



Întâlnirea 5

Testarea White Box

Obiective Întâlnire 5

- Să înțelegi ce este un Statement Coverage
- Să folosești Decision Coverage
- Exerciții practice
- Sumar - Testarea Statică vs Testarea Dinamică

Tehnici de testare WHITE BOX



- ❑ Reprezintă testarea pe baza structurii interne a unei componente sau a unui sistem și presupune cunoașterea codului sursă pe baza căruia acționează programul. We need to know what is happening on the inside.
- ❑ Se mai numește și testare structurală, testarea cutiei de sticlă, testarea cutiei transparente sau backend testing.
- ❑ Un mix între testare blackbox și testare whitebox se numește testarea cutiei gri sau grey-box testing.

STATEMENT COVERAGE

❖ **Instrucțiune (Statement/Instruction)** – Este un element al unui cod scris într-un limbaj de programare care acționează ca o activitate pe care sistemul trebuie să o facă (*executa asta!*). Mai poartă numele de statement.

❖ **Criteriu de acoperire (Coverage Criteria)**- Putem spune că un statement a fost acoperit dacă a fost executat cel puțin o dată.

❖ **Acoperire (Coverage)** – procentajul statementurilor executate în raport cu numărul total de statementuri din cod.

DECISION COVERAGE

- ❖ Constă în măsurarea procentajului de decizii (de regulă luate pe baza rezultatului evaluării unei condiții ca fiind adevărată sau falsă) ce au fost testate în cadrul unei suite de testare. În această tehnică, testele sunt create pentru a acoperi rezultatele specific ale ramurilor decizionale.
- ❖ Criteriu de acoperire: fiecare decizie trebuie să fie acoperită atât pe ramura de adevărat, cât și pe cea de fals cel puțin o dată.
- ❖ **Decision coverage este mai acoperitor (mai puternic) decat statement coverage.**
- ❖ **100% decision coverage garantează 100% statement coverage, dar nu și invers.**

Complexitatea Ciclomatică

- Complexitatea ciclomatică reprezintă o unitate de măsură pentru complexitatea codului.
- Este de regulă o noțiune folosită mai degrabă de către echipa de dezvoltare pentru a evalua eficacitatea codului și a înțelege dacă e nevoie să fie făcute îmbunătățiri asupra lui.
- Formula pentru calculul complexității ciclomatice este: $L - N + 2P$ (formula în română) sau $E - N + 2P$ (formula în engleză).
- Pentru testare este suficient să știm formula de calcul. NU o vom aplica dincolo de examenul de ISTQB.

E = Edges, L = Laturi (numărul de linii care unesc doua noduri)

N = Numărul de noduri

P = Numărul de partiții independente (scheme în care toate nodurile sunt legate între ele de cel puțin o linie)

Elementele unui scheme

Nodes



Line for connecting two nodes





Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 1

```

INPUT  a
INPUT  b
PRINT  2
    IF a >= 16 THEN
        PRINT 5
        IF a >= 10 THEN
            PRINT 3
        ENDIF
    ELSE
        IF a < 35 THEN
            a = a + 6
            a = a + 4
            PRINT 0
        ENDIF
        a = b + 1
        PRINT 8
    ENDIF
        b = a + 1
        PRINT 2
        b = a + 8

```

Answers:

2,2

3,3

2,4

2,3



Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 2

```
INPUT  a
INPUT  b
b = b + 1
a = b + 3
PRINT 2
a = a + 1
IF a > 33 THEN
    PRINT 7
    IF b <= 15 THEN
        PRINT 7
        b = b + 5
        PRINT 9
    ENDIF
ENDIF
```

Answers:

- a) 0,3
- b) 1,3
- c) 0,4
- d) 1,4



Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 3

```
INPUT a
INPUT b
IF a >= 38 THEN
    IF a > 38 THEN
        PRINT 1
    PRINT 0

    ENDIF
    PRINT 2
    a = a + 5
ELSE
    b = b + 1
ENDIF
a = b + 6
a = b + 9
PRINT 7
PRINT 8
```

Answers:

- a) 2,3
- b) 3,4
- c) 3,3
- d) 2,2



Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 4

```
INPUT a
INPUT b
PRINT9
b=a+4
PRINT10
b=a+10
IF a>39 THEN
    b=b+1
    b=a+10
ELSE
    b=a+5
    IF a<=39 THEN
        PRINT0
    ENDIF
ENDIF
ENDIF
```

Answers:

- a) 1,2
- b) 3,3
- c) 1,4
- d) 2,3



Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 5

```
INPUT  a
INPUT  b
IF a >= 21 THEN
    PRINT 3
    IF b>5 THEN
        PRINT 1
    ELSE
        a = a + 6
    ENDIF
ENDIF
PRINT 0
PRINT 10
PRINT 2
a = a + 9
```

Answers:
a) 2,3
b) 1,2
c) 2,2
d) 3,1



Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 6

```

INPUT  a
INPUT  b
b = b + 8
IF b < 42 THEN
    PRINT 2
    PRINT 4
    IF a <= 8 THEN
        PRINT 3
        a = a + 8
    ENDIF
ELSE
    a = a + 3
    a = a + 5
ENDIF
b = a + 4
PRINT 3
IF a >= 29 THEN
    a = a + 7
ENDIF

```

Answers:

- a) 2,3
- b) 2,4
- c) 1,3
- d) 1,2



Pentru pseudocodul alăturat desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 7

```
INPUT  a
INPUT  b
b = b + 10
b = b + 2
IF b >= 12 THEN PRINT 9
                a = b + 6
                b = a + 4
ENDIF
IF b >= 7 THEN b = b + 1
                a = b + 8
                a = b + 7
ENDIF
b = a + 10
```

Answers:

- a) 1,2
- b) 0,2
- c) 2,1
- d) 1,3



Pentru exercițiul alăturat scrie pseudocodul aferent, apoi pe baza lui desenează schema de whitebox și calculează numărul de test case-uri necesare pentru a avea 100% statement coverage și 100% decision coverage și respectiv complexitatea ciclomatică pentru fiecare din schemele de mai jos.

Exercițiul 8

Dacă un client are peste 65 de ani, atunci va primi un discount de 15% la orice călătorie.

Dacă un senior peste 65 de ani va călători în timpul iernii când cererea este mai mică, atunci va primi un discount adițional de 10%.

Dacă seniorul alege să călătorească la clasa întâi, atunci va trebui să plătească o taxă adițională de 3%, altfel va trebui să plătească doar taxa de procesare de 1%.

Dacă un client are sub 65 de ani, acesta poate primi un discount de 10% dacă va călători cu cel puțin un copil.

De asemenea, reducerea de sezon și taxele de clasă se aplică și celor sub 65 de ani.

În urma acestui curs ar trebui să poți face următoarele:

- ☐ Să înțelegi diferența între testare statică și testare dinamică.
- ☐ Să înțelegi diferența între testare statică și testare dinamică.
- ☐ Să poți să explici diferitele tehnici de testare și să le folosești în viața reală.

Testarea Statică vs Testarea Dinamică - SUMAR -

“Design-ul este atât o chestiune de a
găsi probleme cât și o chestiune de a
le rezolva”.

