PROIECT de laborator TESTARE si VERIFICARE

**CERINTA PROBLEMA:**

Să se implementeze un program care primește de la tastatură un număr natural n, 1<=n<100,si un sir de de n numere s, un număr natural m, 1<=m<=n, și un număr d, d>0. Să se întoarcă adevărat dacă suma primelor numere de pe (m-1) pozitii inmultita cu numarul natural m divide d, altfel fals.

**I. Testare functionala**

**1.Partitionarea in clase de echivalenta**

**Clasele de echivalenta ale intrarilor:**

Domeniul de intrări:

-> 1<=**n**<100 lunigimea sirului de numere intregi

-> un sir **s** de numere naturale de lungime n

-> un numar intreg **m**, 1<m<n

-> un numar intreg **d**, d>0

Clase de echivalenta :

* pentru **n** :

**N\_1 = {1,2,...,99}**

**N\_2 = {n| n<1}**

**N\_3 = {n| n>99}**

* pentru **s** nu determinam clase de echivalenta
* pentru **m**

**M\_1 = {1,2,...,n}**

**M\_2 = {m| m<1}**

**M\_3 = {m| m>n}**

* pentru **d**

**D\_1 = {d| d>=0}**

**D\_2 = {d| d<0}**

**Clasele de echivalenta ale iesirilor:**

**O\_1 = {"M divide suma."}**

**O\_2 = {"M nu divide suma."}**

**0\_3 = {“Nu se satisfac conditiile“}**

**Prin urmare, avem doua clase de echivalenta finale :**

**R\_1(s) = {o | suma primelor (m-1)\*m elemente divide d }**

**R\_2(s) = {o | suma primelor (m-1)\*m elemente nu divide d }**

**R\_3(s) = {o | parametrii de intrare nu satisfac conditiile }**

**Asa obtinem urmatoarele clase de echivalenta globale :**

C\_1111 = {(n,s,m,d) | n in N\_1, m in M\_1, d in D\_1, r in R\_1}

C\_1112 = {(n,s,m,d) | n in N\_1, m in M\_1, d in D\_1, r in R\_2}

C\_111 = {(n,s,m,d) | n in N\_1, m in M\_1, d in D\_1}

C\_112 = {(n,s,m,d) | n in N\_1, m in M\_1, d in D\_2}

C\_12 = {(n,s,m,d)| n in N\_1, m in M\_2}}

C\_13 = {(n,s,m,d)| n in N\_1, m in M\_3}}

C\_2 = {(n,s,m,d)| n in N\_2}

C\_3 = {(n,s,m,d)| n in N\_3}

**Alegem urmatoarele date pentru test:**

|  |  |
| --- | --- |
| C\_1111 = (10,{4,7,2,6,9,2,3,3,7,9},9,6) | **TRUE** |
| C\_1112 = (11,{3,7,4,9,1,2,1,6,7,3,3},3,6) | **FALSE** |
| C\_112 = (4,{3,7,4,9},2,-1) | **d is lower or equal 0** |
| C\_12 = (5,{3,7,4,9,7},0,3) | **m is lower or equal 1** |
| C\_13 = (5,{3,7,4,9,7},9,3) | **m is bigger or equal n** |
| C\_2 = (-5,{3,7,4,9},2,5) | **n is lower than 1** |
| C\_3 = (102,{3,7,4,9},2,5) | **n is bigger than 99** |
| C\_4 = (4,{1,2,3},2,5) | **length of s is not equal with n** |

**2.**  **Analiza valorilor de frontiera**

Frontiere: -> pentru n:0,1,99,100

-> pentru m: {1,n}

-> d din {1, .., infinit}

Clase de frontiera:

N\_1 = {(n,s,m,d)| n=1 sau n=99}

N\_2 = {(n,s,m,d)| n=0}

N\_3 = {(n,s,m,d)| n=100}

M\_1 = {(n,s,m,d)| m=1}

M\_2 = {(n,s,m,d)| m=n}

D\_1 = {(n,s,m,d)| d=1}

D\_2 = {(n,s,m,d)| d=0}

|  |  |
| --- | --- |
| Date de intrare | Rezultat |
| C\_111=(1,{1},1,1) | **m is lower or equal 1** |
| C\_121=(99,{1},1,1) | **length of s is not equal with n** |
| C\_211=(0,{},1,1) | **n is lower than 1** |
| C\_311=(100,{...},1,1) | **n is bigger than 99** |

**3. Graful cauza efect**

Cauze:

