi.</p> <p>Cunoaște categoriile comune de operatori.</p> <p>Cunoaște tipurile de operatori: unari, binari, ternari.</p> <p>Cunoaște notația operatorilor (infix, pre-fix, post-fix).</p>
27. Enumerați și explicați instrucțiunile fluxului de control	L4	K	<p>Știe că instrucțiunile fluxului de control sunt constructe de programare care determină ordinea în care instrucțiunile sunt executate într-un program. Cunoașteți logica fiecărei instrucțiuni de control.</p>
28. Este conștient de cum să scrie un fișier sursă simplu.	L4	S	<p>Este capabil să creeze un nou fișier sursă cu extensia de fișier corespunzătoare programării limbajului.</p> <p>Știe cum să aplice regula de sintaxă pentru a crea o aplicație simplă.</p>
29. Este conștient de cum să ruleze un fișier sursă.	L4	S	<p>Știe să diferențieze pașii exacti în funcție de limbajul de programare utilizat, instrumentele instalate și mediul de dezvoltare.</p> <p>Este capabil să compileze pentru a genera un fișier executabil din codul sursă.</p> <p>Este capabil să ruleze programul în (a) interfața de linie de comandă (terminal), (b) interfața IDE.</p>
30. Implementarea metodelor de management de proiect	L4	S	<p>Este capabil să organizeze codul în proiecte și să gestioneze fișierele de proiect. Aceasta include crearea structurilor de proiect, adăugarea de dependențe și configurarea căilor de construcție.</p>

## NIVEL AVANSAT (NIVELUL 5 ȘI NIVELUL 6)

DOMENIUL DE COMPETENȚĂ 2 : CREARE DE CONȚINUT DIGITAL			
COMPETENȚA 3.4: PROGRAMARE			
AVANSAT			
Rezultatele învățării	Nivel	K-S-A	Descriere
31. Declarați metode/funcții	L5	K	Știe că o declarație de metodă în programare se referă la procesul de definire a caracteristicilor și comportamentului unei funcții în cadrul unei clase sau al unui modul. Știe care sunt elementele cheie ale unei declarații de metodă (consecvente în majoritatea limbajelor de programare): numele, tipul returnării, parametrii, corpul metodei.
32. Definiți metode/funcții	L5	K	Știe că definiția unei metode în programare se referă la implementarea comportamentului, logicii și funcționalității metodei care a fost declarată mai devreme. O definiție de metodă include corpul metodei, algoritmul și logica, instrucțiunea return.
33. Aplica metode/functii	L5	S	Știe cum să apeleze o metodă din diferite părți ale codului pentru a executa acțiunile specificate.
34. Este conștient de operațiunile de intrare/ieșire (I/O).	L5	K	Știe că operațiunile de intrare/ieșire (I/O) în programare se referă la procesul de citire a datelor din surse externe (intrare) sau de trimitere a datelor către destinații externe (ieșire). Operațiunile I/O sunt esențiale pentru interacțiunea cu utilizatori, fișiere, resurse de rețea, baze de date și alte sisteme externe.
35. Efectuați operațiuni de intrare/ieșire (I/O).	L5-6	S	Știe cum să: <ul style="list-style-type: none"> <li>- citire intrare/scriere ieșire: de la/la consolă, fișiere, rețea.</li> <li>- format de iesire.</li> <li>- gestionarea erorii.</li> <li>- serializați și deserializați.</li> </ul>
36. Este la curent cu bibliotecile și modulele	L5-6		Știe că se referă la procesul de utilizare a componentelor de cod pre-scrise pentru a extinde funcționalitatea unei aplicații software. Bibliotecile și modulele sunt colecții de funcții, clase și resurse care au fost create de alți dezvoltatori pentru a oferi funcționalități specifice.