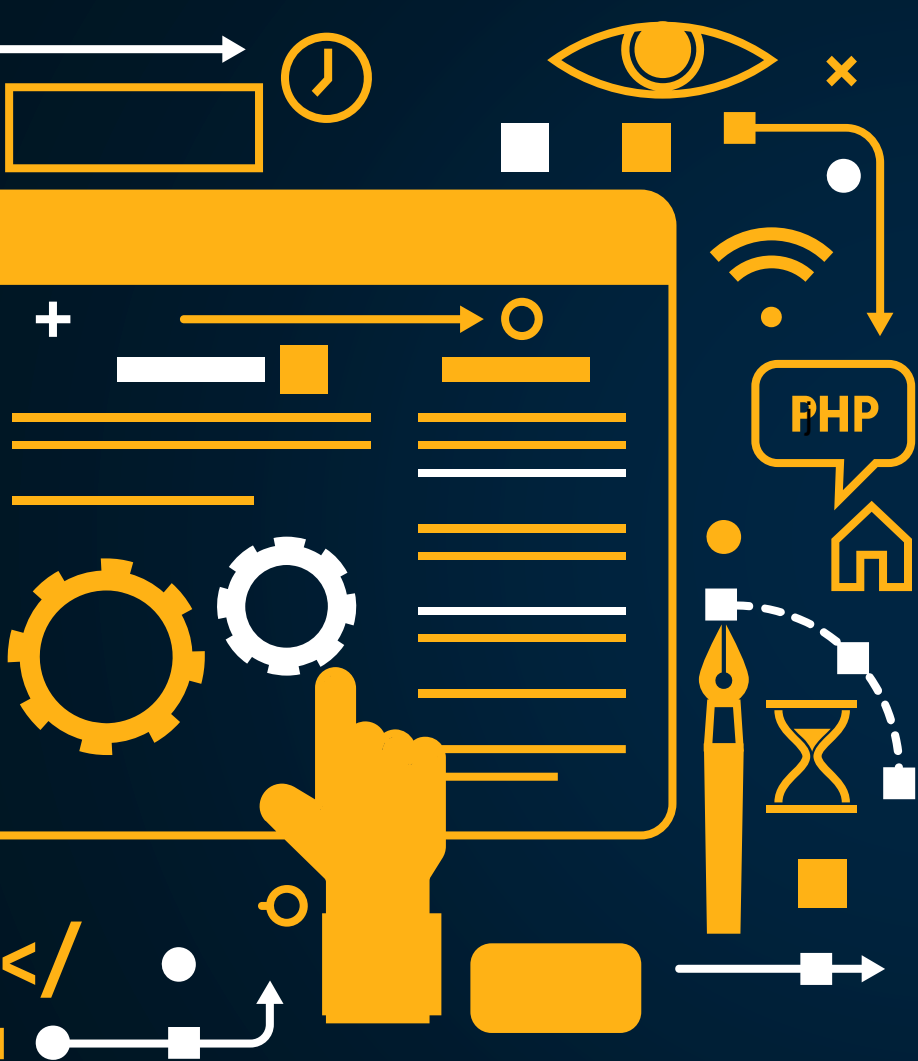
- Bryan Lawson



Întrebare de interviu:

- Cum ai testa o matură?
- Ce reprezintă testarea White Box?
- Care este mai acoperitoare dintre Statement Coverage și Decision Coverage?

Întrebări & curiozități?



Complexitatea Ciclomatică

Testarea White Box

Ce Este Complexitatea Ciclomatică?

- Măsura complexității unui program se bazează pe numărul drumurilor independente care pot fi alese în timpul executării programului. Drept urmare, cu cât avem mai multe decizii în program, cu atât acesta va fi mai complex.

Calculul Complexității Ciclomatrice

Formula pentru calculul complexității ciclomatrice este: $L - N + 2P$ (formula în română) sau $E - N + 2P$ (formula în engleză).

- E = Edges, L = Laturi (numărul de linii care unesc două noduri).
- N = Numărul de noduri.
- P = Numărul de partiții independente (scheme în care toate nodurile sunt legate între ele de cel puțin o linie).

Evaluarea Complexității Ciclomatice

Părerile pentru evaluarea complexității ciclomatice sunt împărțite, dar în general putem să ne orientăm după următoarele intervale:

- 1-10 - program simplu, nu prea există riscuri;
- 11-20 – complexitate mică, riscuri scăzute;
- 21-50 – prea complex, riscul este mediu și este recomandat să luăm măsuri de reducere a complexității.
- > 50 - prea complex. Greu de testat. Riscuri majore care se pot reproduce.

Complexitatea Ciclomatică

- Complexitatea ciclomatică reprezintă o unitate de măsură pentru complexitatea codului.
- Este de regulă o noțiune folosită mai degrabă de către echipa de dezvoltare pentru a evalua eficacitatea codului și a înțelege dacă e nevoie să fie făcute îmbunătățiri asupra lui.
- Formula pentru calculul complexității ciclomatice este: $L - N + 2P$ (formula în română) sau $E - N + 2P$ (formula în engleză).

E = Edges, L = Laturi (numărul de linii care unesc două noduri);

N = Numărul de noduri;

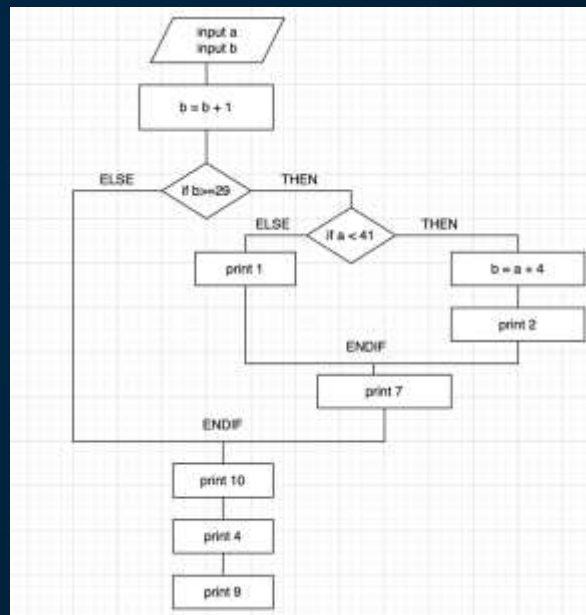
P = Numărul de partiții independente (scheme în care toate nodurile sunt legate între ele de cel puțin o linie).