C1: n<1

C2: n>=100

C3: s.length != n

C4: m<=1

C5: m>=n

C6: d<=0

C7: not(C1&C2&C3&C4&C5&C6) & divide d

Efecte:

E1: arunca Exception cu mesajul “n is lower than 1"

E2: arunca Exception cu mesajul “n is bigger than 99"

E3: arunca Exception cu mesajul “length of s is not equal with n "

E4: arunca Exception cu mesajul “m is lower or equal 1"

E5: arunca Exception cu mesajul “ m is bigger or equal n "

E6: arunca Exception cu mesajul “d is lower or equal 0"

E7: returneaza TRUE

Cazuri de test:

Pentru E1 -> C1=1, restul cazurilor nu ne intereseaza

|  |  |
| --- | --- |
| C1 | 1 |
| C2 | 0 |
| C3 | 0 |
| C4 | 0 |
| C5 | 0 |
| C6 | 0 |
| C7 | 0 |
| E1 | 1 |
| E2 | 0 |
| E3 | 0 |
| E4 | 0 |
| E5 | 0 |
| E6 | 0 |
| E7 | 0 |

Pentru E2 -> C2=1, restul cazurilor nu ne intereseaza

|  |  |
| --- | --- |
| C1 | 0 |
| C2 | 1 |
| C3 | 0 |
| C4 | 0 |
| C5 | 0 |
| C6 | 0 |
| C7 | 0 |
| E1 | 0 |
| E2 | 1 |
| E3 | 0 |
| E4 | 0 |
| E5 | 0 |
| E6 | 0 |
| E7 | 0 |

Pentru E3 -> C3=1, restul cazurilor nu ne intereseaza

|  |  |
| --- | --- |
| C1 | 0 |
| C2 | 0 |
| C3 | 1 |
| C4 | 0 |
| C5 | 0 |
| C6 | 0 |
| C7 | 0 |
| E1 | 0 |
| E2 | 0 |
| E3 | 1 |
| E4 | 0 |
| E5 | 0 |
| E6 | 0 |
| E7 | 0 |

Pentru E4 -> C4=1, restul cazurilor nu ne intereseaza

|  |  |
| --- | --- |
| C1 | 0 |
| C2 | 0 |
| C3 | 0 |
| C4 | 1 |
| C5 | 0 |
| C6 | 0 |
| C7 | 0 |
| E1 | 1 |
| E2 | 0 |
| E3 | 0 |
| E4 | 1 |
| E5 | 0 |
| E6 | 0 |
| E7 | 0 |

Pentru E5 -> C5=1, restul cazurilor nu ne intereseaza

|  |  |
| --- | --- |
| C1 | 0 |
| C2 | 0 |
| C3 | 0 |
| C4 | 0 |
| C5 | 1 |
| C6 | 0 |
| C7 | 0 |
| E1 | 0 |
| E2 | 0 |
| E3 | 0 |
| E4 | 0 |
| E5 | 1 |
| E6 | 0 |
| E7 | 0 |

Pentru E6 -> C6=1, restul cazurilor nu ne intereseaza

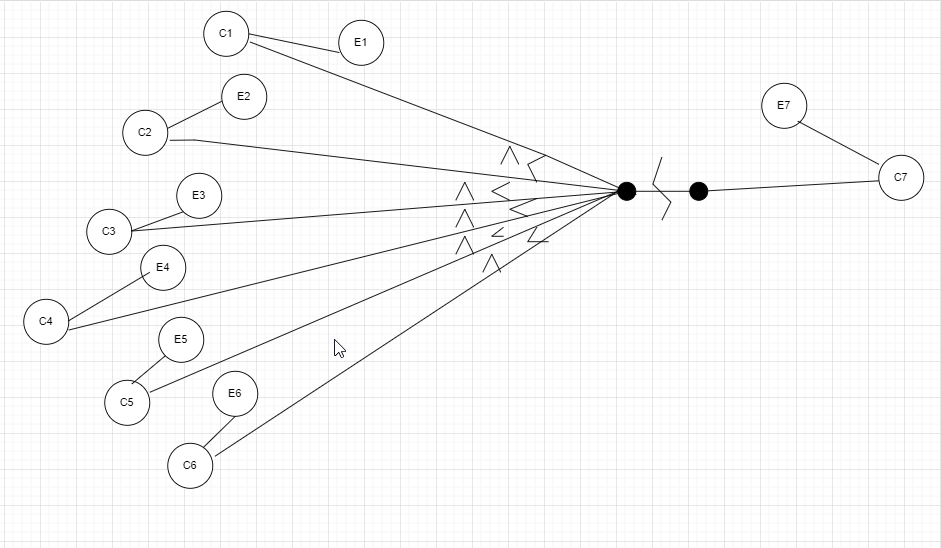
|  |  |
| --- | --- |
| C1 | 0 |
| C2 | 0 |
| C3 | 0 |
| C4 | 0 |
| C5 | 0 |
| C6 | 1 |
| C7 | 0 |
| E1 | 0 |
| E2 | 0 |
| E3 | 0 |
| E4 | 0 |
| E5 | 0 |
| E6 | 1 |
| E7 | 0 |

