

PROIECT

TESTAREA SISTEMELOR SOFTWARE

I. Specificația Programului și Obiective

1.1. Descriere Generală

Scopul acestui proiect este dezvoltarea și testarea unui modul software destinat automatizării procesului de acordare a bursei studentești. Programul implementează logica de decizie bazată pe performanța academică (media) și situația socială (venitul) a studenților.

1.2. Reguli de Business

Funcționalitatea este implementată în clasa **ScholarshipSystem**, metoda **decideScholarship**.
Regulile de acordare sunt următoarele:

1. **Validare:** Se returnează EROARE dacă datele de intrare nu sunt în intervalele valide (Media 1-10, Absențe ≥ 0 , Venit ≥ 0)
2. **Bursa de Merit:** Se acordă pentru **Media** ≥ 9.50 și **Absențe** ≥ 10
3. **Bursa de Studiu:** Se acordă pentru **Media** ≥ 8.50 și **Absențe** ≥ 20 (dacă nu e eligibil pentru Merit)
4. **Bursa Socială:** Se acordă pentru **Venit** < 1500 RON, **Media** ≥ 5.00 și **Absențe** < 30
5. **Respins:** Orice alt caz

II. Generarea Datelor de Test (Black-Box Testing)

Au fost aplicate trei tehnici de testare funcțională pentru a genera setul de teste unitare.

2.1. Equivalence Partitioning (EP)

Domeniul de intrare a fost împărțit în clase de echivalență, selectându-se câte o valoare reprezentativă pentru fiecare clasă validă și invalidă.

Tip Clasă	Descriere Clasă (Condiții)	Valori Intrare (Media, Absențe, Venit)	Rezultat Așteptat
Validă	Condiții Merit Îndeplinite	9.80, 8, 2000	BURSA_MERIT
Validă	Condiții Studiu Îndeplinite	9.00, 15, 2000	BURSA_STUDIU
Validă	Condiții Social Îndeplinite	7.50, 25, 1000	BURSA_SOCIALA
Validă	Nicio condiție îndeplinită	9.00, 35, 3000	RESPINS
Invalidă	Media în afara intervalului	0.50, 0, 0	EROARE

2.2. Boundary Value Analysis (BVA)

Au fost testate valorile aflate la limitele intervalelor de decizie pentru a verifica corectitudinea operatorilor relaționali (\leq vs $<$).

- **Limita Merit (Media):**
 - Input: 9.50 → Rezultat: BURSA_MERIT
 - Input: 9.49 → Rezultat: BURSA_STUDIU
- **Limita Merit (Absențe):**
 - Input: 10 → Rezultat: BURSA_MERIT
 - Input: 11 → Rezultat: BURSA_STUDIU
- **Limita Social (Venit):**
 - Input: 1499 → Rezultat: BURSA_SOCIALA
 - Input: 1500 → Rezultat: RESPINS

2.3. Cause-Effect Graphing

S-a modelat logica folosind un graf cauză-efect.

- **Cauze (Input):** C1 (Media ≥ 9.5), C2 (Abs ≤ 10), C3 (Media ≥ 8.5) etc.
- **Efecte (Output):** E1 (Merit), E2 (Studiu)
- Testele implementate (ex: **testCE_Merit_Logic**) verifică relația:
 - IF (C1 AND C2) THEN E1

III. Analiza Acoperirii (Code Coverage)

Pentru verificarea completitudinii setului de teste, s-a folosit utilitarul de *Code Coverage* integrat în IntelliJ IDEA.

Rezultate Obținute:

- **Class Coverage:** 100% (1/1)
- **Method Coverage:** 100% (19/19)
- **Line Coverage:** 100% (27/27)