37. Operați cu biblioteci și module	L5-6	S	<p>Știe cum să refolosească codul care a fost deja scris și testat, reducând dublarea efortului.</p> <p>Știe cum să se dezvolte mai rapid prin valorificarea bibliotecilor și modulelor existente.</p> <p>Știe cum să se concentreze asupra logicii de bază utilizând soluțiile existente pentru a implementa aspectele unice ale unei aplicații.</p> <p>Știe cum să caute colaborarea comunității care dezvoltă, întreține și actualizează biblioteci.</p>
38. Enumerați și explicați paradigmele de programare.	L5-6	K	<p>Știe că paradigmele de programare sunt abordări fundamentale pentru structurarea și organizarea codului, definind modul în care diferitele componente ale unui program interacționează între ele. Fiecare paradigmă reprezintă un mod diferit de a gândi despre proiectarea programelor, iar limbaje de programare diferite suportă adesea una sau mai multe paradigme.</p> <p>Știe să enumere principalele paradigme și caracteristicile acestora:</p> <ul style="list-style-type: none"> <li>- Programare imperativă.</li> <li>- Programare funcțională.</li> <li>- Programare orientată pe obiecte (OOP).</li> <li>- Programare procedurală.</li> <li>- Programare logică.</li> <li>- Programare bazată pe evenimente.</li> <li>- Programare simultană și paralelă.</li> <li>- Programare structurată.</li> </ul>
39. Explicați conceptele pentru programarea procedurală imperativă	L6	K	<p>Știe că programarea procedurală imperativă este o paradigmă de programare care se concentrează pe specificarea unei secvențe de pași sau proceduri pe care un program ar trebui să le urmeze pentru a îndeplini o sarcină.</p>
40. Explicați conceptele pentru programarea orientată pe obiecte (OOP)	L6	K	<p>Știe că programarea orientată pe obiecte (OOP) este o paradigmă de programare care se concentrează pe utilizarea obiectelor, claselor și metodelor pentru a structura codul și a rezolva probleme.</p>
41. Explicați conceptele pentru logica declarativă	L6	K	<p>Știe că programarea logică declarativă pune accentul pe exprimarea relațiilor și a faptelor fără a specifica fluxul de control.</p>
42. Explicați conceptele pentru funcțională declarativă	L6	K	<p>Știe că programarea funcțională declarativă se concentrează pe exprimarea calculului ca funcții matematice fără a se baza pe starea mutabilă sau pe efectele secundare.</p>

43. Enumerați tehnicile de concurență	L6	K	<p>Știe că concurența este capacitatea unui program de a gestiona și executa mai multe sarcini simultan, îmbunătățind eficiența și capacitatea de răspuns.</p> <p>Enumerați și explicați tehnicile de concurență:</p> <ul style="list-style-type: none"> <li>- multi-threading.</li> <li>- multi-procesare.</li> </ul>
44. Efectuați concurență	L6	S	<p>Poate identifica sarcini independente și poate determina ce părți ale programului pot rula concomitent fără a interfera unele cu altele.</p> <p>Poate folosi abstracții de concurență la nivel înalt care simplifică gestionarea firelor și proceselor.</p>
45. Testare și depanare	L6	K	<p>Știe că testarea și depanarea sunt faze critice în dezvoltarea software-ului care ajută la asigurarea calității, fiabilității și corectitudinii unui program. Aceste procese implică identificarea și remediarea erorilor, defectelor și comportamentelor neașteptate din cod.</p>
46. Depanați o aplicație	L6	S	<p>Știe cum să aplice tehnici de depanare:</p> <ul style="list-style-type: none"> <li>- Imprimați declarații.</li> <li>- Depanare interactivă.</li> <li>- Înregistrare folosind biblioteci pentru a înregistra informații despre execuția programului.</li> <li>- Revizuire a Codului.</li> <li>- Creați un caz de testare care reproduce problema.</li> </ul>
47. Testați o aplicație	L7	S	<p>Știe cum să testeze funcționalitățile unei aplicații conform specificațiilor. Poate efectua mai multe tipuri de testare:</p> <ul style="list-style-type: none"> <li>- Testarea unitară.</li> <li>- Testarea integrării.</li> <li>- Testare funcțională.</li> <li>- Testare de regresie.</li> <li>- Test de performanță.</li> <li>- Testare de securitate.</li> </ul>
48. Testare și depanare	L6	A	<p>Este deschis să se angajeze în testare și depanare, ținând cont de faptul că aceste procese sunt iterative și necesită atenție la detalii, răbdare și abordări sistematice.</p>

49. Explicați diferențele dintre compilat și interpretat (scripting)	L6	K	Cunoașterea limbajelor compilate și interpretate sunt două abordări diferite pentru executarea codului în programare. Știe că în limbajele compilate, codul sursă este transformat în cod mașină sau într-o reprezentare intermediară de către un compilator înainte de execuție. Știe că în limbile interpretate, codul sursă este executat rând cu linie sau instrucțiune cu instrucțiune de către un interpret.
50. Caracteristicile limbajului compilat	L6	K	Știe că performanța unui limbaj compilat este mai bună deoarece codul este tradus direct în codul mașinii, care este eficient și optimizat pentru hardware-ul țintă, dar este de obicei mai puțin portabil deoarece este specific arhitecturii și sistemului de operare pentru care a fost compilat .
51. Caracteristicile limbajului interpretat	L6	S	Știe că limbile interpretate au, în general, performanțe mai lente decât limbile compilate din cauza sarcinii de traducere în timp real, dar sunt de obicei mai portabile, deoarece codul nu este legat de o anumită platformă sau arhitectură.
52. Este conștient de tastarea limbajului de programare	L4	K	Știe că limbajele de programare sunt tastate static sau dinamic. Știe că în limbajele tipizate static, tipurile de date sunt cunoscute la momentul compilării, iar verificarea tipului este efectuată în timpul fazei de compilare. Știe că în limbile cu tastare dinamică, tipurile de date sunt determinate în timpul execuției.
53. Explicați conceptele pentru limbaje de marcare	L6	K	Știe că limbajele de marcare sunt o modalitate de adnotare a textului pentru a-i defini structura, prezentarea sau comportamentul în cadrul unui document. Știți că limbajele de marcare se bazează pe etichete/elemente de marcare, atribute, imbricare, validare, interoperabilitate pe diferite platforme, dispozitive și software.
54. Transformarea și stilizarea limbajelor de marcare	L6	S	Știe cum să aplice tehnologiile de transformare pentru stilarea/prezentarea conținutului structurii (markup).