Pentru E7 -> ~(C1^C2^C3^C4^C5^C6)

|  |  |
| --- | --- |
| C1 | 0 |
| C2 | 0 |
| C3 | 0 |
| C4 | 0 |
| C5 | 0 |
| C6 | 0 |
| C7 | 1 |
| E1 | 0 |
| E2 | 0 |
| E3 | 0 |
| E4 | 0 |
| E5 | 0 |
| E6 | 0 |
| E7 | 0 |

**Tabelul final este :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| C2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| C3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| C4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| C5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| C6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| C7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| E1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| E2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| E3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| E4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| E5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| E6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| E7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

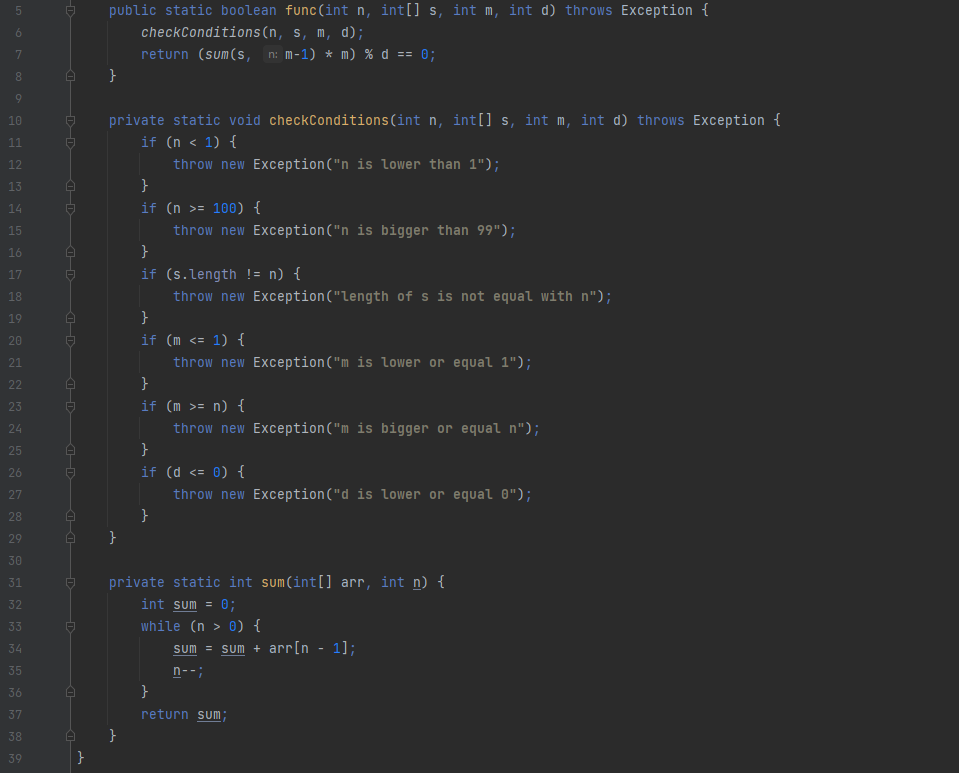


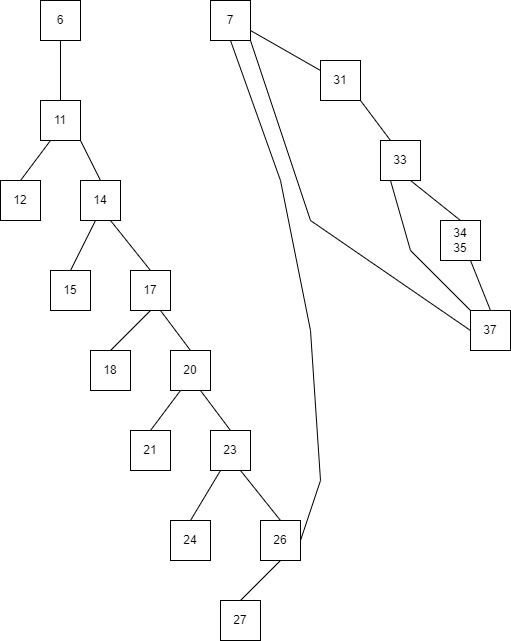
Datele de test pentru acest grafic sunt urmatoarele:

|  |  |
| --- | --- |
| T\_1=(0,{},,) | n is lower than 1 |
| T\_2=(100,{},M,D) | n is bigger than 99 |
| T\_3=(4,{1,2,3,4,5},2,3) | length of s is not equal with n |
| T\_4=(3,{1,2,3},1,7) | m is lower or equal 1 |
| T\_5=(5,{1,2,3,4,5},5,3) | m is bigger or equal n |
| T\_6=(8,{1,2,3,4,5,6,7,8},3,-1) | d is lower or equal 0 |
| T\_7=(5,{1,2,3,4,5},4,4) | TRUE |

3. Sa se transforme programul intr-un graf orientat si, pe baza acestuia, sa se gaseasca un set de teste care satisface criteriul modified condition/decision coverage (MC/DC).

Programul din metoda main:



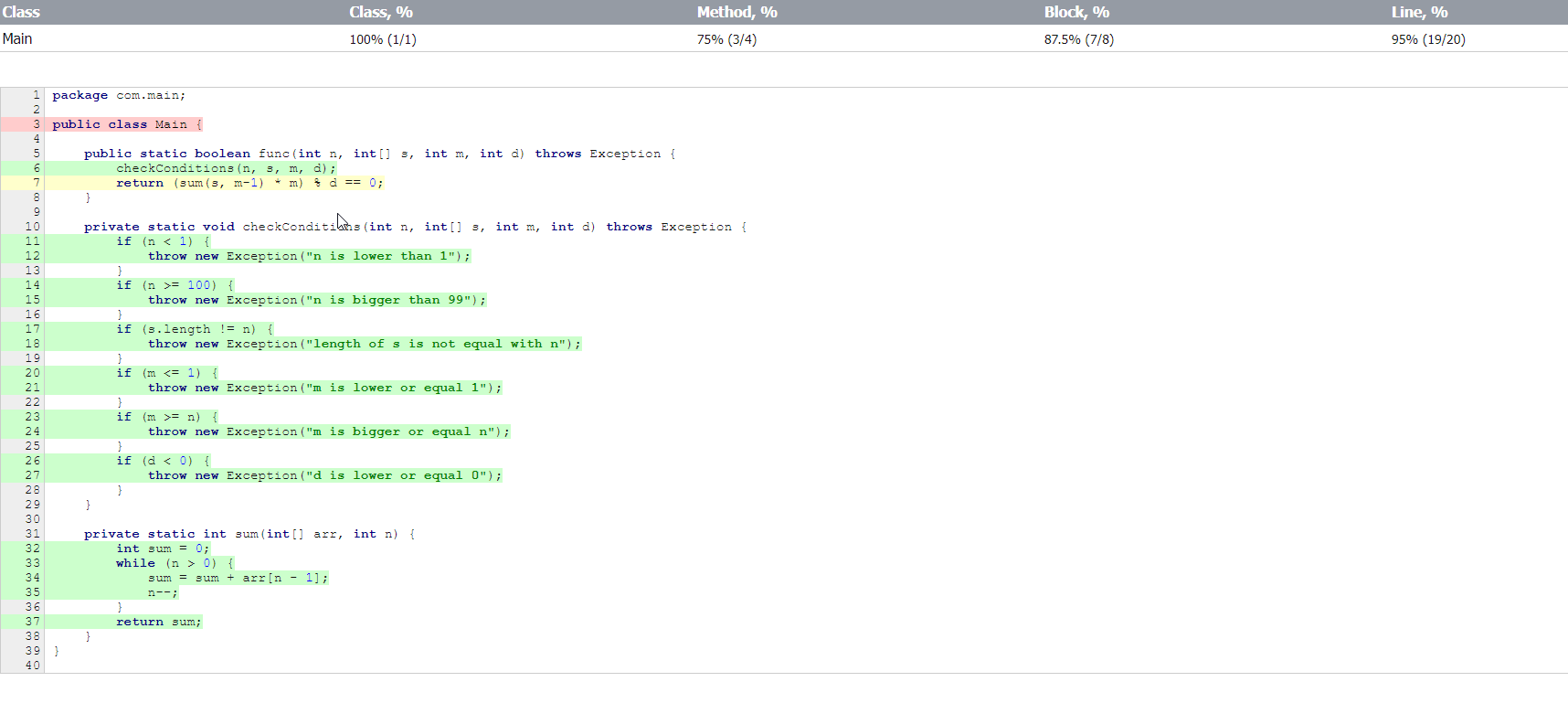
Graful orientat al programului : 