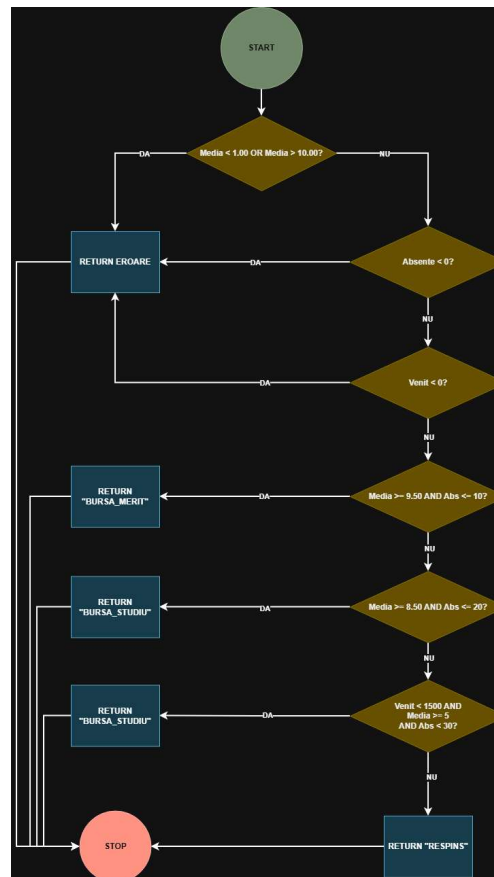
Seturile de teste EP și BVA, combinate cu testele structurale pentru ramurile de eroare, au asigurat parcurgerea tuturor instrucțiunilor din cod.

Coverage Summary for Class: ScholarshipSystemTest (<empty package name>)

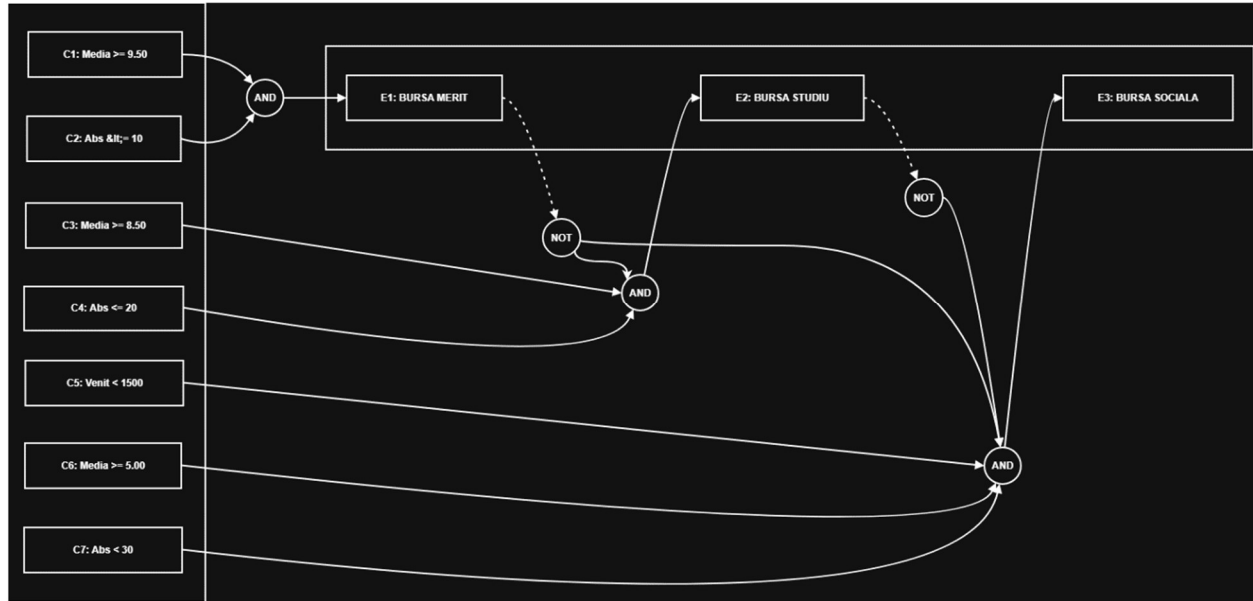
Class	Class, %	Method, %	Line, %
ScholarshipSystemTest	100% (1/1)	100% (19/19)	100% (27/27)

IV. Testare Structurală (MC/DC)

4.1. Diagrama



4.2. Modified Condition/Decision Coverage (MC/DC)



Pentru decizia complexă de acordare a bursei sociale:

if (venit < 1500 && media >= 5.00 && absente < 30)

S-a demonstrat independența fiecărei condiții atomice prin următorul set de teste:

ID Test	Venit < 1500	Media ≥ 5.00	Absențe < 30	Rezultat	Observații
T1	True (1000)	True (7.00)	True (25)	SOCIAL	Toate condițiile True
T2	False (2000)	True (7.00)	True (25)	RESPINS	Venitul decide rezultatul
T3	True (1000)	False (4.50)	True (25)	RESPINS	Media decide rezultatul

ID Test	Venit < 1500	Media ≥ 5.00	Absențe < 30	Rezultat	Observații
T4	True (1000)	True (6.00)	False (35)	RESPINS	Absențele decid rezultatul

Aceste cazuri sunt acoperite de testele: testCoverage_Social_ForceEntry, testEP_Valid_Respins, testCoverage_Social_Fail_Media și testCoverage_Social_Fail_Absente.

