test-e.ro

CAIET ELEV

Lecția 2

Structuri condiționale (if)

Cum ia un program o decizie: de la situații din viața reală la condiții, operatori și cele trei forme — if, if-else, if-elif-else.

DISCIPLINA Informatică **CLASA** a IX-a · mate-info **LIMBAJ** Python 3

NUMELE ELEVULUI .....

CUM FOLOSEȘTI CAIETUL

Acest caiet adună exercițiile propuse din Lecția 2. Rezolvă-le pe hârtie — corectezi cod, prezici rezultate, desenezi scheme logice și scrii programe. Spațiile punctate și liniate sunt pentru scrisul tău. Verificarea automată și evaluarea rămân pe platformă.

CUPRINS

1 Recapitulare prin depanare .....
2 Decizii din viața reală .....
3 Operatori și expresii .....
4 Structura condițională .....
5 Aplicații cu if .....
6 Think-Pair-Share · ax + b = 0 .....
7 Verifică-ți codul .....

SECȚIUNEA 1 · RECAPITULARE

1 Recapitulare prin *depanare*

Fiecare program de mai jos ar trebui să funcționeze, dar are **3 erori**. Citește cu atenție, taie ce e greșit și scrie forma corectă a fiecărei linii.

A

EXERCIȚIUL A · 3 ERORI

Suma a două numere

DEPANARE

Programul ar trebui să citească două numere întregi și să afișeze suma lor.

```
x = int(input("x = "))
y = int(input("y = )
x + y = suma
Print("Suma este:", suma)
```

CELE 3 LINII CORECTATE

B

EXERCIȚIUL B · 3 ERORI

Ultima cifră și numărul trunchiat

DEPANARE

Programul ar trebui să citească un număr întreg și să afișeze **ultima lui cifră** și **numărul trunchiat** (numărul fără ultima cifră).

```
numar = input("Introduceti un numar intreg: ")
ultima_cifra = numar % 10
numar_trunchiat = numar / 10
print("Ultima cifra:" ultima_cifra, "Trunchiat:", numar_trunchiat)
```

CELE 3 LINII CORECTATE

C

SCRIE SINGUR · PSEUDOCOD + PYTHON

Ultima cifră a sumei

PSEUDOCOD SCRIERE COD

Se citesc două numere naturale `a` și `b`. Calculează și afișează **ultima cifră a sumei** lor. Exemplu: pentru `a = 27` și `b = 48`, suma este 75, iar ultima ei cifră este 5.

HINT *ultima cifră a unui număr se obține cu restul împărțirii la 10.*

PASUL 1 · PSEUDOCOD

PSEUDOCODUL TĂU

PASUL 2 · PYTHON

SOLUȚIA TA · PYTHON

SECȚIUNEA 2 · CONCEPTE

2 Decizii din *viața reală*

În multe situații, decizia se ia la un **prag**. Contează dacă valoarea aflată exact pe prag intră sau nu în ramură. Completează, pentru fiecare situație, condiția potrivită folosind unul dintre operatorii `<` `≤` `>` `≥`.

? EXERCIȚIU · COMPLETEAZĂ CONDIȚIA

Ce condiție se potrivește?

COMPLETARE

Dobânda se acordă dacă suma depusă este de **cel puțin** 1000 lei:

`if suma` ——— `1000:`

Ești supraponderal dacă IMC-ul este **strict mai mare** decât 25:

`if imc` ——— `25:`

Elevul este promovat dacă media este **cel puțin** 5:

`if media` ——— `5:`

HINT *„cel puțin" înseamnă că valoarea de pe prag intră în ramură (≥ sau ≤); „strict mai mare/mic" o exclude (> sau <).*

SECȚIUNEA 3 · OPERATORI

3 Operatori și *expresii*

Prezice rezultatul fiecărei expresii. Încercuiește `True` sau `False` și, la nevoie, notează pașii de evaluare pe marginea paginii. Ține cont de precedență: ****** apoi *** / // %**, apoi **+ −**, apoi comparațiile, apoi **not**, **and**, **or**.

1–5 EXERCIȚIU · PREZICE REZULTATUL

Cât produce expresia?

DRY-RUN GRILĂ

1. `10 + 2 * 3 > 15` `True` `False`

2. `20 - 6 // 2 == 16` True False

3. `2 ** 3 + 1 <= 3 * 3` True False

4. `(7 % 2 == 1) and (10 // 3 == 3)` True False

5. `not (5 > 8) and (4 == 5)` True False

6

EXERCIȚIU · ORDONARE

Ordinea de evaluare

ORDONARE

Pentru expresia de mai jos, numerotează operatorii de la **1** (se aplică primul) la **5** (ultimul), după precedență.

```
a + b == c and not (d < e)
```