|  |  |
| --- | --- |
| Deciziile programului sunt : | Conditii individuale : |
| **if( n<1 || n>20 || m < 1 || m> n || d < 0 )** | **n<1** ; **n>20;** **m<1**; **m>n** ;**d<0** |

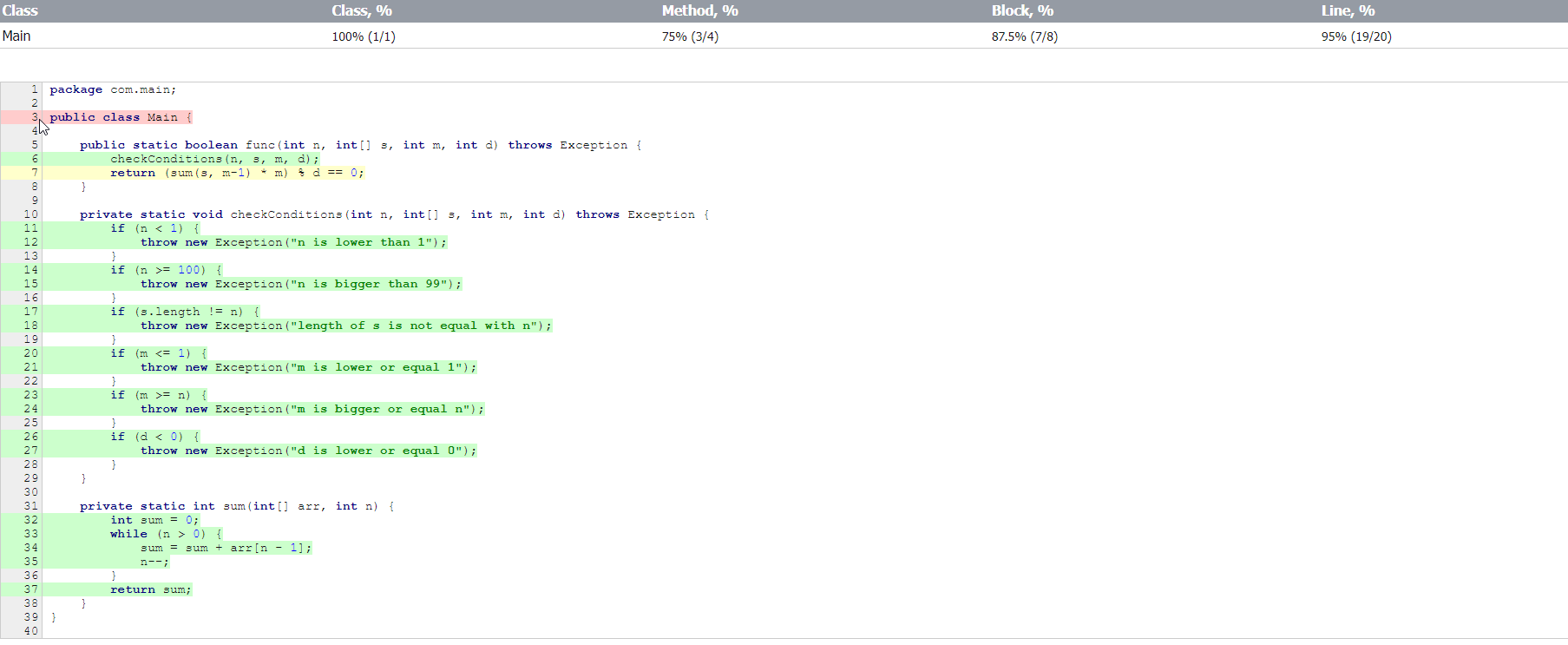
Setul de date care satisface decision coverage este urmatorul:

|  |  |
| --- | --- |
| T1=() |  |
| T2=() |  |

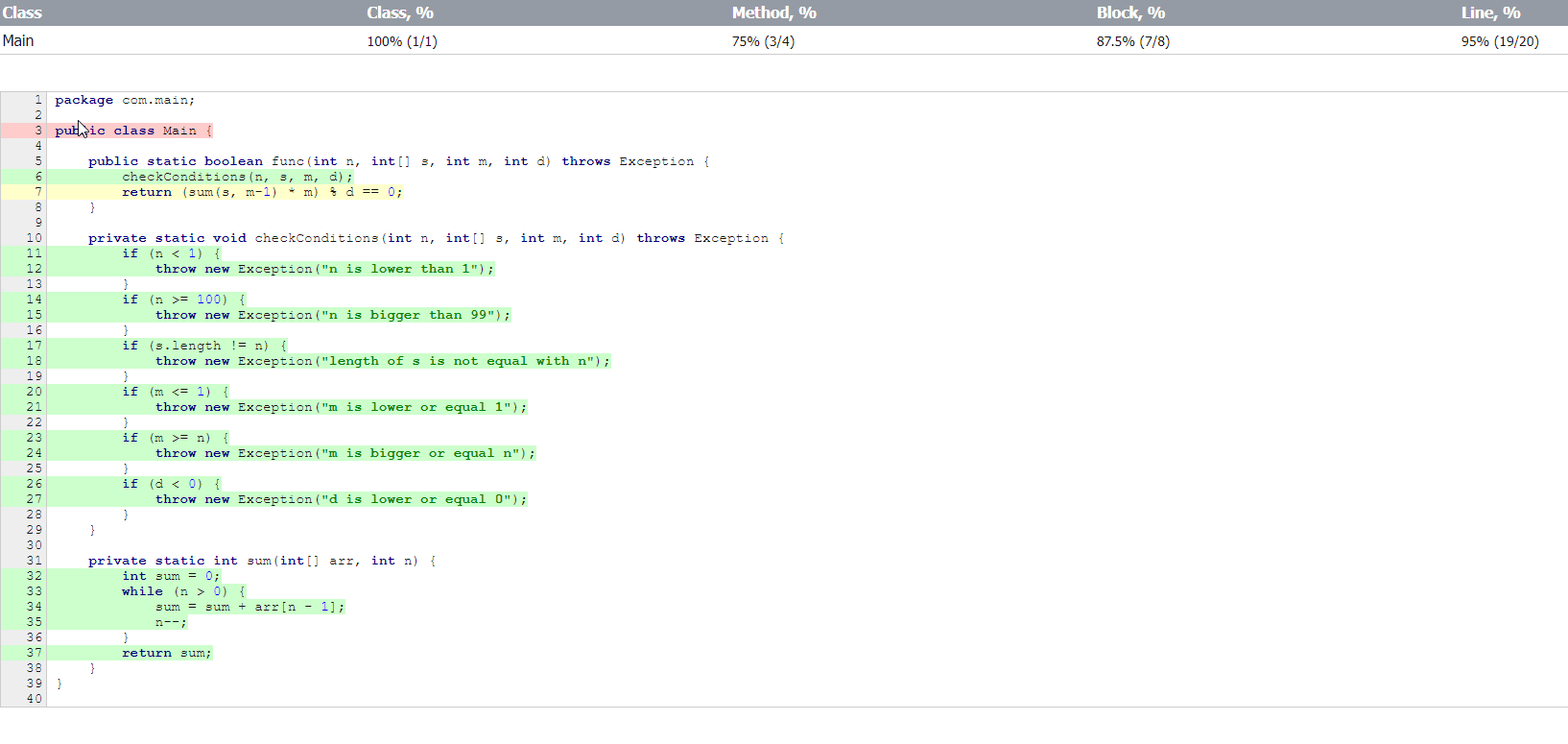
Test coverage :

1. **EquivalencePartitioning**

**2. BoundaryValueAnalysis**



**3. CauseEffectGraphing**



4. Sa se scrie un mutant de ordinul 1 echivalent al programului.

LINIA 25:  -> 

5. Pentru unul dintre cazurile de testare de mai sus sa se scrie un mutant ne-echivalent care sa fie omorat de catre test si un mutant ne-echivalent care sa nu fie omorat de catre test.

- Mutant ne-echivalent omorat:

LINIA 16: -> if (s.length != n) { => if (s.length == n) {

- Mutant ne-echivalent care nu a fost omorat:

LINIA 20: -> if (s.length != n) { => if (s.length <= n) {