## EXPERT (NIVELUL 7 ȘI NIVELUL 8)

### DOMENIUL DE COMPETENȚĂ 2 : CREARE DE CONȚINUT DIGITAL

#### COMPETENȚA 3.4: PROGRAMARE

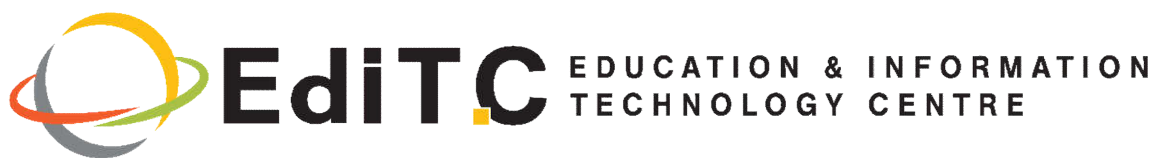
##### EXPERT

Rezultatele învățării	Nivel	K-S-A	Descriere
55. Stabiliți declarații și cerințe pentru probleme complexe cu definiție limitată	L7	A	Puneți întrebări critice legate de domeniul aplicației pentru a afla declarația problemei sau cerințele care nu sunt bine definite sau clare. Colaborați cu experți în domeniu.
56. Proiectați soluții pentru a rezolva probleme complexe cu mulți factori care interacționează	L8	S	Știe cum să combine o abordare sistematică cu una strategică, cum să folosească expertiza în domeniu, abilitățile tehnice, analiza datelor, modelarea, colaborarea și dorința de a se adapta și de a repeta pe măsură ce înțelegerea problemei evoluează.
57. Proiectează implementarea sistemelor informatice complexe	L7	S	Este capabil să selecteze platforma țintă, limbile sau cadrele de dezvoltare, să gestioneze resursele personale (timp și competențe), luând în considerare ecosistemul și bibliotecile, caracteristicile de securitate, suportul și întreținerea pe termen lung a limbii, integrarea cu sistemele existente, bugetul și licențiere.
58. Supervizează implementarea sistemelor informatice complexe	L7	A	Are grijă ca aplicațiile de succes să fie construite folosind diverse limbaje și tehnologii. Păstrați o minte deschisă și fiți dispus să vă adaptați pe măsură ce proiectul progresează și cerințele evoluează.
59. Supraveghează conformitatea cu specificațiile	L7	A	Are grijă să urmeze specificațiile exacte, să implementeze cele mai bune practici privind dezvoltarea codului, să proiecteze structuri, controlul versiunilor și alte concepte care ajută la finalizarea unei aplicații sofisticate și care poate fi întreținută.
60. Implementarea metodelor de management de proiect	L8	S	Știe să gestioneze un proiect software urmând etapele de planificare, execuție și livrare, respectând principiile managementului echipei, managementului riscului și controlului calității.



61. Îndrumați-i pe alții în analiza și dezvoltarea aplicațiilor	L7	A	Integrați-vă cunoștințele și contribuiți activ la practicile profesionale, dând putere echipei și organizației dvs. să exceleze în proiectele lor. Ia în considerare învățarea continuă, mentorat, proiecte de colaborare, creați documentație, dezvolta șabloane, dezvolta standarde de codare, proiectează modele reutilizabile.
62. Propune noi idei și procese în domeniu	L8	A	Încurajați-i pe toată lumea să introducă concepte, strategii, metode sau abordări inovatoare care pot aduce schimbări pozitive, progrese sau eficiență pentru a ajuta organizațiile să rămână competitive și adaptabile într-un mediu în schimbare rapidă.

## Coordonator proiect:



## Parteneri:



Co-funded by  
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.