

MINISTERUL EDUCAȚIEI NAȚIONALE

PROGRAMA ȘCOLARĂ

PROGRAMARE – LIMBAJUL PYTHON

[Curriculum la decizia școlii pentru liceu]

București, 2014

NOTĂ DE PREZENTARE

Efortul de a consolida caracterul științific al disciplinei Informatică pentru ciclul inferior al liceului presupune – pe lângă aprofundarea și sistematizarea nucleului conceptual al acesteia – și o parțială detașare a studiului științific de instrumentele tehnice, particulare, specifice programării: limbajul de programare, mediile de dezvoltare, instrumentele de implementare, depanare și testare. Ar trebui să recunoaștem că, dincolo de preferințele sau deprinderile noastre profesionale, aceasta ar sprijini dezvoltarea unei didactici mai serioase și mai stabile a disciplinei Informatică.

De aceea, disciplina opțională Limbajul de programare Python își propune să deschidă orizontul elevilor care au o pregătire de bază la Informatică, prezentând un nou limbaj de programare, limbaj de mare actualitate care este în același timp și ușor de învățat. Urmărind trendul internațional, trebuie menționat că limbajul Python este cuprins în programa oficială de studiu pentru high school în Statele Unite ale Americii și în unele dintre țările Europei Occidentale. Datorită tradiției actuale de instruire, folosind exclusiv unul dintre limbajele Pascal sau C, o parte covârșitoare a elevilor noștri de liceu au parte de o formare unilaterală, cunoscând un singur limbaj de programare și respingându-le, de regulă, pe toate celelalte.

Opționalul oferă posibilitatea studiului elementelor de programare, cu accent pe sintaxa și pe facilitățile oferite de limbajul Python. Se presupune că elevii au deja o bază formată din deprinderi de gândire algoritmică și de structurare elementară a datelor, această ultimă competență fiind doar recomandată, nu obligatorie. Astfel, opționalul este bine integrat pentru studiu la nivelul claselor a X-a sau a XI-a de liceu teoretic, specializările matematică-informatică sau științele naturii, cu 1 oră/săptămână. El poate fi adaptat pentru clasa a IX-a, adăugând un capitol introductiv de algoritmică, pentru elevii care nu au parcurs conținuturile disciplinei Informatică de la clasa a IX-a, completând eventual opționalul în clasa a X-a, pentru a cuprinde și noțiunile avansate de programare și bibliotecile Python specializate.

Competențele dobândite prin studiul acestui opțional oferă absolvenților de liceu un sprijin în dezvoltarea competențelor de programare, oferind, la sfârșitul claselor de liceu, un profil de programator flexibil, capabil de adaptare și comprehensiune.

În elaborarea programei școlare au fost respectate principiile de proiectare curriculară specifice curriculumului național, valorificându-se în același timp opiniile unor profesori cu o bogată experiență didactică.

Programa are următoarele componente:

- Notă de prezentare
- Competențe generale
- Valori și atitudini
- Competențe specifice și conținuturi
- Sugestii metodologice

COMPETENȚE GENERALE

1. Implementarea algoritmilor folosind limbajul Python
2. Utilizarea structurilor de date specifice limbajului Python
3. Aplicarea instrumentelor avansate de implementare în limbajul Python
4. Elaborarea unor proiecte care să valorifice capacitățile de proiectare și programare

VALORI ȘI ATITUDINI

- Atitudine critică și reflexivă față de informația disponibilă
- Conștientizarea impactului social, economic și moral al utilizării calculatorului;
- Inițiativă în abordarea și rezolvarea unor sarcini variate, utilizând instrumente informatice;
- Manifestarea unui mod de gândire creativ în structurarea și rezolvarea problemelor de informatică;
- Manifestarea disponibilității de a evalua/ autoevalua activități practice.

COMPETENȚE SPECIFICE ȘI CONȚINUTURI

1. Implementarea algoritmilor folosind limbajul Python

Competențe specifice	Conținuturi
<p>1.1. Identificarea instrucțiunilor și structurilor de control specifice limbajului Python</p> <p>1.2. Aplicarea regulilor de sintaxă în scrierea programelor</p> <p>1.3. Utilizarea subprogramelor</p> <p>1.4 Implementarea unor algoritmi cunoscuți cu ajutorul limbajului Python</p>	<p><u>Sintaxa limbajului Python</u></p> <ul style="list-style-type: none">• Variabile. Gestiunea automată a memoriei• Sintaxa instrucțiunilor și structurilor• Atribuirea• Expresii. Comentarii• Citirea și scrierea de la dispozitivele standard• Structura de decizie• Structuri repetitive (while, for); range, break, continue• Funcții predefinite• Subprograme. Parametri <p><u>Aplicații de programare</u></p> <ul style="list-style-type: none">• Implementare unor algoritmi cu structuri liniare, alternative, repetitive• Implementarea unor probleme folosind subprograme

2. Utilizarea structurilor de date specifice limbajului Python

Competențe specifice	Conținuturi
<p>1.1. Identificarea principalelor structuri de date specifice limbajului Python</p> <p>1.2. Aplicarea algoritmilor de prelucrare a structurilor de date</p>	<p><u>Structuri de date</u></p> <ul style="list-style-type: none">• Liste• Dicționare• Șiruri de caractere <p><u>Aplicații de programare</u></p> <ul style="list-style-type: none">• Implementare unor algoritmi cu șiruri de valori și matrice folosind listele• Implementarea unor algoritmi de prelucrare a șirurilor de caractere