ORDINEA: + · < · == · NOT · AND

4 SECȚIUNEA 4 · TEMA CENTRALĂ

Structura *condițională*

A

EXERCIȚIU · DRY-RUN

Execută if-elif-else în minte

DRY-RUN

Programul calculează calificativul după notă. Pentru fiecare valoare a lui `n`, scrie ce afișează.

```
if n < 5:
    print("insuficient")
elif n < 7:
    print("suficient")
elif n < 9:
```

```
    print("bine")
else:
    print("foarte bine")
```

n	Ce afișează?
3	
5	
7	
10	

B

EXERCIȚIU · CONCEPEREA SCHEMEI

Construiește schema logică

DRAG & DROP SCHEMĂ LOGICĂ

Traduce pseudocodul într-o schemă logică. Folosește: **oval** pentru START/STOP, **paralelogram** pentru citire/scriere, **romb** pentru condiție (două ieșiri: DA și NU).

```
citește n
dacă n >= 0 atunci
    scrie "pozitiv"
altfel
    scrie "negativ"
```

DESENEAZĂ SCHEMA AICI · NU UITA RAMURILE DA / NU

SECȚIUNEA 5 · APLICAȚII

5 Aplicații cu *if*

Pentru fiecare problemă urmează același drum: înțelegi cerința, desenezi **schema logică**, scrii **pseudocodul** și îl traduci în **program Python**.

1

DECIZIE CU O SINGURĂ RAMURĂ · IF SIMPLU

Mesaj cu privire la vreme

SCHEMĂ SCRIERE COD

Se citește probabilitatea de ploaie `p` (în procente). Dacă `p >= 70`, afișează `"Ia umbrela!"`. Dacă nu, nu se afișează nimic.

SCHEMA LOGICĂ

DESENEAZĂ SCHEMA

PROGRAM PYTHON

SOLUȚIA TA · PYTHON

DECIZIE CU DOUĂ RAMURI · IF-ELSE

2 *Suma cifrelor: pară sau inversare*

SCHEMĂ SCRIERE COD

Se citește un număr natural `n` de două cifre (`10 ≤ n ≤ 99`). Dacă suma cifrelor lui `n` este pară, afișează `"suma para"`; altfel, afișează numărul obținut prin inversarea cifrelor lui `n`.

HINT *cifrele sunt* `u = n % 10` *și* `z = n // 10`*; suma e pară dacă* `(u + z) % 2 == 0`*; numărul inversat este* `10 * u + z`*.*

SCHEMA LOGICĂ

DESENEAZĂ SCHEMA · RAMURILE DA / NU

PROGRAM PYTHON

SOLUȚIA TA · PYTHON

3

DECIZIE CU MAI MULTE RAMURI · IF-ELIF-ELSE

Indicele de masă corporală (IMC)

SCHEMĂ SCRIERE COD

Se citesc masa (kg) și înălțimea (m). Se calculează `imc = masa / inaltime ** 2` și se afișează categoria: **subponderal** (imc < 18.5), **normal** (18.5 ≤ imc < 25), **supraponderal** (25 ≤ imc < 30), **obezitate** (imc ≥ 30).

PROGRAM PYTHON

SOLUȚIA TA · PYTHON

EXERCIȚII SUPLIMENTARE

Exersează singur

SCRIERE COD

R. Se citește un număr întreg `n`. Afișează `"pozitiv"` dacă `n > 0`, altfel `"nepozitiv"`.

P. Se citesc trei numere întregi distincte `a`, `b`, `c` (câte unul pe linie). Afișează-le în ordine crescătoare.

REZOLVĂRILE TALE · PYTHON

SECȚIUNEA 6 · THINK-PAIR-SHARE

6 Ecuația $ax + b = 0$ EXTINDERE · OPȚIONAL

Rezolvarea ecuației de gradul I `a·x + b = 0` are mai multe cazuri, după valorile lui `a` și `b`.

1 ETAPA 1 · THINK

Gândește singur

Ce cazuri apar în funcție de `a` și `b`? Ce se întâmplă când `a == 0`? Notează ideile tale.

IDEILE TALE

2 ETAPA 2 · PAIR

Compară cu colegul

Care sunt diferențele dintre ideile voastre? Există vreo capcană la verificarea `a == 0`?

CONCLUZIILE DISCUȚIEI

3

ETAPA 3 · SHARE

Scrie programul complet

SCRIERE COD

Scrie programul care citește `a` și `b` și tratează toate cazurile: `a ≠ 0` (o soluție), `a == 0 and b == 0` (infinit de soluții), `a == 0 and b ≠ 0` (nicio soluție).

SOLUȚIA TA · PYTHON

7 SECȚIUNEA 7 · VERIFICĂ-ȚI CODUL

Verifică-ți *codul*

Scrie pe hârtie, apoi testează pe platformă. Gândește-te la cazurile-limită.

1

PROBLEMA 1

Maximul a două numere

SCRIERE COD

Se citesc două numere întregi `a` și `b`. Afișează valoarea celui mai mare dintre ele (folosind `if-else`).

SOLUȚIA TA · PYTHON

2

PROBLEMA 2

Clasificarea unui triunghi

SCRIERE COD

Se citesc trei laturi `a`, `b`, `c`. Dacă formează un triunghi (inegalitatea triunghiului), afișează `"echilateral"`, `"isoscel"` sau `"oarecare"`; altfel, afișează că nu formează un triunghi.

SOLUȚIA TA · PYTHON