Calcul Practic

Pe baza pseudocodului alăturat trebuie să desenăm schema care va sta la baza calculului.

```

INPUT a
INPUT b
b = b + 1
IF b >= 29 THEN
    IF a < 41 THEN b = a + 4
    PRINT 2
ELSE PRINT 1
ENDIF
PRINT 7
ENDIF
PRINT 10
PRINT 4
PRINT 9
  
```



N = 12

L = 13

P = 1

CC = E-N+2P

CC = 12-11+2

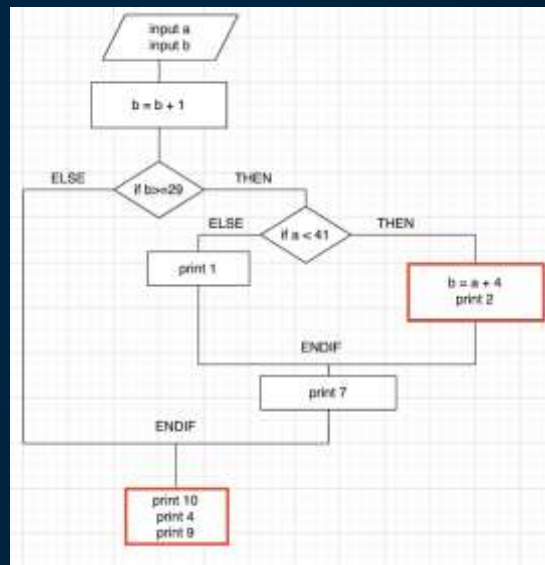
CC = 3

Simplificarea Schemei Anterioare

Pentru simplificarea schemei putem să grupăm instrucțiunile (NU DECIZIILE) pentru a reduce numărul de noduri și linii.

```

INPUT a
INPUT b
b = b + 1
IF b >= 29 THEN
    IF a < 41 THEN b = a + 4
    PRINT 2
ELSE PRINT 1
ENDIF
PRINT 7
ENDIF
PRINT 10
PRINT 4
PRINT 9
  
```



$N = 7$
 $E = 8$
 $P = 1$

$V(G) = E - N + 2P$
 $V(G) = 9 - 8 + 2$
 $V(G) = 3$

Simplificarea calculelor

- Dacă mai multe statementuri sunt plasate consecutive, putem să le grupăm într-un singur nod pentru simplificarea graficului.
- Complexitatea ciclomatică poate de asemenea să fie calculată prin adunarea tuturor deciziilor, la care adăugăm o unitate.

$$CC = D + 1$$

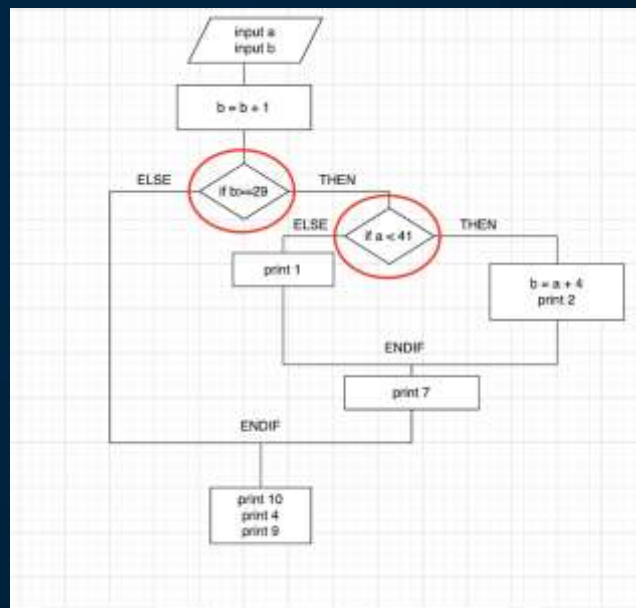
D = Nod decizional

Simplificarea Adițională

Pentru simplificarea calculului putem să numărăm câte decizii avem în schemă.

```

INPUT a
INPUT b
b = b + 1
IF b >= 29 THEN
    IF a < 41 THEN b = a + 4
    PRINT 2
ELSE PRINT 1
ENDIF
PRINT 7
ENDIF
PRINT 10
PRINT 4
PRINT 9
  
```



$D = 2$

$CC = D + 1$

$CC = 3$



**Notează întrebările
pentru trainer**