3. Aplicarea instrumentelor avansate de implementare în limbajul Python

Competențe specifice	Conținuturi
<p>3.1. Utilizarea mediului de dezvoltare Python</p> <p>3.2. Utilizarea fișierelor</p> <p>3.3. Aplicarea principiilor OOP în programare</p> <p>3.4. Identificarea modalităților avansate de implementare</p>	<p><u>Mediul de lucru</u></p> <ul style="list-style-type: none">• Descărcarea și instalarea• Modul comandă și lucrul cu module• Depanarea programelor <p><u>Lucrul cu fișiere</u></p> <ul style="list-style-type: none">• fișiere text• alte tipuri de fișiere <p><u>Programare orientată obiect</u></p> <ul style="list-style-type: none">• clase și obiecte• clase abstracte• Duck typing <p><u>Tehnici specifice Python</u></p> <ul style="list-style-type: none">• List comprehensions• Map, filter, lambda• metode ”magic”

4. Elaborarea unor proiecte care să valorifice capacitățile de proiectare și programare

Competențe specifice	Conținuturi
4.1. Identificarea etapelor realizării unui proiect 4.2. Proiectarea unei aplicații 4.3. Utilizarea bibliotecilor specializate Python	Proiectarea și implementarea unei aplicații <ul style="list-style-type: none">• Proiectare; limbajul UML• Etapele realizării unei aplicații• Depanarea programelor• Testarea aplicației Biblioteci specializate <ul style="list-style-type: none">• Expresii regulate• Random• CSV• JSON• Cereri HTTP• Grafică□

SUGESTII METODOLOGICE

Opționalul **Programare – Limbajul Python** oferă elevilor oportunitatea de a-și forma competențe avansate de programare și de pregătire pentru integrarea în comunitatea programatorilor Web.

Între competențe și conținuturi există o relație bine determinată: atingerea competențelor implică conținuturile tematice, iar parcurgerea acestora asigură dobândirea de către elevi a competențelor dorite.

Pentru construirea competențelor dorite, activitățile de învățare – predare vor avea un caracter interactiv, centrat pe elev, cu pondere sporită pe activitățile de învățare, pe activitățile practice și mai puțin pe cele teoretice.

Ordinea de parcurgere a temelor aferente conținuturilor din curriculum rămâne la alegerea cadrelor didactice, cu condiția respectării succesiunii logice în abordarea acestora.

Sugestii cu privire la procesul și metodele de predare / învățare

Locul de desfășurare a activităților de învățare se recomandă a fi un laborator echipat cu stații de lucru dotate standard, cu conexiune Internet. Analizele, lucrările și proiectele elevilor, studiile de caz vor fi prezentate la videoproector.

Se vor promova metodele de predare-învățare activ-participative care conduc la rezolvarea problemelor. Ca un argument în favoarea acestor metode se remarcă următoarele avantaje:

- sunt centrate pe elev/cursant și activitate
- pun accent pe dezvoltarea gândirii, formarea aptitudinilor și a deprinderilor
- încurajează participarea elevilor/cursanților, inițiativa, implicarea și creativitatea
- determină un parteneriat profesor-elev/cursant prin realizarea unei comunicări multidirecționale.

Se recomandă renunțarea la expunere și orientarea către metode bazate pe rezolvarea unor sarcini de lucru, utilizându-se cu precădere rezolvarea unei game cât mai variate de aplicații practice și punându-se accent pe realizarea cu exactitate și la timp a cerințelor sarcinilor de lucru. Realizarea proiectelor în cadrul activităților practice va urmări dezvoltarea abilităților de lucru în echipă.

Se recomandă utilizarea documentațiilor vaste și a tutorialului oferite de site-ul Python, utilizată inițial sub îndrumarea profesorului, ulterior pentru studiul individual. Se vor evita astfel prelegerile steci cu caracter de transmitere de informații, ci se va pune accentul pe activități practice demonstrative și pe asistența activităților practice independente.

Profesorii vor alege cele mai potrivite metode: descoperire, discuția în grup, studiul de caz, observația individuală, lucrarea practică, simularea, vizionarea de prezentări. Specificul disciplinei impune metode didactice interactive, recomandând mai ales învățarea prin metode practice/activități de laborator, teme/proiecte .

Sugestii cu privire la evaluare

Evaluarea trebuie să vizeze mai ales capacitatea de a rezolva probleme și de a realiza aplicații.

Pentru evaluare se recomandă a fi utilizate cu precădere, alături de metodele tradiționale:

- observarea sistematică a comportamentului elevilor care permite evaluarea conceptelor, capacităților, atitudinilor față de o sarcină dată, a comunicării
- investigația
- autoevaluarea prin care elevul conștientizează nivelul la care a ajuns, având astfel posibilitatea de a-și impune un ritm propriu și eficient de învățare
- metoda proiectelor

Ca instrumente de evaluare se pot folosi:

- fișe de observație (pentru probe practice)
- fișe test (pentru probe scrise și online)
- fișe de autoevaluare (pentru probe orale, scrise și online)
- proiecte
- studiu de caz