Ministrul Educației, Culturii și Cercetării din Moldova

Universitatea Tehnică a Republicii Moldova

Inginerie Software și Automatică

**REPORT**

Laboratorul N.4

*Testarea Software*

Grupa : TI-163 Chiriac Ion

Verificat : M.Catruc

Chisinau, 2019

**Cerinte :**

* - Proiectarea cazurilor de testare.
* - algoritmul McCabe.
* - Criteriile de acoperire.
* - Controlul fluxului de date.
* - Tehnica White-Box.

**Obiective :**

* Înțelegerea noțiunii de cazuri de testare.
* Înțelegerea metodologiei de dezvoltare a aplicației:
  + Dezvoltare condusă de test
  + Dezvoltare bazată pe cod
* Formarea competențelor de partiționare în criteriile de acoperire.
* Utilizați graficul fluxului de date.

**Sarcina:**

1. Desenați graficul de control al debitului programului.

2. Testați programul în funcție de criteriile de acoperire

3. Dovada eșantionului în cazul în care rezultatele eronate pot fi obținute pentru diferite criterii de acoperire.

4. Observați cazurile speciale și testele elaborate.

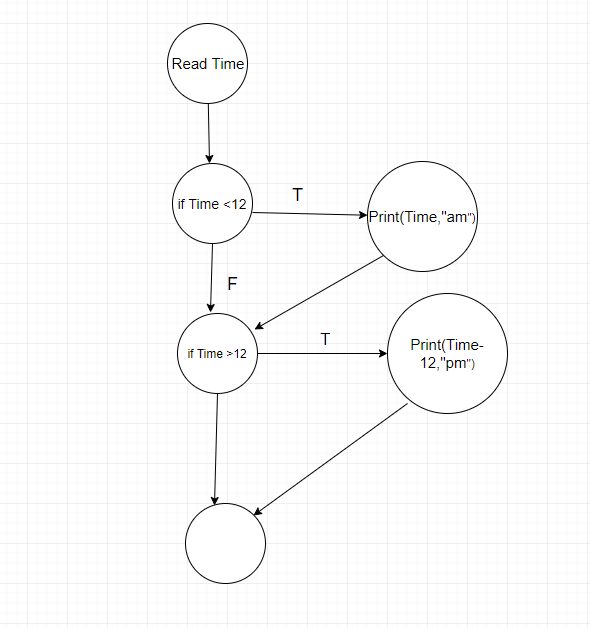
5. Utilizați tehnica McCabe pentru a testa căile de bază.

**Implimentarea sarcini:**

1: **Begin**  
2: **Read Time**  
3: **If Time < 12 Then**  
4: **Print(Time, ”am”)**  
5: **Endif**  
6: **If Time > 12 Then**  
7: **Print(Time-12, “pm”)**  
8: **Endif**  
9: **End**

**Sarcina 1:**

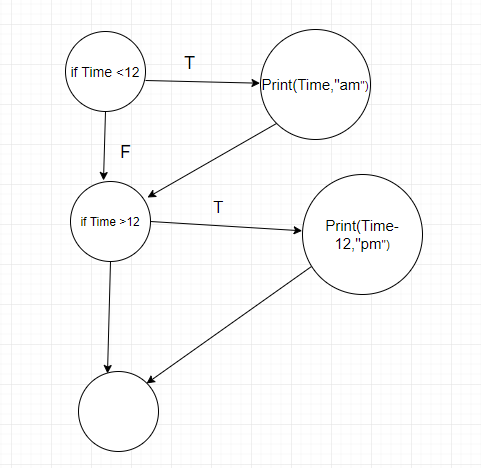
Graficul de control al fluxului este una din cele 2 tehnici de testare a cutiei transparente.

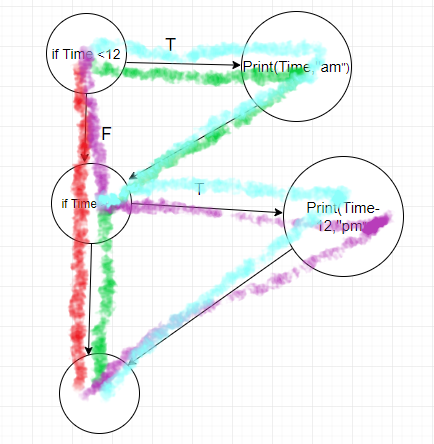


**Sarcina 2 :**

Deci, în această sarcină trebuie să utilizăm criteriile de acoperire pentru programul nostru. De asemenea, știm că există mai multe tipuri de acoperire.