V. Testarea prin Mutații

S-au creat mutanți software pentru a evalua calitatea testelor.

5.1. Mutant Echivalent (Ordinul 1)

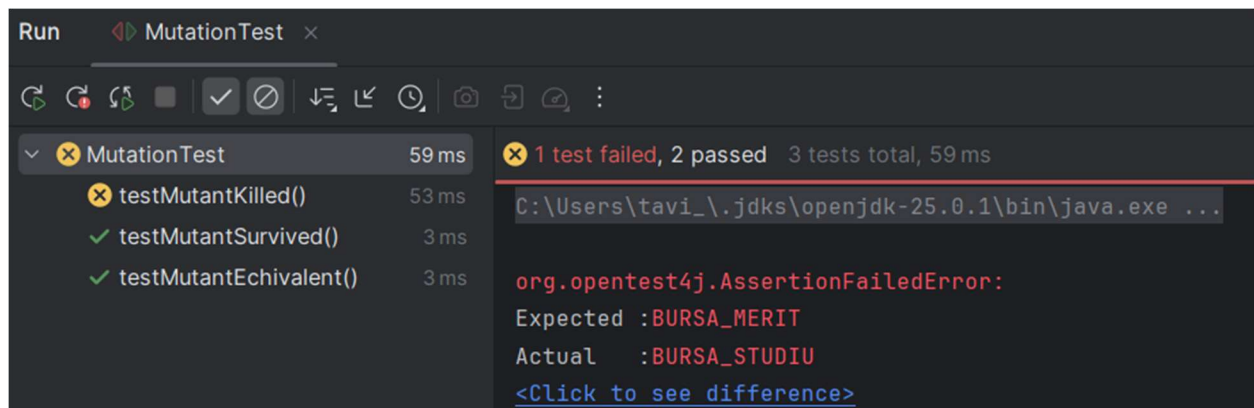
- **Cod Original:** if (absente < 0)
- **Cod Mutant:** if (absente <= -1)
- **Rezultat:** Testele au trecut (Verde)
- **Explicație:** Deoarece absente este un număr întreg, condițiile sunt logic identice; niciun test nu poate detecta diferența, deoarece comportamentul programului nu se schimbă

5.2. Mutant Ne-Echivalent OMORÂT (Killed)

- **Modificare:** S-a modificat pragul de absențe pentru Bursa de Merit din 10 în 5
- **Cod Mutant:** if (media >= 9.50 && absente <= 5)
- **Rezultat:** Testul testMutantKilled (cu 8 absențe) a eșuat (**Roșu**)
- **Concluzie:** Testul a detectat eroarea introdusă, omorând mutantul

5.3. Mutant Ne-Echivalent SUPRAVIETUITOR (Survived)

- **Modificare:** S-a relaxat condiția mediei pentru Merit de la 9.50 la 9.00
- **Cod Mutant:** `if (media >= 9.00 && absente <= 10)`
- **Rezultat:** Testul `testMutantSurvived` (cu media 9.80) a trecut (**Verde**)
- **Concluzie:** Deoarece valoarea de test (9.80) satisface ambele condiții (și pe cea corectă, și pe cea greșită), testul nu a putut detecta eroarea



```
Run MutationTest x
1 test failed, 2 passed 3 tests total, 59 ms
testMutantKilled() 53 ms
testMutantSurvived() 3 ms
testMutantEchivalent() 3 ms
C:\Users\tavi_\.jdk\openjdk-25.0.1\bin\java.exe ...
org.opentest4j.AssertionFailedError:
Expected :BURSA_MERIT
Actual   :BURSA_STUDIU
<Click to see difference>
```

VI. Bibliografie

[1] Suport de curs Testarea Sistemelor Software.

[2] Documentația JUnit 5.

[3] Documentația IntelliJ IDEA Code Coverage.

VII. Link catre git repository

https://github.com/ciupacabral/Proiect_TSS.git