**Level 0** : “100% covering the instructions”.



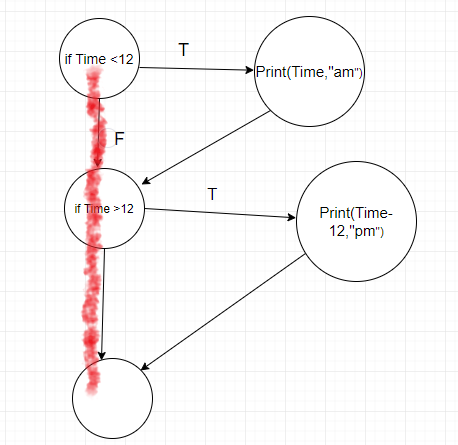
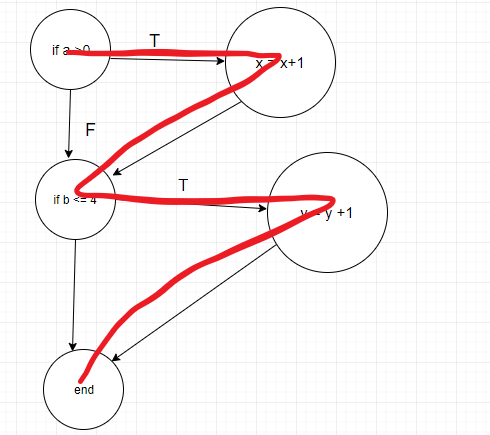


Putem vedea că putem genera diferite moduri de execuție, în acest caz există 4 moduri. Să vedem dacă putem aplica doar un singur caz pe acest tip de criterii de acoperire, deoarece putem că este cel mai mic decât restul tipurilor de acoperire.

**Răspuns**: putem avea doar un caz de test, **Timp = 12.** Am obținut că aceste criterii vor pierde o mulțime de metode netestate.

**Level 1 :** “100% acoperirea deciziilor sau acoperirea sucursalelor”.

În exemplul anterior nu pot examina ambele cazuri din acest criteriu, doar primul, când ieșirea va fi Falsă. Pentru a doua parte, în cazul în care rezultatul trebuie să fie Adevărat și toate deciziile ar trebui să fie acoperire, voi face un alt exemplu.



a = 3 , b = 3.

**Primul caz** : c1s1c2s3; c1c2s3 **Al doilea caz :** : c1s1c2s2s3

**Level 2:** “100% acoperirea condițiilor”

Acest nivel înseamnă, de asemenea, că toate condițiile ar trebui să fie trecute cel puțin o dată și o dată rezultatul ar trebui să fie Adevărat și a doua oară ar trebui să fie Fals. Deci, voi da un exemplu conform celui de-al doilea exemplu de cod.

**Cazuri :** a = 3; b = 3; a = 0; b = 5;

**Level 3:** “100% acoperirea deciziilor / condițiilor”

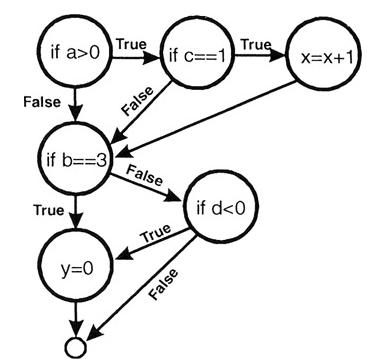
Acest nivel înseamnă că putem face "și" logică între condiții, dar acest lucru nu se va executa vreodată.

**Level 4:** “100% acoperirea condițiilor multiple”.

Acest nivel înseamnă a efectua logica "sau" între condiții. Să luăm un alt exemplu de cod pentru acest tip de nivel.

if (a>0 && c==1) {x=x+1;}

if (b==3 || d<0) {y=0;}

 a>0, c=1, b=3, d<0

a≤0, c=1, b=3, d≥0

a>0, c≠1, b≠3, d<0

a≤0, c≠1, b≠3, d≥0

**Sarcina 4:**

Nu putem încheia testarea și procesul de efectuare a cazurilor de testare, fără a testa căile de bază. Testarea structurală se bazează pe munca lui Tom McCabe, care a folosit tipologia graficului de control al fluxului pentru efectuarea cazurilor de testare. Deci, pentru acest proces există mai mulți pași:

* Extrageți graficul de control al debitului la software.
* Calculați numărul ciclomatic (C).
* Calculați un set de căi de bază (C).
* Creați testul pentru fiecare mod de bază.
* Executați testele.

**Primul cod :** C = 6-5+2 = 3;  
**Al doilea cod:** C = 6-5+2 = 3;

|  |  |
| --- | --- |
| **Complexitatea ciclomatică** | **Nivelul de complexitate și risc** |
| 1-10 | un program simplu, fără riscuri prea mari |
| 11-20 | risc mai complex, moderat |
| 21-50 | risc mai complex, moderat |
| greater than 50 | program netestabil, risc foarte mare |

**Concluzie :**

În această lucrare de laborator am obținut abilități care funcționează cu testul White-Box și criteriile de acoperire a acestui tip de testare. De asemenea, am învățat despre testele structurale McCabe. În plus, modul de efectuare a testelor minime pentru testarea cât mai mult posibil a produselor.